

IMPLEMENTASI NETWORK NOTIFICATION SYSTEM DENGAN MENGGUNAKAN TEKNOLOGI FIREBASE CLOUD MESSAGING (FCM) BERBASIS ANDROID

Dessyanto Boedi Prasetyo⁽¹⁾, Rizki Inka Miftah⁽²⁾, Rifki Indra Perwira⁽³⁾

Program Studi Teknik Informatika, Fakultas Teknik Industri, UPN "Veteran" Yogyakarta
Jalan Babarsari 2, Kec. Depok, Kabupaten Sleman, DIY 55281, Indonesia
e-mail : dess95@gmail.com⁽¹⁾, rizkiinkamiftah@gmail.com⁽²⁾, rifki@upnyk.ac.id⁽³⁾

Abstract

A quality internet networks became an important factor in supporting the activities of lecture at Universitas Pembangunan Nasional "Veteran" Yogyakarta, so that required the existence of oversight on a regular basis to ensure the sustainability of the network the Internet is good. As a network administrator would certainly not be in the workplace, so that required a system that supports the performance of the network administrator at the time of being out of work. On the research this time made an application that allows a network administrator can receive a notification message containing the incident (event) that occur on mikrotik router devices with parameters exist. The design of the system aided by Firebase Cloud Messaging (FCM) technology that acts as a web service. The sent messages can be stored in a database which can be seen on the admin web page. System development methods used namely GRAPPLE which includes stages of gathering requirements, analysis, design, development, and deployment of the system. In terms of the development of this system required configuration of mikrotik router device to connect two mikrotik router to PC Servers are available. The results of this research that is has successfully build the application Network Notification System by using Firebase Cloud Messaging (FCM) technology based on android. For testing step that use functionality test of the system by performing a simulation on every function of the system that was build and it was concluded that the function of the system can be run well.

Keywords : *network, router mikrotik, firebase cloud messaging, android.*

Jaringan internet yang berkualitas menjadi suatu faktor penting dalam mendukung kegiatan perkuliahan pada Universitas Pembangunan Nasional "Veteran" Yogyakarta sehingga diperlukan adanya pengawasan secara teratur untuk menjamin keberlangsungan jaringan internet yang baik. Sebagai seorang administrator jaringan tentu saja tidak selalu berada di tempat kerjanya sehingga dibutuhkan sebuah sistem yang mendukung kinerja administrator jaringan di saat berada di luar tempat kerjanya. Pada penelitian kali ini dibuatlah sebuah aplikasi yang memungkinkan seorang *administrator* jaringan dapat menerima pesan notifikasi berisikan kejadian (*event*) yang terjadi pada perangkat *router mikrotik* dengan parameter-parameter yang ada. Perancangan sistem kali ini dibantu dengan teknologi *Firestore Cloud Messaging (FCM)* yang berfungsi sebagai *web service*. Pesan yang telah terkirim dapat tersimpan dalam *database* yang dapat dilihat pada halaman *web admin*. Metode pengembangan sistem yang digunakan yaitu *GRAPPLE* yang meliputi tahapan pengumpulan kebutuhan, analisis, perancangan, pengembangan, dan penyebaran sistem. Dalam hal pengembangan sistem ini dibutuhkan konfigurasi *device router mikrotik* agar dapat terhubung dua buah *router mikrotik* ke *PC Server* yang tersedia. Hasil dari penelitian ini yaitu berhasil membangun aplikasi *Network Notification System* dengan menggunakan teknologi *Firestore Cloud Messaging (FCM)* berbasis *android*. Untuk pengujian yang dilakukan menggunakan pengujian fungsionalitas dari sistem dengan melakukan simulasi pada setiap fungsi dari sistem yang dibuat dan disimpulkan bahwa fungsi sistem dapat berjalan dengan baik.

Kata Kunci : *jaringan, router mikrotik, firebase cloud messaging, android.*

1. PENDAHULUAN

Universitas Pembangunan Nasional "Veteran" Yogyakarta memiliki jaringan internet yang berguna sebagai sarana komunikasi antar civitas akademika Universitas Pembangunan Nasional "Veteran" Yogyakarta. Bagi mahasiswa tersendiri internet kampus berguna untuk fasilitas pendukung perkuliahan semacam mengerjakan tugas dan kegiatan praktikum.

Namun sering kali terjadi permasalahan pada jaringan internet, antara lain koneksi data yang dikirim lambat atau bahkan tidak sampai ke tujuan. Komunikasi sering mengalami *timeout*, *data traffic* padat, hingga masalah keamanan yang dapat menyebabkan ketidaknyamanan bagi pengguna. Jaringan internet memerlukan sebuah alat yang berfungsi sebagai pengatur jalur lalu-lintas data sehingga tepat pada sasarannya yang disebut dengan *router*. Diyakini bahwa *router* mampu menjawab tantangan daripada permasalahan jaringan komputer itu sendiri dengan didukungnya fasilitas-fasilitas yang dimiliki *router*. Perangkat *router* yang didukung dengan fasilitas-fasilitas yang dimiliki dapat mengatasi permasalahan yang terjadi pada jaringan. Dalam hal ini *router* yang akan dijadikan sebagai bahan penelitian adalah *router Mikrotik* yang tersedia pada Universitas Pembangunan Nasional “Veteran” Yogyakarta.

Permasalahan yang akan dihadapi oleh teknisi/*administrator* jaringan, dimana keadaan jaringan harus selalu terpantau agar situasi jaringan di lingkungan Universitas Pembangunan Nasional “Veteran” Yogyakarta dapat terkendali dengan baik. Seorang teknisi/*administrator* tentu tidak akan selalu berada di ruang kantor UPT TIK (Teknologi Informasi dan Komunikasi) Universitas Pembangunan Nasional “Veteran” Yogyakarta untuk memantau kondisi *router mikrotik* secara *real-time*. Dengan ini dibutuhkan suatu sistem yang dapat membantu kinerja teknisi/*administrator* jaringan untuk memantau kondisi *router mikrotik* disaat mereka berada di luar kantor atau sedang dalam kepentingan tertentu, maka dari itu pada penelitian ini akan mengembangkan sistem pengiriman notifikasi berupa info aktifitas-aktifitas yang terjadi pada *router mikrotik* berisikan parameter seperti *limit bandwidth*, perubahan *IP*, *reboot device*, dan pesan *error*. Pada pesan *error* ini akan berguna juga untuk mengantisipasi masalah keamanan, semacam terdapat kegiatan ilegal dengan seseorang melakukan proses *login* ke *router mikrotik* gagal.

Pada penelitian sebelumnya (Himmi and Abidin, 2015) merancang aplikasi monitoring jaringan berbasis web yang dibantu dengan aplikasi *putty* milik *mikrotik*, dan menggunakan protokol *SNMP* yang berisikan info-info yang dibutuhkan dalam *monitoring* jaringan dan akan dikirimkan melalui *email* kepada *administrator* jaringan di Dinas Pendidikan Kota Blitar. Kemudian terdapat penelitian serupa (KUSUMA, 2015a) melakukan perancangan sistem *monitoring* jaringan berbasis *SNMP* dengan memanfaatkan *syslog* dari aplikasi *monitoring The Dude*. Dari sistem *monitoring* ini, dapat diketahui mengenai informasi seperti status *device*, suhu dan *resource cpu* dari *device*, trafik data dari *device*.

Pada penelitian ini menggunakan teknologi *Firebase Cloud Messaging (FCM)* yang berfungsi sebagai *web service* penyedia layanan *push notification* atau pesan notifikasi yang berisikan aktifitas-aktifitas yang terjadi pada *router mikrotik*. *FCM* ini juga berperan untuk menjembatani antara *server* dengan perangkat *Android* agar dapat terjadi aktifitas pengiriman pesan (*push notification*). Dari penjabaran permasalahan di atas, aplikasi *Network Notification System* berbasis *Android* dengan menggunakan teknologi *Firebase Cloud Messaging (FCM)* dapat menjadi alternatif untuk membantu kesigapan kinerja teknisi/*administrator* jaringan untuk memberikan pelayanan terbaik saat terjadi permasalahan jaringan internet di lingkungan civitas akademika Universitas Pembangunan Nasional “Veteran” Yogyakarta.

Berdasarkan observasi dan wawancara yang dilakukan terhadap pihak *administrator* jaringan dari UPT TIK (Teknologi Informasi dan Komunikasi) Universitas Pembangunan Nasional “Veteran” Yogyakarta terdapat rumusan masalah sebagai berikut:

- a) Adanya laporan dari satuan kerja mengenai lambatnya proses lambatnya penanganan permasalahan jaringan.
- b) Tingkat mobilitas dari seorang *administrator* jaringan yang cukup tinggi dikarenakan mencakup dua kampus yang berbeda lokasi.
- c) Tingkat *downtime* jaringan yang masih berjalan cukup panjang.
- d) Belum adanya aplikasi *mobile* yang memungkinkan untuk memantau kondisi jaringan di Universitas Pembangunan Nasional “Veteran” Yogyakarta.

Dalam proses penyusunan dan perancangan sistem, terdapat beberapa batasan-batasan penelitian sebagai berikut:

- a) Ruang lingkup penelitian dilakukan pada *router mikrotik* yang menggunakan versi *Operating System* yang sama dengan yang digunakan pada Universitas Pembangunan Nasional "Veteran" Yogyakarta.
- b) Parameter-parameter seperti informasi mengenai *IP router* beserta *traffic bandwidth router mikrotik* yang melampaui batas atas, saat terjadi galat (*error*) pada *router mikrotik*, ketika adanya penambahan atau pengurangan *device*, serta ketika *device router mikrotik reboot*.
- c) Menggunakan bahasa pemrograman *Java* dan *PHP* serta menggunakan teknologi *Firebase Cloud Messaging (FCM)*.

Adapun tujuan dari dilakukannya penelitian ini adalah untuk mengembangkan aplikasi *Network Notification System* menggunakan teknologi *Firebase Cloud Messaging (FCM)* berbasis *Android* yang dapat melakukan pemantauan kondisi jaringan secara *real time* dengan mengirimkan pesan notifikasi kepada perangkat *Android* seorang *administrator* jaringan yang berisikan empat parameter pesan yaitu *bandwidth*, *error*, perubahan *IP*, dan *reboot device*.

2. TINJAUAN LITERATURE

Pada penelitian yang dilakukan oleh G.J. Ohara (2005) yang berjudul "*Aplikasi Sistem Monitoring Berbasis Web untuk Open Cluster*" mengenai perancangan aplikasi pemantauan sistem *Open Cluster* milik LIPI. *Open cluster* yang dikembangkan oleh LIPI adalah alternatif pembentukan komputasi paralel fisik berbiaya murah yang dibentuk dari sejumlah komputer dan dibuka untuk publik. Pada umumnya pemakaian sebuah *cluster* cenderung hanya untuk jaringan tertutup dengan akses dari luar yang terbatas. Selain itu dibutuhkan juga manajemen untuk pengaturan *node*, *item monitoring*, *power control* dan juga pengaturan *user*. Untuk pengumpulan data digunakan *SNMP (Simple Network Management Protocol)* dan tools lain yang mendukung (Ohara, 2005).

Penelitian lain terkait dengan monitoring jaringan yaitu yang dilakukan oleh R. Pradikta, A. Affandi, dan E. Setijadi (2013) yang berjudul "*Rancang Bangun Aplikasi Monitoring Jaringan dengan Menggunakan Simple Network Management Protocol*" membahas mengenai perancangan dan pembuatan aplikasi *monitoring* jaringan yang dapat digunakan sebagai perantara untuk mengambil dan mengolah nilai *SNMP* sekaligus terdapat sistem penyimpanan atau *database* sehingga dapat ditampilkan laporan informasi tentang kondisi jaringan yang meliputi *availability* perangkat dan trafik pada *transport TCP* (Pradikta et al., 2013).

Pada penelitian terkait yang dilakukan oleh Febrian Prayoga (2016) yang berjudul "*Perancangan Prototype Aplikasi Pengumuman Kelas Menggunakan Teknologi Firebase Cloud Message Pada Android*" membahas pembuatan aplikasi untuk pengumuman kelas secara *real time* dengan studi kasus Fakultas Teknologi Dan Informasi Universitas Kristen Satya Wacana Salatiga dengan metode *Waterfall*. Aplikasi ini nantinya akan digunakan untuk mengirimkan pengumuman kepada mahasiswa-mahasiswa yang mengambil kelas tertentu dalam bentuk *push notification* menggunakan teknologi *FCM* (Prayoga, 2016).

National Institute of Standards and Technology (NIST), Information Technology Laboratory memberikan dua buah catatan mengenai pengertian komputasi awan. Pertama, komputasi awan masih merupakan paradigma yang berkembang. Definisi, kasus penggunaan, teknologi yang mendasari, masalah, risiko, dan manfaat akan terus disempurnakan melalui perdebatan baik oleh sektor publik maupun swasta. Definisi, atribut, dan karakteristik akan berkembang dan berubah dari waktu ke waktu. Kedua, industri komputasi awan merupakan ekosistem besar dengan banyak model, vendor, dan pangsa pasar. Definisi ini mencoba untuk mencakup semua pendekatan berbagai awan (Peter Mell, 2011).

Dari kedua catatan tersebut *NIST* memberikan definisi komputasi awan adalah model untuk memungkinkan kenyamanan, *on-demand* akses jaringan untuk memanfaatkan bersama suatu sumberdaya komputasi yang terkonfigurasi (misalnya, jaringan, *server*, penyimpanan, aplikasi, dan layanan) yang dapat secara cepat diberikan dan dirilis dengan upaya manajemen yang minimal atau interaksi penyedia layanan. Model komputasi awan mendorong ketersediaan dan terdiri dari lima karakteristik, tiga model layanan, dan empat model penyebaran.

Firestore Cloud Messaging dirancang untuk menyediakan koneksi ke perangkat *Android* melalui pesan notifikasi. *FCM* ini dapat diandalkan, dengan 98% akurasi pesan dikirim ke perangkat yang terhubung dengan kecepatan 500ms atau kurang, dan juga memiliki skala yang besar, dengan infrastruktur yang menghasilkan lebih dari satu triliun pesan setiap minggunya. *Firestore Cloud Messaging* menawarkan berbagai metode untuk memberikan kontrol yang mudah (Moroney, 2017).

3. METODE PENELITIAN

3.1 Metodologi Pengembangan Sistem

Metodologi yang digunakan pada proses pengembangan sistem ini yaitu *Guide Lines for Rappid Application Engineering (GRAPPLE)*. *GRAPPLE* menurut (Schmuller, 1999) merupakan metodologi yang fleksibel dan memberikan panduan yang jelas dalam proses pengembangan sistem. Metodologi *Guide Lines for Rappid Application Engineering (GRAPPLE)* memiliki lima tahapan yaitu:

- a) Pengumpulan kebutuhan (*Requirement gathering*)
- b) Analisis (*Analysis*)
- c) Perancangan (*Design*)
- d) Pengembangan (*Development*)
- e) Penyebaran (*Deployment*)

Pada bagian ini akan membahas tiga tahapan yang ada didalam metodologi *Guide Lines for Rappid Application Engineering (GRAPPLE)* yaitu pengumpulan kebutuhan (*requirement gathering*), analisis (*analysis*), dan perancangan (*design*).

a) Pengumpulan Kebutuhan (*Requirement Gathering*)

Pada tahap ini dilakukan proses identifikasi dan merencanakan teknologi yang digunakan untuk pembuatan aplikasi dan perancangan *server*. Proses identifikasi dilakukan dengan melakukan wawancara kepada pihak *administrator* jaringan Universitas Pembangunan Nasional "Veteran" Yogyakarta Tahap perencanaan sangatlah penting untuk proses jalannya pengembangan aplikasi *Network Notification System* menggunakan teknologi *Firestore Cloud Messaging (FCM)* berbasis *Android* supaya tidak berjalan secara acak (*random*) dan ditentukan empat parameter-parameter yang dibutuhkan yaitu *limit bandwidth*, perubahan *IP*, informasi *error*, dan *device reboot*.

b) Hasil Analisis yang Dilakukan

Pada tahap ini dilakukan analisis yang diperlukan dalam membuat aplikasi *Network Notification System* menggunakan teknologi *Firestore Cloud Messaging (FCM)* berbasis *Android*. Dalam tahap ini analisis dilakukan dengan memetakan hasil observasi dan wawancara dengan pihak *administrator* jaringan Universitas Pembangunan Nasional "Veteran" Yogyakarta yang didapatkan informasi bahwa selama ini banyak laporan masuk dari satuan kerja mengenai lambatnya proses penanganan jaringan dikarenakan tingkat mobilitas yang tinggi dan keterbatasan jumlah *administrator* menyebabkan kegiatan pemantauan jaringan tidak berjalan secara *real time*, maka dari itu dibutuhkan sebuah terobosan aplikasi yang dapat memberikan pesan notifikasi mengenai kegiatan (*event*) yang berhubungan dengan *router mikrotik* dan berisikan empat parameter pesan yaitu *bandwidth*, *error*, perubahan *IP*, dan *reboot device*.

c) Perancangan (*Design*)

Pada tahap desain perancangan merupakan hasil dari proses analisis yang digambarkan melalui perancangan *prototype user interface* berupa mockup desain *web admin* beserta desain aplikasi *Android* yang akan dibuat.

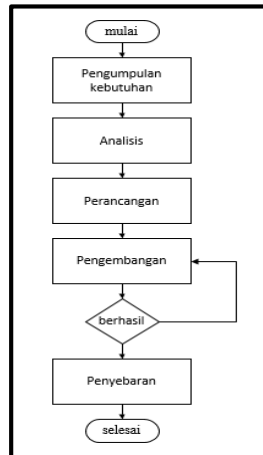
d) Pengembangan (*Development*)

Pada tahap ini proses pengembangan sistem (*system development*) dilakukan dengan proses konfigurasi pendukung aplikasi yaitu *server*, *mikrotik*, dan *FCM*. Setelah proses konfigurasi pendukung berhasil dilakukan, mulai dengan proses *coding* untuk membuat aplikasi *Android*

dengan aplikasi *Android studio*, kemudian dibuat *web admin* dengan bahasa *PHP* menggunakan aplikasi *sublime text*.

e) Penyebaran (*Deployment*)

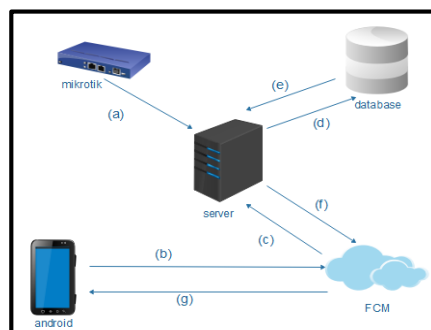
Pada tahap penyebaran ini akan dilakukan pendistribusian aplikasi kepada pengguna dalam hal ini administrator jaringan UPN “Veteran” Yogyakarta.



Gambar 1. *Flowchart* proses pengembangan sistem

3.2 Arsitektur Sistem

Pada bagian ini akan dijelaskan mengenai arsitektur dari proses jalannya sistem seperti gambar berikut.

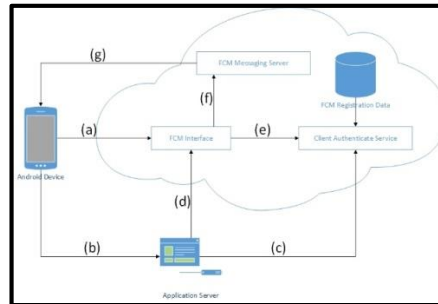


Gambar 2. Arsitektur sistem

- Mikrotik* akan terus menerus mengirimkan info *log* ke *server* untuk selanjutnya di lakukan proses *parsing* sesuai parameter yang dibutuhkan.
- Ketika aplikasi *user* pada perangkat *Android* dijalankan dan melakukan registrasi, *device* akan mendapatkan identitas berupa *token* dari *FCM*.
- Firestore Cloud Messaging (FCM)* mengirim data *user* yang telah melakukan registrasi beserta *token* devicenya.
- Server* menjalankan fungsi *insert data* untuk memasukkan data ke dalam *database*. Pada tahap ini juga *server* akan melakukan proses *parsing* data *log mikrotik* dan mengirimnya ke dalam *database*.
- Server* dapat memperoleh data pesan maupun *users* untuk dikelola maupun pesan untuk dikirimkan menuju *user*.
- Server* mengirimkan pesan kepada *user* melalui *FCM*.
- Firestore Cloud Messaging (FCM)* mengirimkan pesan *push notification* ke perangkat *Android*.

3.3 Arsitektur Cloud

Jika sebelumnya telah dijelaskan mengenai arsitektur sistem yang berjalan, pada bagaian ini akan dijelaskan mengenai arsitektur daripada *Firebase Cloud Messaging (FCM)* pada khususnya seperti pada gambar berikut.



Gambar 3. Arsitektur *cloud*

- Android device* akan teregistrasi ke layanan *Firebase Cloud Messaging (FCM)* dan menerima ID register.
- ID register dikirimkan menuju *application server*.
- Application server* meminta autentikasi ke *Client Authenticate Service* dari *Firebase Cloud Messaging (FCM)* untuk mendapatkan *token*.
- Pesan dikirimkan ke perangkat *android* berdasarkan *token* yang didapatkan sebelumnya.
- Token* dan ID register diverifikasi kecocokannya.
- Pesan akan diarahkan untuk segera dikirim ke perangkat *android*.
- Pesan berhasil terkirim sebagai notifikasi.

4. HASIL DAN PEMBAHASAN

Uji Coba Aplikasi Network Notification System

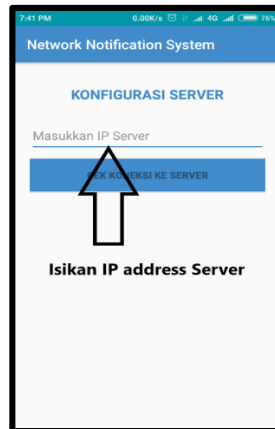
Pada sub bahasan ini akan dilakukan uji coba fungsionalitas dari aplikasi *Network Notification System*. Uji coba fungsionalitas berfungsi untuk menguji aplikasi ini berjalan dengan baik atau tidak. Kemudian dilakukan evaluasi terhadap kesalahan atau kekurangan pada aplikasi. Untuk menguji fungsionalitas dari aplikasi ini dilakukan sesuai langkah-langkah berikut.

- Jalankan aplikasi *Network Notification System* di *android*, klik menu *register device* untuk selanjutnya masuk ke halaman koneksi *server*.



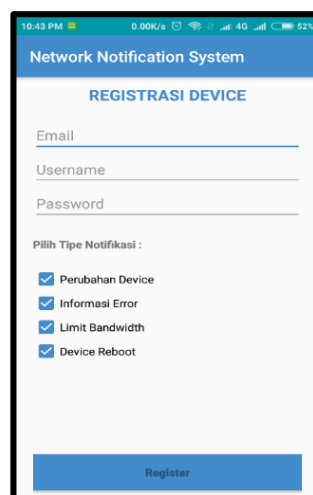
Gambar 4. Halaman awal aplikasi

- Setelah masuk ke halaman koneksi *server*, masukkan *IP address* dari *server*.



Gambar 5. Halaman koneksi server

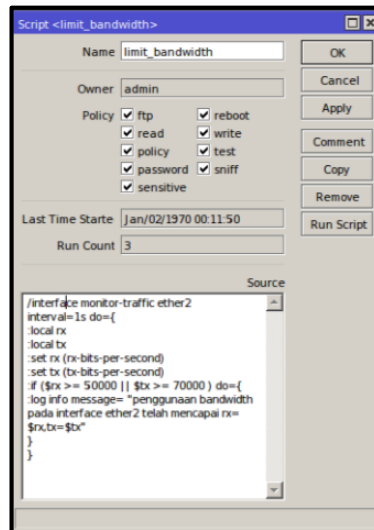
3. Jika berhasil *connect*, selanjutnya berganti menuju halaman registrasi, isikan data berupa *email*, *username*, *password*, dan kebutuhan notifikasi yang diinginkan berupa empat buah parameter.



Gambar 6. Halaman registrasi

4. Jika berhasil maka muncul halaman sukses registrasi, namun jika belum berhasil akan muncul peringatan.
5. Data hasil registrasi dapat dicek pada *database MySQL* maupun halaman *web admin*.
6. Pengujian proses mengirim *log file* dari *router mikrotik* menuju ke *database MySQL* yang pertama yaitu untuk menguji pengiriman *log file* parameter *limit bandwidth*, langkah yang dilakukan untuk mengkonfigurasi *router mikrotik* pada parameter *Limit Bandwidth* dengan menambahkan *actions* dan *rules* pada menu *system* → *logging* untuk menghubungkan *file log* dari *router mikrotik* menuju ke server yang telah dibuat pada tahap sebelumnya. Pada menu *actions* ditambahkan fitur dengan nama “notif”, *type* diisi dengan *remote*, dan *remote address* diisi dengan alamat *IP* dari server.

Selanjutnya memasuki *tab rules*, untuk menambahkan *log action* yang telah dibuat sebelumnya ke menu *rules* ini, tambahkan sebuah *script* pada menu *system* → *scripts* yang berisi *script limit bandwidth*, langkah ini dilakukan dikarenakan pada *system logging mikrotik* belum terdapat *log* berisi peringatan batasan (*limit*) dari *bandwidth* yang telah digunakan untuk kemudian dilakukan perintah *run script*. Perintah yang ditambahkan sebagai berikut.



Gambar 7. Menambahkan script limit bandwidth

Kemudian pada prosesnya jika terdapat keadaan kapasitas *bandwidth* mencapai batas yang telah ditentukan, maka akan tampil pada *log file* yang nantinya akan dikirimkan ke *device android*.

- Untuk parameter kedua yaitu perubahan *IP* langkah yang dilakukan untuk mengkonfigurasi *router mikrotik* untuk parameter Perubahan *IP* yaitu dengan menambahkan *actions* dan *rules* pada menu *system* → *logging* untuk menghubungkan *file log* dari *router mikrotik* menuju ke server yang telah dibuat pada tahap sebelumnya. Pada menu *actions* ditambahkan fitur dengan nama "notif", *type* diisi dengan *remote*, dan *remote address* diisi dengan alamat *IP* dari server.

Selanjutnya memasuki *tab rules*, untuk menambahkan *log action* yang telah dibuat sebelumnya ke menu *rules* ini. Langkah selanjutnya yaitu untuk mengecek jika adanya perubahan *IP* yang terjadi dapat dicek pada menu *IP* → *IP Address*. Jika terjadi adanya perubahan *IP* oleh *admin* ataupun *user* lain dapat terlihat pada *log file* yang nantinya akan dikirimkan menuju *device android*.

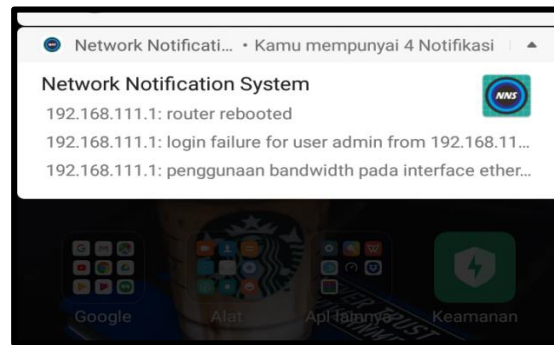
- Parameter ketiga yaitu informasi *error* langkah yang dilakukan untuk mengkonfigurasi *router mikrotik* untuk parameter informasi *error* yaitu dengan menambahkan *actions* dan *rules* pada menu *system* → *logging* untuk menghubungkan *file log* dari *router mikrotik* menuju ke server yang telah dibuat pada tahap sebelumnya. Pada menu *actions* ditambahkan fitur dengan nama "notif", *type* diisi dengan *remote*, dan *remote address* diisi dengan alamat *IP* dari server.

Selanjutnya memasuki *tab rules*, untuk menambahkan *log action* yang telah dibuat sebelumnya ke menu *rules* ini. Jika terjadi adanya informasi *error* dapat terlihat pada *log file* yang nantinya akan dikirimkan menuju *device android*.

- Parameter keempat yaitu *reboot device* langkah yang dilakukan untuk mengkonfigurasi *router mikrotik* untuk parameter *device reboot* yaitu dengan menambahkan *actions* dan *rules* pada menu *system* → *logging* untuk menghubungkan *file log* dari *router mikrotik* menuju ke server yang telah dibuat pada tahap sebelumnya. Pada menu *actions* ditambahkan fitur dengan nama "notif", *type* diisi dengan *remote*, dan *remote address* diisi dengan alamat *IP* dari server.

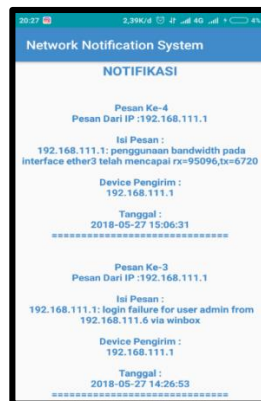
Selanjutnya memasuki *tab rules*, untuk menambahkan *log action* yang telah dibuat sebelumnya ke menu *rules* ini. Jika terjadi *device mikrotik* melakukan *reboot* dapat terlihat pada *log file* yang nantinya akan dikirimkan menuju *device android*.

10. Selanjutnya dilakukan proses pengiriman data dari *database MySQL* menuju ke perangkat *android*. Jadi, pada proses ini *server* akan terus menjalankan sebuah URL (<http://localhost/mikrotik/pesan/sendpesan>) setiap menitnya dengan bantuan *crontab* untuk mengirim pesan dalam bentuk *array*. Dalam fungsi *sendpesan* terdapat pengecekan kondisi jika status pesan terkirim adalah "0" maka pesan tersebut akan dikirimkan dan diubah statusnya menjadi "1" agar tidak terkirimkan kembali.
11. Pesan notifikasi akan muncul pada layar *popup android* berupa jumlah pesan dan *preview* pesan yang akan ditampilkan.

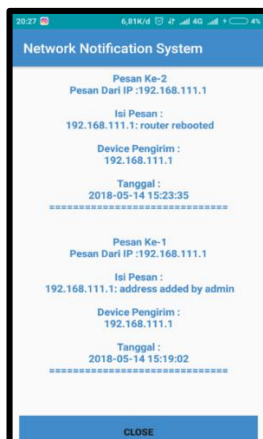


Gambar 8. *Popup notifikasi pesan android*

12. Setelah pesan notifikasi diklik, maka akan muncul halaman *display* pesan berisi data judul pesan, isi pesan, *IP* pengirim, dan tanggal kejadian (*event*) pada *router mikrotik* terjadi. Pesan yang masuk diurutkan berdasarkan pesan terbaru yang berada di posisi paling atas, dan pesan-pesan yang telah terkirimkan akan masuk pada halaman *web admin* yang telah dibahas sebelumnya untuk menjadi bahan evaluasi. Gambar 9 merupakan tampilan pesan dari parameter *limit bandwidth* dan parameter informasi *error*, sedangkan Gambar 10 merupakan tampilan pesan dari parameter perubahan *IP* dan *reboot device* pada perangkat *android*.



Gambar 9. Halaman tampil pesan 1



Gambar 10. Halaman tampil pesan 2

Dari rangkaian percobaan pengujian fungsionalitas yang telah dilakukan, fungsi-fungsi yang terdapat pada aplikasi *Network Notification System* maupun halaman *web admin* berjalan dengan baik, mulai dari proses registrasi, pengiriman *log file* dari empat buah parameter pada perangkat *router mikrotik* menuju ke dalam *database* hingga sukses mengirimkan pesan notifikasi menuju perangkat *android*.

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut.

- Durasi *downtime* jaringan dapat tereduksi sehingga meringankan kinerja *administrator* dan jaringan di lingkungan dapat berjalan lebih optimal.
- Sistem dapat bekerja secara maksimal pada *device router Mikrotik A* (RB 941 lite) yang dapat mengirimkan keseluruhan parameter.
- Terdapat kendala pada parameter *device reboot* pada *Mikrotik B* (RB 750) dikarenakan terdapat perbedaan karakteristik beberapa *device Mikrotik* dalam hal pencatatan *log activity* yang dilakukan sebelum *device* menyala.
- Sistem yang dibuat hanya memungkinkan pengiriman *log activity* yang dicatat setelah *device* menyala (*link up*) dikarenakan membutuhkan adanya daya.
- Setelah dilakukan percobaan dengan membandingkan beberapa *device router mikrotik* berbeda disimpulkan bahwa mayoritas karakteristik pencatatan *log activity* mengenai *reboot device* dilakukan sebelum *device* menyala (*link up*).

Beberapa saran dan rekomendasi yang berguna untuk pengembangan aplikasi *Network Notification System* selanjutnya sebagai berikut.

- Dapat terintegrasi dengan perangkat jaringan lain selain *mikrotik* sehingga sistem dapat digunakan untuk jangkauan sistem jaringan yang lebih luas.
- Dapat diintegrasikan pada fungsi *radius* dalam hal *user management* pada *router mikrotik* yang dapat menangani proses autentikasi *user*.

DAFTAR PUSTAKA

- Himmi, 2015. Rancang Bangun Aplikasi Monitoring Network Berbasis Web Menggunakan HTML5 Pada Dinas Pendidikan Kabupaten Blitar. Universitas Kanjuruhan Malang
- Kamto, 2015. Pemantauan Traffic Jaringan VPN-MPLS WAN (Studi Kasus: PT PLN (Persero) Distribusi Jawa Tengah Dan Daerah Istimewa Yogyakarta). STMIK ProVisi Semarang.
- Kusuma, 2015. Perancangan Sistem Monitoring Perangkat Jaringan Berbasis SNMP. Universitas Muhammadiyah Surakarta.
- Moroney, L., 2017. Firebase Cloud Messaging, in: The Definitive Guide to Firebase. Springer, pp. 163–188.
- Nurzam, 2017. Rancang Bangun Aplikasi Media Laporan Aspirasi dengan Firebase Cloud Messaging Berbasis Mobile. SEMNASTEKNOMEDIA online 5, 4–5.

-
- Ohara, G.J., 2005. Aplikasi Sistem Monitoring Berbasis Web Untuk Open Cluster. Bdg. Sekolah Tinggi Teknologi TELKOM.
- Peter Mell, T.G., 2011. The NIST Definition of Cloud Computing.
- Pradikta, 2013. Rancang Bangun Aplikasi Monitoring Jaringan dengan Menggunakan Simple Network Management Protocol. Institut Teknologi Surabaya.
- Prayoga, F., 2016. Perancangan Prototype Aplikasi Pengumuman Kelas Menggunakan Teknologi Firebase Cloud Message pada Android. Program Studi Teknik Informatika FTI-UKSW.
- Riza, T.A., 2011. Implementasi Manajemen Traffic Dan Bandwidth Internet Dengan IPCOP. Jurnal INKOM 4, 22–28.
- Sagita, A., Prasetijo, A.B., Isnanto, R.R., 2011. Aplikasi Pemantau Statistik Layanan-Layanan TCP dan UDP Berbasiskan SNMP++ Pada Sebuah Jaringan Area Lokal. Universitas Diponegoro.
- Schmuller, J., 1999. Understanding the Foundations of the UML. Sams publishing.
- Wun, Y.A.O., 2014. Monitoring Trafik Jaringan dan Pengaturan PC Router Berbasis Web (Studi Kasus: Labkom STIKOM Surabaya). STIKOM Surabaya.
-