

PUSH NOTIFICATION MONITORING SISTEM PINTU AIR BERBASIS ANDROID MENGUNAKAN FIREBASE CLOUD MESSAGING

Frandia Fernando¹, Arini², Feri Fahrianto³

^{1,2,3}Teknik Informatika, Fakultas Sains dan Teknologi
Universitas Islam Negeri Syarif Hidayatullah Jakarta

¹frandiasans@gmail.com, ²arini@uinjkt.ac.id, ³feri_fahrianto@uinjkt.ac.id

ABSTRACT

Flood disaster is very common, especially in Jakarta City. One way to control water debit is to create a dam or reservoir. In each reservoir there is a water gate which serves to dispose of unwanted water gradually or continuously according to the water volume present in the dam. The Regional Disaster Management Agency (BPBD) Jakarta City needed a system that can monitor any existing water gates of the reservoir, this is to facilitate water gate inspectors to provide information. By using firebase cloud messaging technology that will be applied in the application of push-based android water doors. The application will provide notification and water gate data in real time, making it easier for users to get data in real time. Firebase Cloud Messaging is a cross platform solution that allows you to send messages reliably at no cost. In addition, the use of API as a processing medium to pull data from the web BPBD, optimize its function to mobile android. BPBD as the agency that tackling the disaster can know the state of the environment quickly and accurately. Applications created not only made for BPBD only but the general public can also to know the condition of the water gate. In addition, there are also ways of handling floods and what to do at each level of water level. For the next, this application can add other technollgy also IoT technology.

Keywords: *Push Notification, Firebase, Cloud Messaging, Monitoring, Android*

ABSTRAK

Bencana banjir sangat sering terjadi di Kota Jakarta. Salah satu cara untuk mengendalikan debit air adalah dengan membuat suatu bendungan atau waduk. Di setiap waduk terdapat pintu air untuk membuang air yang berlebihan secara bertahap sesuai dengan volume air yang ada pada bendungan tersebut. Badan Penanggulangan Bencana Daerah (BPBD) kota Jakarta membutuhkan sistem yang dapat memonitor setiap pintu air waduk yang ada untuk mempermudah pekerjaan pengawas pintu air dalam memberikan informasi. Sistem ini menggunakan teknologi *firebase cloud messaging* lintas *platform* pada aplikasi *push notification* pintu air berbasis *Android*. Aplikasi akan memberikan *notification* dan data atau kondisi pintu air secara *real time* sehingga mempermudah *user* untuk mendapatkan data secara *real time*. Sistem juga menggunakan API sebagai media pemrosesan untuk menarik data dari situs web BPBD, fungsinya untuk mengoptimalkan ke *mobile android* sehingga BPBD selaku instansi yang menanggulangi bencana dapat mengetahui keadaan lingkungan secara cepat dan tepat. Sistem juga menyediakan informasi cara penanggulangan terhadap banjir dan apa yang harus dilakukan pada setiap status ketinggian air. Untuk pengembangan selanjutnya sistem dapat dikaji dengan menggunakan teknologi dan perangkat lainnya seperti teknologi IoT.

Kata Kunci: *Push Notification, Firebase, Cloud Messaging, Monitoring, Android*

I. PENDAHULUAN

Lembaga yang menanggulangi bencana di bawah pemerintah dan bertanggung jawab kepada Gubernur BPBD adalah Badan Penanggulangan Bencana Daerah (BPBD). BPBD dibentuk untuk mengatasi ancaman terhadap bencana alam ekstrim dan non-alam seperti banjir, gempa bumi, tanah longsor, wabah penyakit dan lain-lain. Berdasarkan hasil statistik, bencana di Indonesia dari rentang tahun 2011-2017 banjir merupakan masalah utama yang paling sering terjadi dengan persentase tertinggi yaitu 30% dari total keseluruhan bencana [1]. Juga bahwa bencana banjir sangat sering terjadi terutama di daerah-daerah yang dekat dengan aliran sungai [2].

Bendungan atau waduk dibangun untuk menahan laju air, pengaturan ketinggian air perlu dilakukan untuk menghadapi curah hujan, sehingga pengawasan (*monitoring*) perlu dilakukan [3]. Untuk pengaturan air dilakukan oleh pintu air yaitu dengan membuang air yang tidak diinginkan secara bertahap dapat dilakukan dengan cara manual atau otomatis, yang juga dijaga oleh penjaga pintu air yang akan memberikan informasi ketinggian dan status air. Di BPBD kota Jakarta hal ini masih dilakukan secara manual melalui SMS *gateway*, *web* bpbd dan media sosial lain dari penjaga pintu air ke penjaga pintu air lain. Sehingga masyarakat tidak dapat secara langsung menerima dan mengetahui informasi kondisi pintu-pintu air yang ada dan berjaga-jaga jika akan terjadi banjir, padahal kondisi di Jakarta bencana alam yang sering terjadi adalah banjir [4].

Dari kondisi tersebut diperlukan aplikasi yang dapat digunakan untuk memberikan pemberitahuan atas informasi kepada masyarakat secara tepat sasaran dan secara *real time* tentang kondisi air yang berada di bendungan atau waduk dibawah BPBD Kota Jakarta dan juga dapat sebagai *monitoring* [5] untuk para penjaga pintu air terhadap kondisi air. *Output monitoring* dapat untuk mengetahui kesesuaian proses telah berjalan yaitu informasi pintu air dan akan berguna pada perbaikan atas sistem informasi yang ada.

Menurut [6] menerapkan *firebase cloud messaging* untuk memberikan informasi kegiatan belajar mengajar, seperti pindah kelas, pergantian waktu belajar mengajar, atau kelas kosong kepada mahasiswa yang mengambil kelas. Pada aplikasi ini dosen belum bisa

menambahkan atau memindahkan mahasiswa ke kelasnya, pengujian untuk mengirimkan *push notification* ke beberapa ponsel *android* masih belum bisa dilakukan.

Selain itu [7], pada aplikasi ini dapat mengirimkan informasi dari kampus melalui aplikasi *web* kepada mahasiswa dan mengurangi kesalahan sasaran dalam menyampaikan informasi dengan menerapkan *Firebase Cloud Messaging* untuk memberikan informasi perkuliahan secara *real time*.

Menurut [8], *cloud messaging* digunakan untuk memberikan *notification* lokasi kepada *user* yang berintegrasi dengan *location based service*, Selama tahap perancangan antarmuka pengguna tidak terus diuji dan disempurnakan sebelum diimplementasikan sehingga menyita waktu lama dan biaya mahal.

Pada [9], *firebase cloud messaging* dapat digunakan untuk merancang fitur *push notification* pada aplikasi media laporan aspirasi. Pada fitur *push notification Firebase, Cloud Messaging* belum dapat mengirimkan pesan *notification* lintas perangkat pada berbagai aplikasi lain.

Menurut [10], menerapkan *push notification* berbasis *android* sebagai model pesan secara *realtime* dan dapat memberi manfaat secara individu. Pada [11], membangun *firebase push notification* untuk aplikasi informasi perkuliahan berbasis *mobile* seperti untuk notifikasi kehadiran dosen dan informasi kegiatan mahasiswa.

Menurut [12] membangun *push notification system* untuk pesan notifikasi yang dapat memberikan peringatan dalam melakukan kendali dalam hal efisiensi penggunaan listrik. Penelitian [13], menggambarkan *framework* prototipe *software* yang dirancang untuk mengirim *push notifications* menggunakan GCM (*google cloud messaging*) dan APNS (*apple push notification service*).

Dari beberapa studi literatur yang digunakan tersebut maka dalam penelitian ini akan memfokuskan pada:

1. Pembuatan *push notification monitoring system* atas informasi waduk BPBD Kota Jakarta yang akan dikirim ke *mobile (android) user* [14], [15], [16] dan berbentuk *panel notification* [17].
2. Mendukung data secara *real time* dan menggunakan API untuk menarik data dari BPBD [18] dengan menggunakan JSON untuk menarik (*retrieve*) data dari *firebase cloud mesagging* [19].

3. Menggunakan bahasa pemrograman PHP [20] dengan bantuan *framework* CodeIgniter untuk sisi *Coding internal* dan java, XML untuk sisi tampilan User Interface, *database* yang digunakan adalah MySQL, menggunakan perangkat lunak *Android Studio* untuk melakukan pengodean aplikasi ini karena aplikasi ini tergolong ringan atau tidak terlalu memberatkan sistem komputer.
4. Pengiriman informasi secara *real time* ke *user*.
5. Implementasi *push notification* menggunakan *tool* *Firebase Cloud Messaging* [21].
6. Aplikasi memberikan informasi ketinggian air, daftar pintu air, info cuaca, dan info kontak.
7. Sistem *monitoring* dengan mengacu pada 4 hal yaitu: Sederhana dan mudah dimengerti *user friendly*, fokus pada beberapa indikator utama yaitu informasi kondisi pintu air, aplikasi teknis yang terarah dan terstruktur menggunakan pedoman 5W1H (apa, mengapa, siapa, kapan, dimana dan bagaimana), Prosedur pengumpulan dan penggalian data memiliki prosedur tepat dan sesuai [15].

Dengan adanya penelitian ini, aplikasi *push notification monitoring* sistem yang dibuat khususnya pada pintu air dapat sebagai pelayanan dalam menginformasikan kondisi pintu air, aplikasi terhubung langsung dengan *database* yang terdapat dari *server* pusat yaitu BPBD. Aplikasi dapat diakses oleh masyarakat sehingga memudahkan masyarakat untuk mendapatkan informasi pintu air yang diinginkan dan statusnya, juga disediakan cara penanggulangan terhadap banjir dan langkah apa yang harus dilakukan pada setiap status ketinggian air. Keseluruhan ini semua dapat dijadikan sebagai sarana untuk *monitoring* di BPBD kota Jakarta.

II. METODOLOGI

2.1. Metode Pengumpulan Data

Dalam melakukan pengumpulan data peneliti selain menggunakan studi pustaka juga menggunakan metode studi lapangan yaitu metode observasi dan metode wawancara.

A. Metode Wawancara

Peneliti melakukan konsultasi dan tanya jawab langsung dengan staf terkait di Badan

Penanggulangan Bencana Daerah Jakarta (BPBD Jakarta) pada tanggal 10 Januari 2018 di kantor BPBD Jakarta, untuk mendapatkan penjelasan mengenai sistem di BPBD dan kebutuhan untuk *user* akan aplikasi *Push notification monitoring* pintu air berbasis *android*

B. Metode Observasi

Observasi dilakukan pada BPBD Jakarta Jl. Abdul Muis No. 66. Petojo Selatan Jakarta Pusat, dilakukan pada bulan Januari sampai bulan Februari 2018. Objek yang akan di observasi adalah sistem pemberitahuan informasi tentang pintu air Jakarta. Observasi dilakukan untuk mengetahui secara langsung sistem yang sudah berjalan pada BPBD Jakarta mengenai alur dari pemberitahuan kondisi pintu air yang nantinya akan dibuatkan ke model sistem *smartphone android* yang bersifat tepat sasaran dan *real time*.

2.2. Metode Pengembangan Sistem

Tahap ini peneliti menggunakan metode berorientasi objek dengan model pengembangan *Waterfall* [22] dan *tool* UML untuk pemodelannya [23].

A. Requirement Analysis and Definition

Merupakan tahapan penetapan fitur pada aplikasi *push notification* pintu air BPBD Jakarta. Kendala dan tujuan sistem melalui konsultasi dengan pengguna sistem. Semua hal tersebut ditetapkan secara rinci dan berfungsi sebagai spesifikasi aplikasi *push notification monitoring* sistem pada pintu air BPBD Jakarta.

B. System and Software Design

Dalam tahapan ini dibuat suatu arsitektur sistem berdasarkan persyaratan yang telah diperoleh pada fase sebelumnya. Dan juga mengidentifikasi dan menggambarkan abstraksi dasar aplikasi *push notification monitoring* sistem pintu air BPBD Jakarta dan hubungan-hubungannya.

C. Implementation

Tahapan ini merupakan tahap melakukan pengkodean aplikasi *push notification* menggunakan *firebase cloud messaging* dengan bahasa pemrograman Java dan basis datanya MySQL. Hasil dari tahap ini adalah

program komputer sesuai dengan desain yang telah dibuat pada tahap desain

D. Integration and System Testing

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan dengan menggunakan *blackbox testing*. Setelah itu sistem akan dikirim ke pengguna sistem [15].

E. Operation and Maintenance

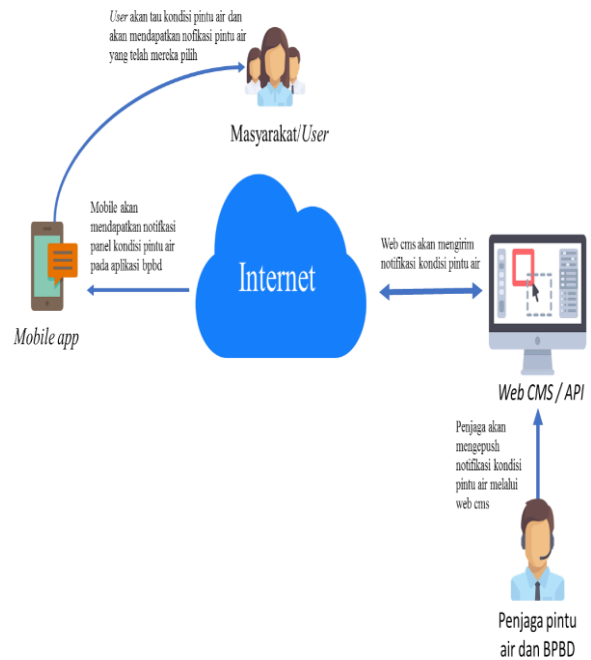
Pada tahap ini, aplikasi dioperasikan dan jika terjadi kesalahan maka akan dilakukan perbaikan dan proses pemeliharaan dapat dilakukan.

III. HASIL DAN PEMBAHASAN

3.1 Requirement Analysis and Definition

Dari hasil analisa sistem yang berjalan diketahui bahwa sistem masih secara manual dimana petugas masih menggunakan *broadcast SMS*, *Whatsapp*, dan media komunikasi lainnya. Informasi mengenai kondisi pintu air dikirimkan secara manual oleh penjaga pintu air ke BPBD melalui *radio* dan setelah BPBD mendapatkan data kondisi pintu air kemudian BPBD memberikan informasi kepada masyarakat melalui media komunikasi.

Sistem usulan yang diajukan, seperti pada Gambar 1, menggunakan *firebase cloud messaging* akan digunakan dalam pembuatan aplikasi *push notification monitoring system* pada pintu air BPBD Jakarta. Aplikasi menggunakan API, dengan memanfaatkan teknologi *firebase cloud messaging* yang terintegrasi dengan web CMS, berbasis *android* yang dapat menyajikan layanan *notification* pintu air secara *real time*, data selalu terintegrasi ke *database*. Aplikasi dapat memberikan informasi kondisi pintu air dengan menggunakan *Firestore database*, aplikasi dijalankan secara *online*, hal ini untuk dapat mendapatkan informasi secara *real time* dari *firebase cloud messaging* dan web cms, sehingga setiap data pintu air yang masuk ke web cms dapat dikirim pada *mobile app* ini dapat di-*install* secara *online* dan dijalankan secara *online* pada *device Android*.

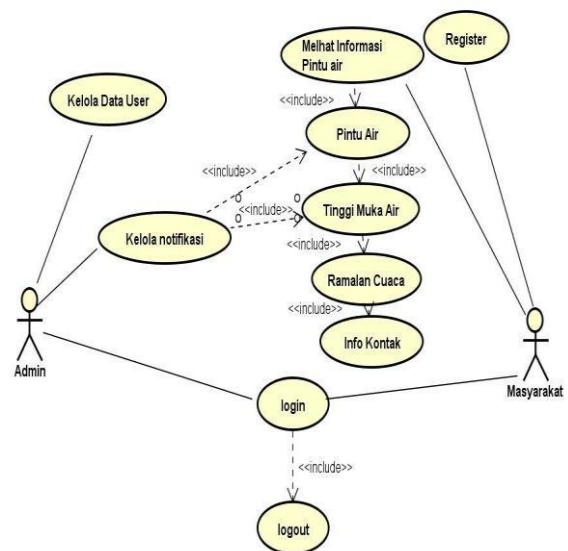


Gambar 1. Sistem usulan *push notification monitoring system*

3.2 System and Software Design

Berikut hasil perancangan dengan menggunakan UML:

A. Usecase Diagram

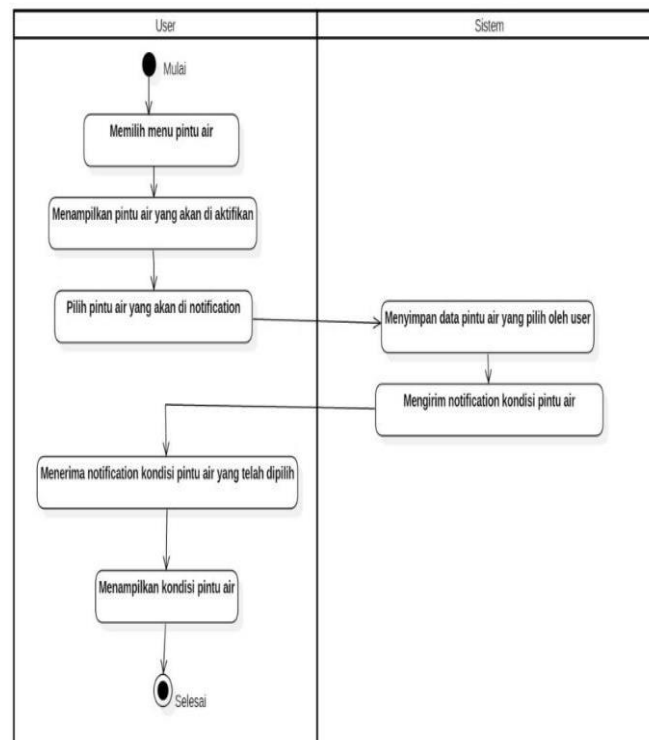


Gambar 2. Usecase diagram

Pada Gambar 2, Admin dan user (masyarakat) adalah actor. Admin akan mengelola *notification* dan data user yang *login*. User dapat melakukan *register* atau *login* untuk memasuki aplikasi dan dapat memilih menu yang ada di aplikasi.

B. Activity Diagram

Berikut salah satu *activity diagram* aplikasi.

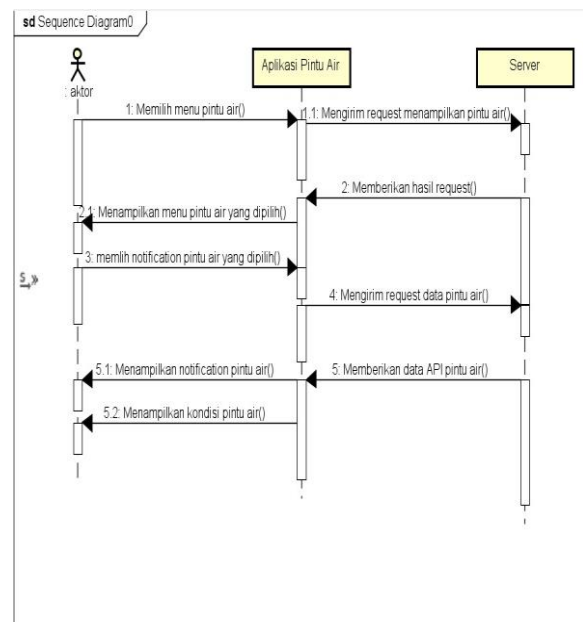


Gambar 3. *Activity diagram* pintu air

Pada Gambar 3, terlihat bahwa aktor membuka aplikasi menu pintu air, kemudian menu menampilkan pintu air yang akan dipilih lalu pilih pintu air yang akan diaktifkan *notification* kemudian sistem akan menyimpan data pintu air yang telah dipilih oleh *user*. Sistem akan mengirim *notification* kondisi pintu air setiap 1 jam ke *user* sesuai dengan pintu air yang telah *user* aktifkan *notification* dan *user* akan menerima *notification* kondisi pintu air sesuai yang mereka pilih.

C. Sequence Diagram

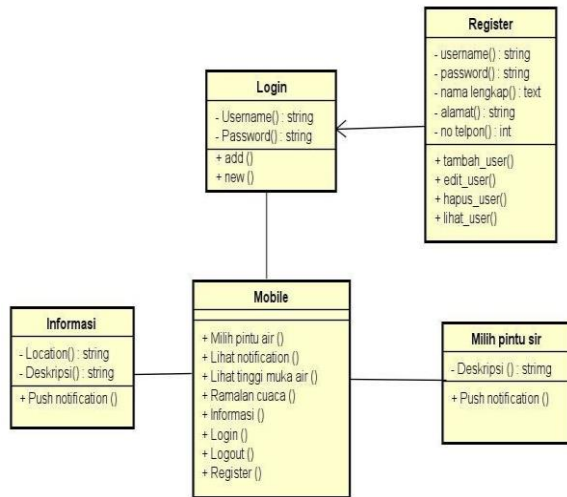
Pada Gambar 4, *user* menekan tombol *login* yang ada pada tampilan menu awal aplikasi. Setelah itu *user* diminta untuk memilih pintu air yang akan di-*notification*-kan, dan sistem menunggu pintu air mana saja yang akan pilih *user*. Setelah *user* memilih pintu air, sistem akan menyimpan data pada web cms dan akan mengirim *notification* pintu air yang dipilih *user*, dan selanjutnya sistem akan memberikan *notification* kondisi pintu air secara *real time* yang dipilih oleh *user*. Setelah penggunaan selesai, *user* keluar dari aplikasi dengan menekan tombol kembali pada *device*, maka sistem akan menutup aplikasi.



Gambar 4. *Sequence diagram* menampilkan data pintu air

D. Class Diagram

Gambar 5, mendeskripsikan struktur objek pada sistem (kelas objek dan hubungan antar kelas objek).



Gambar 5. Class diagram

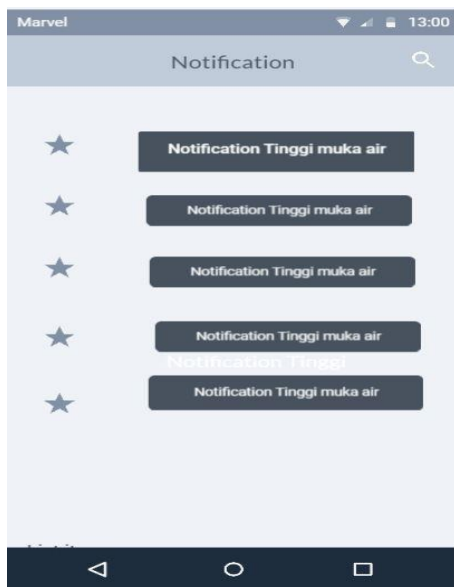
E. Perancangan API

Bentuk respon *server* yang digunakan:

1. Kode 200, menandai *request* berhasil *error*-nya *false*
2. Kode 400, proses verifikasi data yang gagal
3. Kode 404, proses verifikasi yang gagal dikarenakan datanya tidak ada di *database*
4. *Error*, ada 2 *true* dan *false*
5. *Message* hasil dari output API
6. *Device token*, kode didapat dari *jenered firebase* fungsinya sebagai alamat *ip/code handphone* yang akan digunakan untuk mengirim data sesuai API *handphone*

F. Perancangan *User Interface*

Bentuk salah satu perancangan *user interface* aplikasi.



Gambar 6. Mockup notification

3.3 Implementation

Pada Gambar 7, kode *notification* if *remotemessage*==null; tidak menampilkan apa-apa, data berisi if *remotemessage* untuk menampilkan panel *notification*.

```
package com.bpbd.info.fcni;

import ...

public class MyFirebaseMessagingService extends FirebaseMessagingService {

    private static final String TAG = MyFirebaseMessagingService.class.getSimpleName();

    private NotificationUtils notificationUtils;
    private String setData;

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        Log.e(TAG, msg: "From: " + remoteMessage.getFrom());

        if (remoteMessage == null)
            return;

        // Check if message contains a notification payload.
        if (remoteMessage.getNotification() != null) {
            Log.e(TAG, msg: "Notification Body: " + remoteMessage.getNotification().getBody());
            handleNotification(remoteMessage.getNotification().getBody());
        }

        // Check if message
        // a data payload.
        if (remoteMessage.getData().size() > 0) {
            Log.e(TAG, msg: "Data Payload: " + remoteMessage.getData().toString());

            try {
                handleDataMessage(remoteMessage);
            } catch (Exception e) {
                Log.e(TAG, msg: "Exception: " + e.getMessage());
            }
        }
    }
}
```

Gambar 7. Kode notification

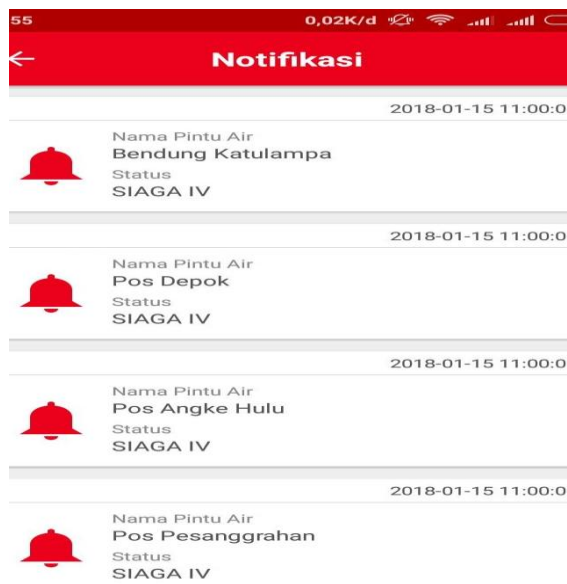
3.4 Integration and Sistem Testing

Berikut adalah salah satu hasil *testing* aplikasi. Jika *user* memilih salah satu *pintu air* maka *user* akan mendapatkan *push notification* informasi *pintu air* tersebut.



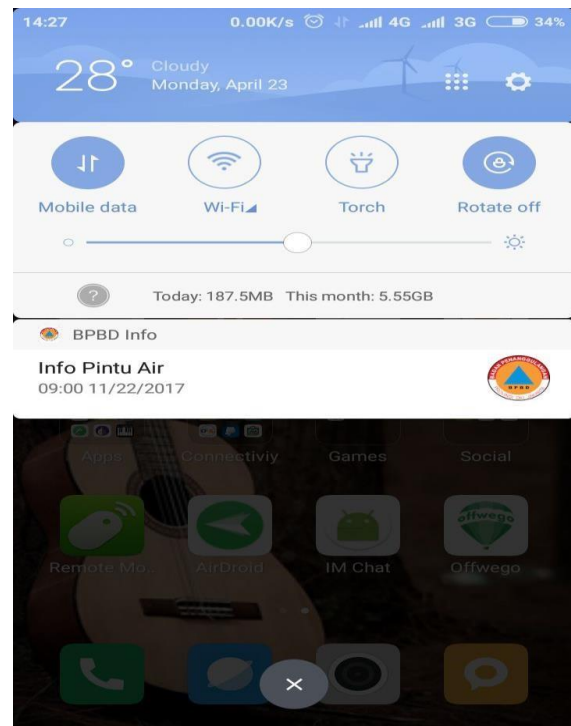
Gambar 8. Pengujian skenario memilih *notification* pintu air

Pada Gambar 8 (Bendungan Katulampa), *user* akan mendapatkan informasi pintu air Bendung Katulampa, *user* juga bisa memilih tidak hanya satu pintu air bisa lebih dari satu



Gambar 9. Pengujian melihat *notification*

Pada Gambar 9, halaman *notification* saat *user* masuk ke dalam menu *notification* maka akan tampil pintu air yang *user* pilih di menu pintu air. Di *notification* ini *user* akan mengetahui kondisi dan siaga pintu air. Pada *panel notification* Info pintu air *user* akan mendapatkan info tanpa harus masuk ke aplikasi (Gambar 10).

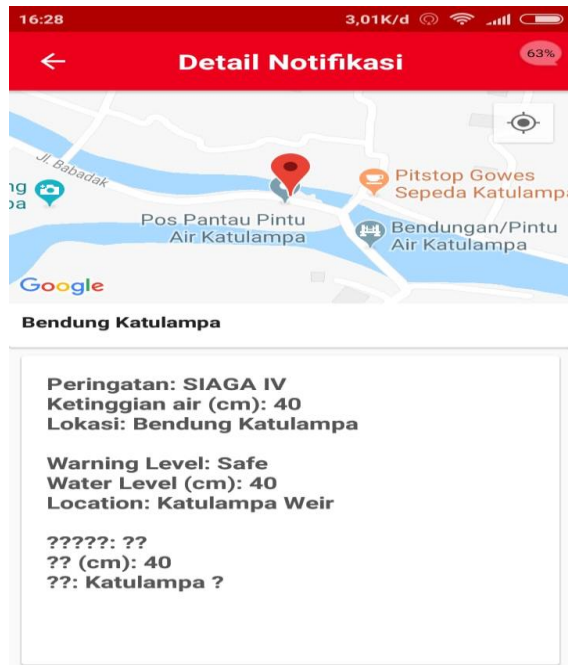


Gambar 10. Tampilan panel *notification*



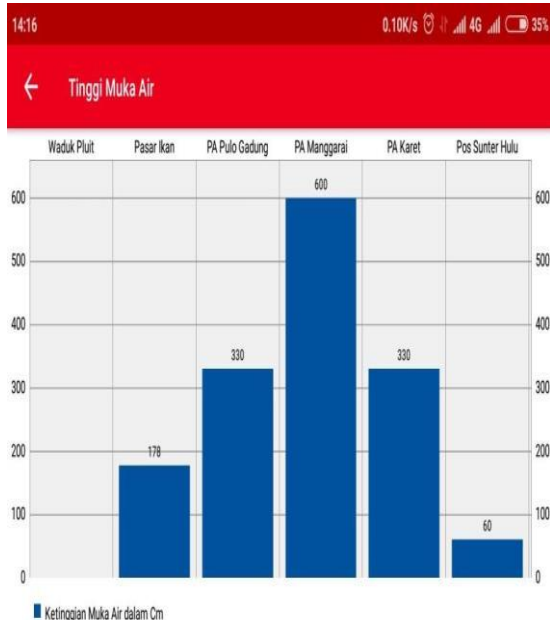
Gambar 11. Tampilan info cuaca

Gambar 11 adalah tampilan info cuaca, ada beberapa informasi seperti *Temp* yang artinya *temperature* cuaca dan *humidity* (kelembapan udara) data yang diambil dari BMKG. *User* akan bisa melihat dan mengetahui cuaca Jabodetabek BPBD Jakarta.



Gambar 12. Tampilan *notification* tinggi muka air

Pada Gambar 12 tampilan *notification* tinggi muka air yang *user* pilih Bendung Katulampa ketinggian air diketahui 40 cm.



Gambar 13. Tampilan menu tinggi muka air

Pada Gambar 13 tampilan menu tinggi muka air, *user* bisa melihat tinggi muka air seluruh Jabodetabek.

3.5 Operation and Maintenance

Berdasarkan dari hasil pengujian semua telah terpenuhi, tahap *operation* dan *maintenance* akan dilakukan oleh pihak BPPD terkait jika memang akan ada perubahan aplikasi.

Tabel 1. *User acceptance testing*.

User Acceptance Testing		
Tanggal Uji : 9 Juli 2018		
Nama	Uji	Hasil
Bapak Hadi (Staff BPBD)	Login & Menggunakan aplikasi	OK
Bapak Nazmul	Login & Menggunakan aplikasi	OK

Dalam Tabel 1 terlihat hasil pengujian aplikasi untuk *user* sebagai bagian dari tahap *operation*.

Dari sistem aplikasi yang telah dibuat dalam penelitian ini:

- A. Untuk *push notification* yang dikirim ke *mobile user* dan berbentuk *panel notification* telah dapat dilakukan, dapat dilihat Gambar 9 dan Gambar 10.
- B. Mendapatkan data secara *real time* dan menggunakan API untuk menarik data (Gambar 7).
- C. Pengiriman informasi secara *real time* ke *user*, juga berhasil dibuat (Gambar 12).
- D. Implementasi *push notification* menggunakan *tool Firebase Cloud Messaging* (Gambar 7).
- E. Aplikasi telah berhasil memberikan informasi ketinggian air (Gambar 12), daftar pintu air (Gambar 9), dan info cuaca (Gambar 11)
- F. Empat keefektifan sistem *push notification* monitoring, telah terimplementasi pada keseluruhan aplikasi dan hasil *user testing* pada Tabel 1.

IV. PENUTUP

Penelitian ini dapat disimpulkan bahwa dengan adanya *firebase cloud messaging* dapat dirancang pada aplikasi *push notification monitoring system* pintu air berbasis *android* ini dapat digunakan oleh *user* (masyarakat) dan informasi ditampilkan dalam bentuk *push notification* secara *mobile* ke *user*, dan aplikasi ini dapat digunakan sebagai media *monitoring* bagi BPPD dan pihak terkait.

DAFTAR PUSTAKA

- [1] BNPB. (n.d.). Statistik bencana Indonesia. " Retrieved March 9, 2018, from <http://dibi.bnpb.go.id/dibi/>
- [2] C. S. Samosir, W. Soetopo & E. Yuliani, "Optimasi Pola Operasi Waduk Untuk Memenuhi Kebutuhan Energi Pembangkit Listrik Tenaga Air (Studi Kasus Waduk Wonogiri). *Jurnal Teknik Pengairan*, 6(1), 108–115, 2015
- [3] T. Kartini & S. Permana, "Analisis Operasional Waduk Ir.H.Djuanda". *Jurnal Konstruksi Sekolah Tinggi Teknologi Garut*, (1), 13–24, 2016
- [4] A. Rosyidie, "Banjir : Fakta dan dampaknya, serta pengaruh dari perubahan guna lahan", *Jurnal Perencanaan Wilayah Dan Kota*, 24(3), 241–249, 2013
- [5] M. Corps, "Monitoring and Evaluation Guidebook, 2005. Retrieved from url: <http://www.mercycorps.org/sites/default/files/file1157150018.pdf>
- [6] F. Prayoga, "Perancangan Prototype Aplikasi Pengumuman Kelas Menggunakan Teknologi Firebase Cloud Message Pada Android", 2016
- [7] R. Sumardi, "Aplikasi Mobile Notification Informasi Perkuliahan Berbasis", 2017
- [8] A. Hisham, "Development of a Location Based Approaching Notification System Using Android Platform. Pakistan", 2017
- [9] F. D. Nurzam, I. N. Fajri & D. Prabowo, " Rancang Bangun Aplikasi Media Laporan Aspirasi Dengan Firebase Cloud Messaging", 37–42, 2017
- [10] M. Siddik & A. Nasution, "Perancangan Aplikasi Push Notification Berbasis Android", *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, Vol. IV No. 2, hlm. 149 – 154, Jun 2018, DOI: 10.33330/jurteksi.v4i2.56, <http://jurnal.stmikroyal.ac.id/index.php/jurteksi>
- [11] J. Setiawan, E. Kristianto & Fredicia "Implementasi Push Notification Pada Informasi Perkuliahan Dan Kegiatan Mahasiswa Berbasis Android, " *J. Tek. dan Ilmu Komput.*, vol. 4, no. 14, pp. 211–219, 2015
- [12] D. E. Kurniawan, "Push Notification System Pada Prototype Kendali Listrik Rumah" *J. Comput. Eng. Syst. Sci.*, vol. 2, no. 1, pp. 89–92, 2017, DOI: <https://doi.org/10.24114/cess.v2i2.6463>
- [13] E. Isikligil. S. Samakay & D. Kilinç, "A Prototype Framework for High Performance Push Notifications," *Int. J. Comput. Appl.*, vol. 166, no. 10, pp. 8–11, 2017, DOI: 10.5120/ijca2017914122
- [14] Supriyanta, W. Pudji & S. Bekti, "Aplikasi Konversi Suara Ke Teks Berbasis Android Menggunakan Google Speech Api, " 2014
- [15] H. A. Khalilulah, "Implementasi Speech Recognition pada Aplikasi Penerjemah Idiom Bahasa Inggris ke Bahasa Indonesia Berbasis Android, " 2016
- [16] P. Zamrony & Juhara, "Panduan Lengkap Pemrograman Android", Andi Publisher, Yogyakarta, 2016
- [17] A. Laysha (n.d.). "Push-to-succeed: a look into Azure Notification Hubs for business." Retrieved from <https://www.scnsoft.com/blog/pushtosucceed-a-look-intoazure-notification-hubs-for-business>
- [18] R. Hilmawan & I. Iskandar, " Implementasi Service-Oriented Architecture Menggunakan Teknologi Web Service Untuk Aplikasi Pendaftaran Pdta Se- Kota Pekanbaru, " *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, 2014
- [19] N. Srivastava, U. Shree, N. R. Chauhan & D. K. Tiwari, "Firebase Cloud Messaging (Android)," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 6, no. 9, pp. 11–18, 2017.
- [20] B. Sidik, "Pemrograman Web dengan PHP. Bandung, 2012. Retrieved from eprints.akakom.ac.id/503/15/115410046_DAFTAR_PUSTAKA.pdf
- [21] Firebase. (n.d.). "Firebase cloud messaging". Retrieved March 15, 2018, from <https://firebase.google.com/docs/cloud-messaging/?hl=id>
- [22] I. Sommerville, " Software Engineering 10th Edition. (In Software Engineering, Ed.), 2015
- [23] A. Mujilan, "*Analisis dan Perancangan Sistem*. Universitas Widya Mandala, 2013.

Hak Cipta

Semua naskah yang tidak diterbitkan, dapat dikirimkan di tempat lain. Penulis bertanggung jawab atas ijin publikasi atau pengakuan gambar, tabel dan bilangan dalam naskah yang dikirimkannya. Naskah bukanlah naskah jiplakan dan tidak melanggar hak-hak lain dari pihak ketiga. Penulis setuju bahwa keputusan untuk menerbitkan atau tidak menerbitkan naskah dalam jurnal yang dikirimkan penulis, adalah sepenuhnya hak Pengelola. Sebelum penerimaan terakhir naskah, penulis diharuskan menegaskan secara tertulis, bahwa tulisan yang dikirimkan merupakan hak cipta penulis dan menugaskan hak cipta ini pada pengelola.