

SISTEM MONITORING UNTUK PERANGKAT IP TAP AGGREGATOR BERBASIS PROTOKOL SNMP DI TELKOMSEL GATOT SUBROTO

System Monitoring for IP Tap Aggregator using SNMP Protocol at Telkomsel Gatot

Subroto

Aqsa Zola¹, Hafiddudin², Nofri Eka Putra³

^{1,2,3}Prodi D3 Teknik Telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom

¹aqsa.zola@gmail.com, hafid@tass.telkomuniversity.ac.id, Nofri_E_Putra@telkomsel.co.id

Abstrak

Sistem *monitoring* pada saat ini tidak cukup hanya melihat suksesnya protokol-protokol dalam *network element*. Pada saat ini, dibutuhkan sebuah *monitoring* untuk *user experience*. Untuk mendapatkan *monitoring* tersebut, dibutuhkan duplikasi trafik. Penduplikasian trafik dapat dilakukan dengan *tap aggregator*. Namun *tap aggregator* mempunyai kelemahan yaitu tidak mempunyai *monitoring*nya sendiri.

Untuk mengatasi permasalahan diatas, maka dibuatlah suatu sistem *monitoring* untuk perangkat ip *tap aggregator* yang mampu mengolah trafik menjadi sebuah grafik setiap lima menit dan mampu mengirimkan notifikasi ketika terdapat kejanggalan pada nilai trafik. Untuk merealisasikan *monitoring tap aggregator*, dibutuhkan snmp protokol untuk *capture* trafik sebelum diolah menjadi grafik dan notifikasi.

Proyek akhir ini menghasilkan sebuah *monitoring* untuk *tap aggregator* yang berisi grafik dari trafik yang didapat oleh snmp, *database* yang sudah terintegrasi dengan *server*, dan notifikasi *telegram* yang dalam pengujian fungsionalitas, dan implementasinya, tingkat keberhasilan fungsionalitas pada setiap *activity* dan *database* sebesar 100%. Dapat disimpulkan bahwa aplikasi dan *database* dapat berfungsi dengan baik.

Kata kunci: SNMP, Tap Aggregator, Monitoring Performance, Monitoring konfigurasi, OID, Bash, MySQL, Telegram.

Abstract

The current monitoring system is not enough to see only the protocols in network elements. At this time, it takes a monitoring for user experience. To get the monitoring, it needs duplication of traffic. Traffic duplication can be done with tap aggregator. However tap aggregator has a weakness that does not have its own monitoring.

To overcome the above problems, a monitoring system for the ip tap aggregator device is able to process traffic into a graph every five minutes and is able to send notifications when there is an awkwardness on the value of traffic. To realize tap aggregator monitoring, a snmp protocol is required to capture traffic before it is processed into graphs and notifications.

This final project produces a monitoring for tap aggregator that contains graphs of snmp traffic, integrated database server, and telegram notification that in testing functionality, and its implementation, the success rate of functionality in each activity and database is 100%. It can be concluded that applications and databases can function properly.

Keywords: SNMP, Tap Aggregator, Monitoring Performance, Configuration monitoring, OID, Bash, MySQL, Telegram.

1. Pendahuluan

SGSN dan GGSN banyak dilewati traffic-traffic seperti user plane, control plane, dan lain-lain. Ketika ada traffic yang bermasalah, admin tidak dapat mengetahuinya dari awal. Ada 2 cara untuk mengatasi masalah ini, yang pertama, dengan menggunakan span port, dan memasang tap aggregator. Dengan span port, diduplikasikan data dalam router service tersebut. Namun karena menduplikasikan data, load dari router bertambah menjadi signifikan. Cara yang kedua, dengan memasang Tap Aggregator.

Tap aggregator ini berfungsi untuk melakukan agregasi terhadap data. Data dibagi menjadi dua, satu untuk keperluan service, satu lagi untuk keperluan monitoring sehingga tidak menimbulkan load tambahan di perangkat networking yang terlibat. Cara kerja system tap aggregator adalah dengan memotong traffic di level physical/cabling. Cara kerja seperti ini sangat berpotensi menjadi single point of failure untuk layanan data. Sehingga sistem ini perlu di monitor dengan baik. Sistem aggregator ini tersedia di industri tanpa sistem monitoring yang terintegrasi. Sistem tap aggregator mendukung protocol simple network management protocol (SNMP), sehingga interfacing data bisa dilakukan dengan metoda walking SNMP. Dengan manipulasi data-data SNMP Tap Aggregator dan mekanisme thresholding, segala kejanggalan parameter operasional Tap Aggregator bisa langsung di ketahui secara realtime. Informasi bisa diolah lebih lanjut dalam bentuk grafik yang menampilkan throughput dan beberapa parameter lain yang didukung oleh SNMP. Kejanggalan pola dari trafik juga bisa dimanipulasi untuk dikirimkan sebagai notifikasi ke engineer terkait, sehingga degradasi layanan bisa cepat terdeteksi.

2. Dasar Teori

2.1. Gprs Core Network

GPRS core network secara umum mempunyai fungsi untuk menyediakan *mobility management*, *session management* dan transport untuk paket Internet Protocol di jaringan GSM dan WCDMA. *Core network* juga menyediakan support untuk fungsi tambahan lain seperti *billing* dan *lawful interception*[4].

2.2. Mysql

SQL (*Structured Query Language*) adalah sebuah konsep untuk pengoperasian data terutama untuk pemilihan dan pemasukan data yang memungkinkan dikerjakan dengan otomatis. Lalu MySQL merupakan sebuah implementasi dari sistem manajemen basis data relasional yang didistribusikan secara gratis atau GPL (General Public License)[2].

2.3. Bash shell

UNIX shell program merupakan sebuah bahasa pemrograman yang prinsip kerjanya menerjemahkan baris per baris. Ada beberapa tipe shell program, namun standar dari shell adalah bash yang hampir ada di setiap operasi sistem linux.

Bash adalah singkatan dari Bourne Again Shell, bahasa pemrograman ini merupakan bahasa paling dianjurkan untuk pemula atau bahkan untuk yang sudah mahir[1].

2.4. SNMP

SNMP adalah protokol aplikasi pada jaringan TCP/IP yang menangani manajemen jaringan yang terdapat pada sebuah perangkat yang berdiri sendiri. Monitoring jaringan dapat dilakukan dengan cara pengumpulan nilai-nilai informasi dari kondisi jaringan jarak jauh atau menggunakan satu pusat pengamatan. Ada tiga komponen dasar yang digunakan pada sistem pemantauan jaringan menggunakan SNMP adalah[6]:

1. Manager SNMP

Manager SNMP adalah sebuah computer yang mengoperasikan perangkat lunak untuk manajemen jaringan.

2. SNMP Agen

SNMP agen adalah sebuah software yang berada pada setiap alat yang akan di monitoring.

3. Manajemen Information Base (MIB)

MIB SNMP merupakan sebuah struktur basis data variable dari jaringan yang dikelola. Struktur ini bersifat hierarki dan memiliki aturan sedemikian rupa sehingga informasi setiap variable dapat dikelola atau di tetapkan dengan mudah. Object identifier atau OID, mengidentifikasi atau memberi nama objek-objek dalam pohon MIB. Penamaan ini dilakukan secara unik seperti nomor telepon yang diorganisasikan secara hierarki. Contoh OID: 1.3.6.1.2.1.31.1.1.1.10.

2.5. NetOptics xStream40

NetOptics xStream40 merupakan sebuah tap aggregator untuk jaringan 40G. Solusi lanjutan ini di rekayasa untuk memberikan pantuan lalu lintas yang tinggi karena jaringan melonjak menjadi 40G. NetOptics xStream40 menawarkan harga murah dan memberikan pilihan tambahan yang efisien untuk navigasi, pengelolaan, menyederhanakan tugas dalam skala besar, dan berbagi lalu lintas yang berkembang di beberapa alat[5].

2.6. Highcharts

Highchart merupakan perpustakaan grafik multi-platform berbasis SVG yang telah dikembangkan dari tahun 2009. Highchart memudahkan untuk menambahkan grafik interaktif dan mudah dioptimalkan ke proyek

web dan seluler. Highchart memiliki dokumentasi yang kuat, responsif, maju, dan dukungan akses terdepan di industri[3].

2.7. Telegram

Telegram adalah sebuah aplikasi untuk saling mengirim pesan dengan teman atau keluarga. Selain itu, telegram mempunyai bot. Bot API(Application Programming Interface) adalah interface berbasis HTTP untuk pengembang bot di telegram. User bisa berinteraksi dengan bot dengan cara memberikannya messages, command, dan inline request[7].

2.8. Bootstrap

Bootstrap adalah sebuah framework css yang memudahkan dalam pembangunan tampilan web. Bootstrap dikembangkan oleh twitter mark otto dan Jacob thronthon. Saat ini bootstrap merupakan open source yang lisensinya dipegang oleh MIT[8].

2.9. PHP

PHP merupakan bahasa pemrograman untuk server, dan powerful tool untuk membuat web dinamis dan web intraktif[9].

Contoh proses perjalanan PHP pada sisi server:

1. User memasukan data diri, dan mengklik submit.
2. Setelah submit, akan php akan mengarah ke script lain yang sudah disetting menggunakan metode post atau get. Dan menampilkan tampilan data diri yang baru saja diinput.
3. Lalu dalam penampilan data diri tersebut terdapat edit dan delete yang prosesnya sama persis dengan memasukan data diri.

2.10 Pembuatan aplikasi

Untuk bisa menghasilkan *program* yang baik memerlukan analisis yang baik pula, baik itu analisis sistem, stuktur *data* maupun analisis *requirement*, selain itu juga diperlukan persiapan-persiapan yang matang. Hal ini berlaku bagi siapapun, bahkan seorang programmer professional sekalipun.

Program memang sudah menjadi kebutuhan pokok bagi masyarakat IT. Karena segala sesuatu yang dilakukan di dalam IT pastilah memerlukan program. Program yang paling sederhana sekalipun setidaknya memiliki 3 bagian:

1. Input – Masukan *data*.
2. Proses – Pemrosesan *input*.
3. Output – Keluaran *program*, kebutuhan yang diharapkan.

Dalam membuat *program*, pemrograman adalah pokok dari *proses* pembuatan *program* itu sendiri namun pemrograman bergantung dari pemahaman persoalan, analisis sistem, perencanaan-perencanaan dalam mendesain *program* itu sendiri.

Selain pemrograman hal yang utama harus dilakukan adalah merencanakan langkah-langkah yang harus diambil dalam menyelesaikan masalah. Karena dengan mengetahui masalah dan langkah-langkah penyelesaian berarti sudah menyelesaikan program tersebut sebanyak 50% dari total pekerjaan, selanjutnya adalah teknis pembuatan itu sendiri yang di kenal dengan pemrograman/koding.

Dalam membuat sebuah *program* setidaknya ada beberapa hal yang perlu anda lakukan:

1. Mendefinisikan Masalah

Masalah/*Problem* disini adalah komponen apa saja yang diperlukan agar *program* ini jalan dikenal dengan masukan/inputnya apa saja, mendefinisikan apa yang nanti akan dilakukan oleh program dan bagaimana keluaran dari program yang diharapkan nantinya. Pada tahap ini juga dikenal *requirement* analisis atau analisa kebutuhan.

2. Perencanaan

Pada tahap ini adalah mendefinisikan langkah-langkah apa saja yang dilakukan oleh program dalam menyelesaikan permasalahan yang ada. Bentuk dari perencanaan itu bisa berupa *flowchart* ataupun algoritma dari *program*, sehingga kita akan tahu proses apa saja yang ada dalam program tersebut. Semakin *detail flowchart* atau algoritma yang dibuat semakin mudah juga pada tahap implementasi/*coding* nantinya.

Flowchart adalah suatu *diagram* menggunakan simbol-simbol khusus yang sudah menjadi *standard* internasional yang berisi langkah-langkah untuk menyelesaikan suatu masalah. Sedangkan algoritma bukan merupakan simbol tapi keterangan-keterangan yang sesuai dengan keinginan kita, tidak ada standarnya. Oleh karena itu *flowchart* biasa juga disebut sebagai algoritma dalam bentuk simbol-simbol khusus yang dihubungkan dengan anak panah.

Membuat *flowchart* terlebih dahulu akan lebih menghemat waktu daripada langsung melakukan *coding* sambil mencoba-coba. Kegiatan mencoba-coba akan menghabiskan waktu ketika implementasi/koding karena harus merubah koding yang lumayan banyak. Karena itu, biasakan membuat *flowchart* terlebih dahulu sebelum memecahkan suatu masalah.

3. Implementasi

Kini saatnya menulis *program*, tahap ini juga mencakup tahap perbaikan *error* dan *testing*. Menulis *program* dengan terstruktur dan sesuai dengan *flowchart* yang telah dibuat.

4. *Testing*

Testing dalam hal ini dibagi menjadi 4 bagian, yaitu:

1. *Unit Testing*

Menguji setiap *unit* dan modul yang terdapat dalam *program* tersebut.

2. *Integration testing*

Menguji integrasi yang dilakukan kepada *program* seperti halnya ketika *program* tersebut sudah diinstall di *client* kita yang membutuhkan integrasi dengan sistem yang lain seperti halnya integrasi dengan *database*.

3. *Validation testing*

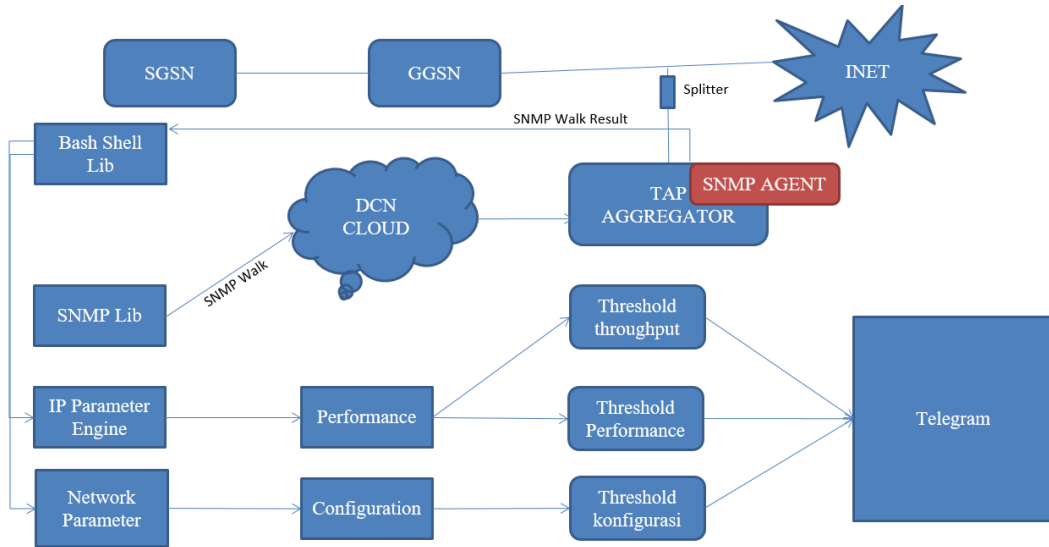
Menguji masukan yang diberikan kepada *program*. Apapun masukannya *program* harus bisa menyelesaikan dengan baik.

4. *Sistem testing*

Pada tahap ini menguji performa dari *program*[13].

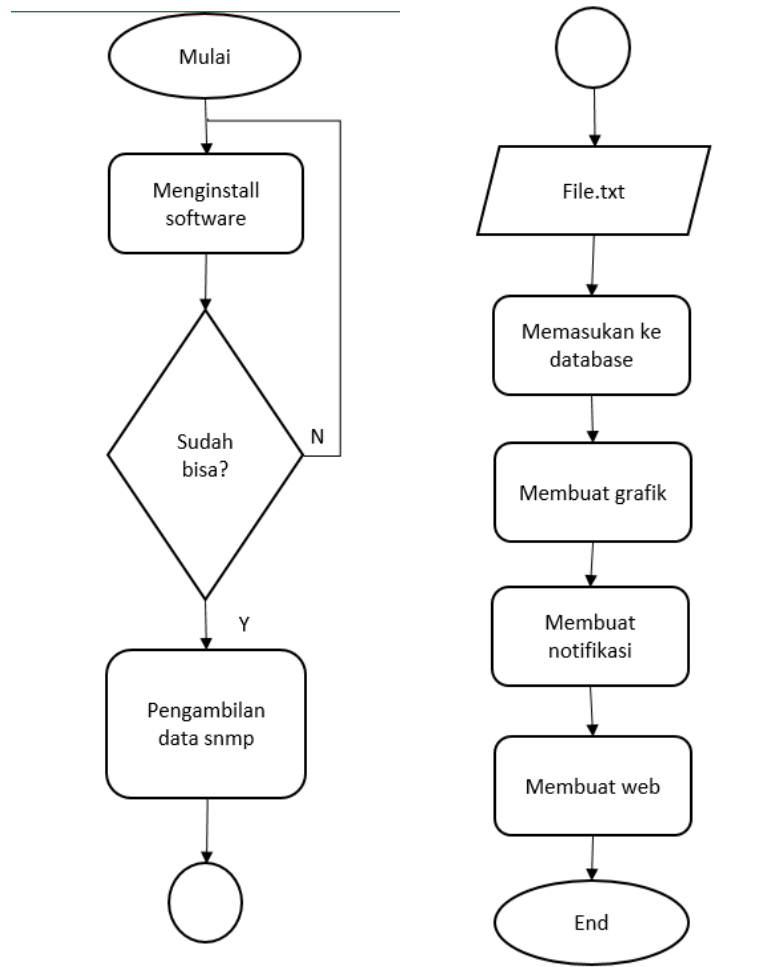
3. Perancangan

3.1. Blok Diagram Sistem



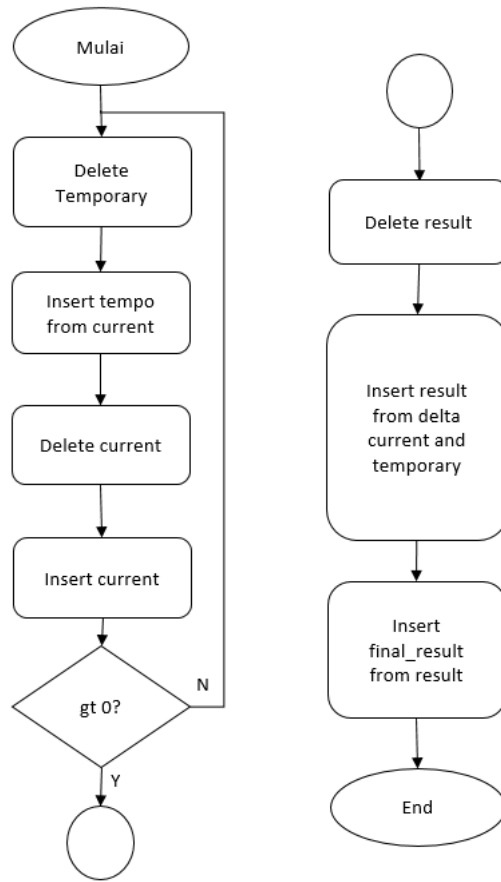
Gambar 3.1. Blok Diagram Sistem

3.2. Diagram Alir Pengerjaan



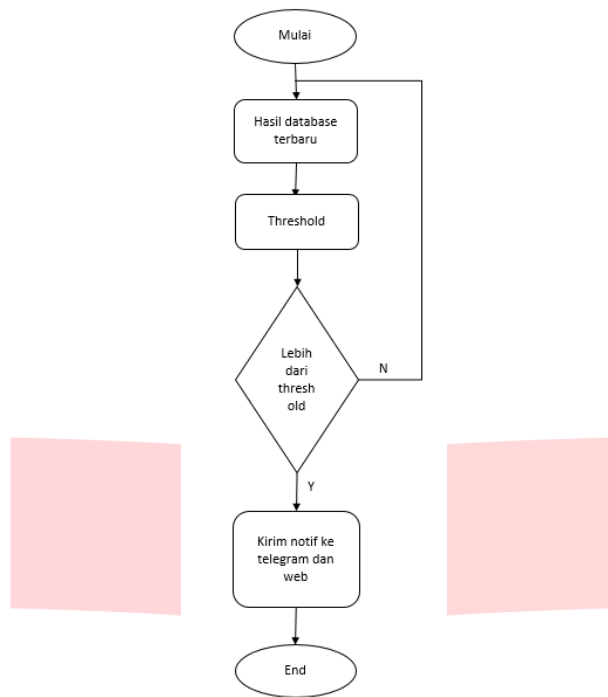
Gambar 3.2. Diagram Alir Pengerjaan

3.3. Diagram alir pemasukan database



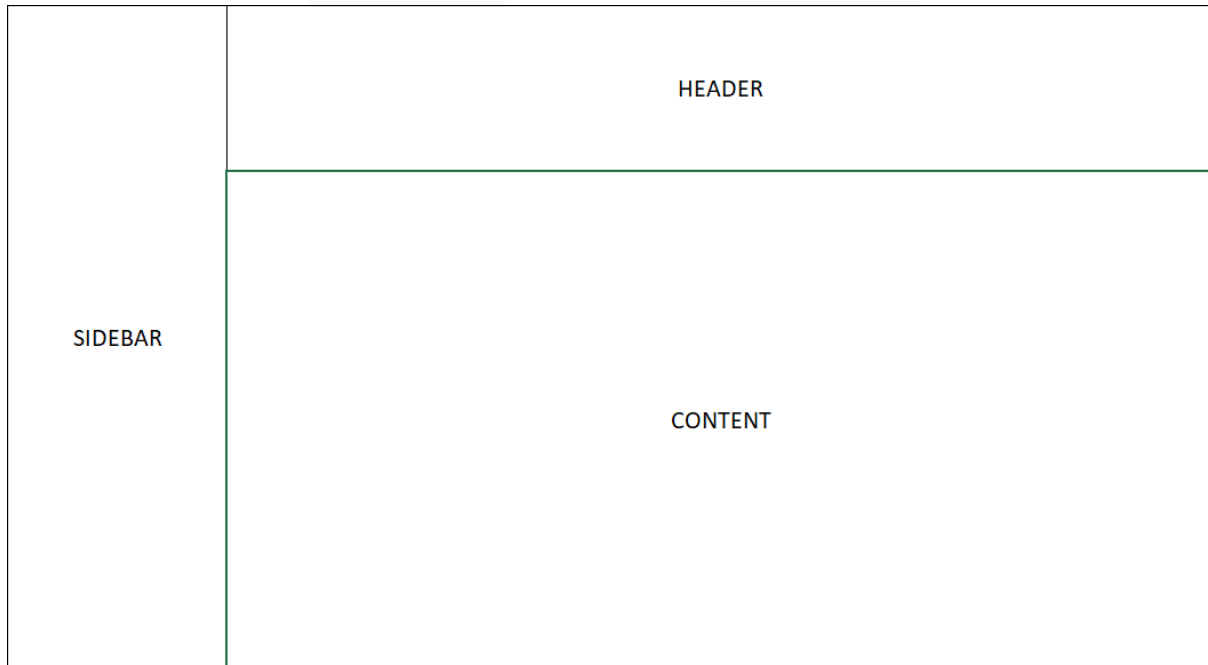
Gambar 3.3 Diagram alir database

3.4. Diagram alir notifikasi



Gambar 3.4. Diagram alir notifikasi

3.5. Layout web



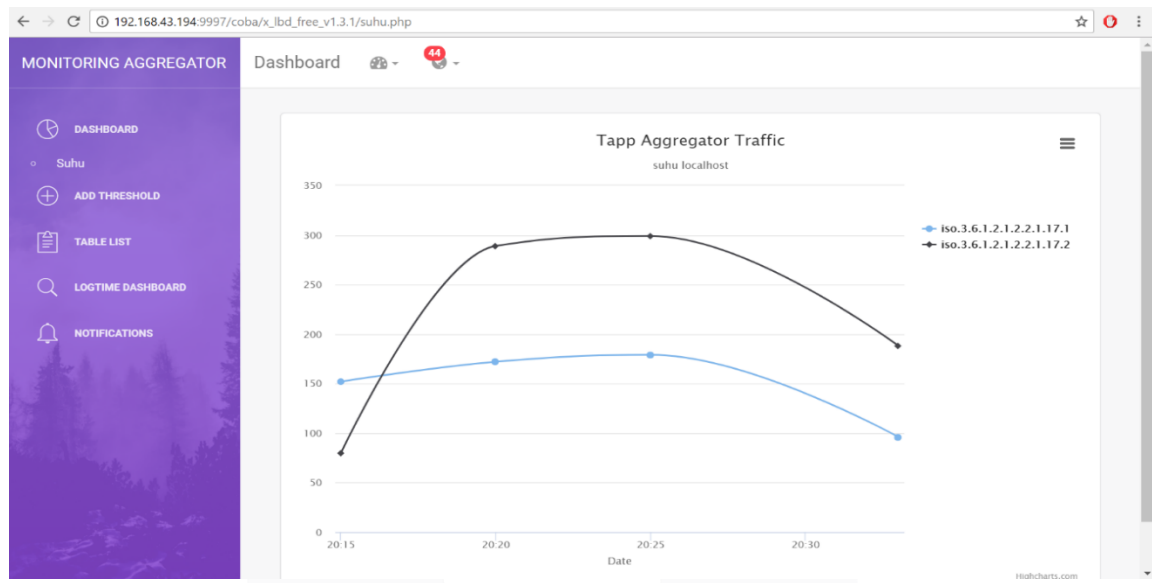
Gambar 3.5. Layout web

4. Pengujian

4.1. Pengujian dengan localhost

Tabel 1. Parameter localhost

No	Community string	IP	Parameter/OID	Alias
1	Public	Localhost	iso.3.6.1.2.1.2.2.1.17	Suhu



Gambar 4.1. Hasil

Pada pengujian ini, dilakukan pada laptop dengan sistem operasi linux Ubuntu. Dijalankan snmpwalk pada parameter seperti tabel 1 dan setelah itu didapatkan grafik seperti pada gambar 4.1. Pengujian ini hanya untuk mencoba sistem apakah sudah bisa menampilkan grafik atau belum.

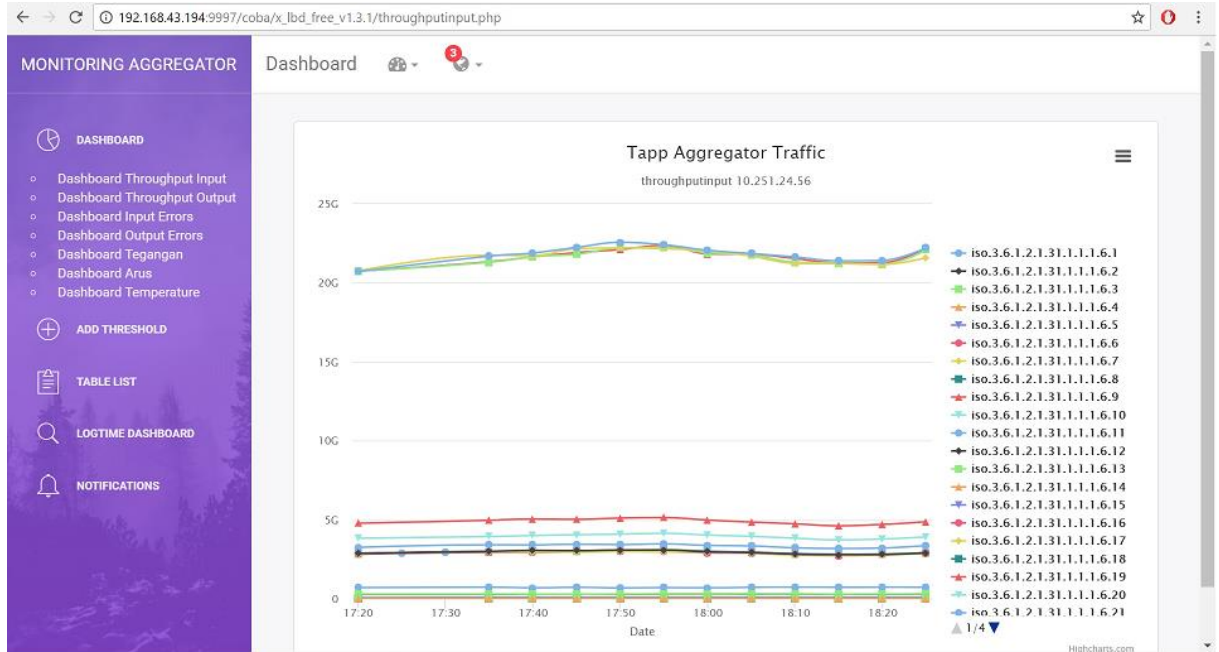
4.2 Pengujian dengan tap aggregator

4.2.1 Pengujian Dashboard

Tabel 2. Parameter tap aggregator

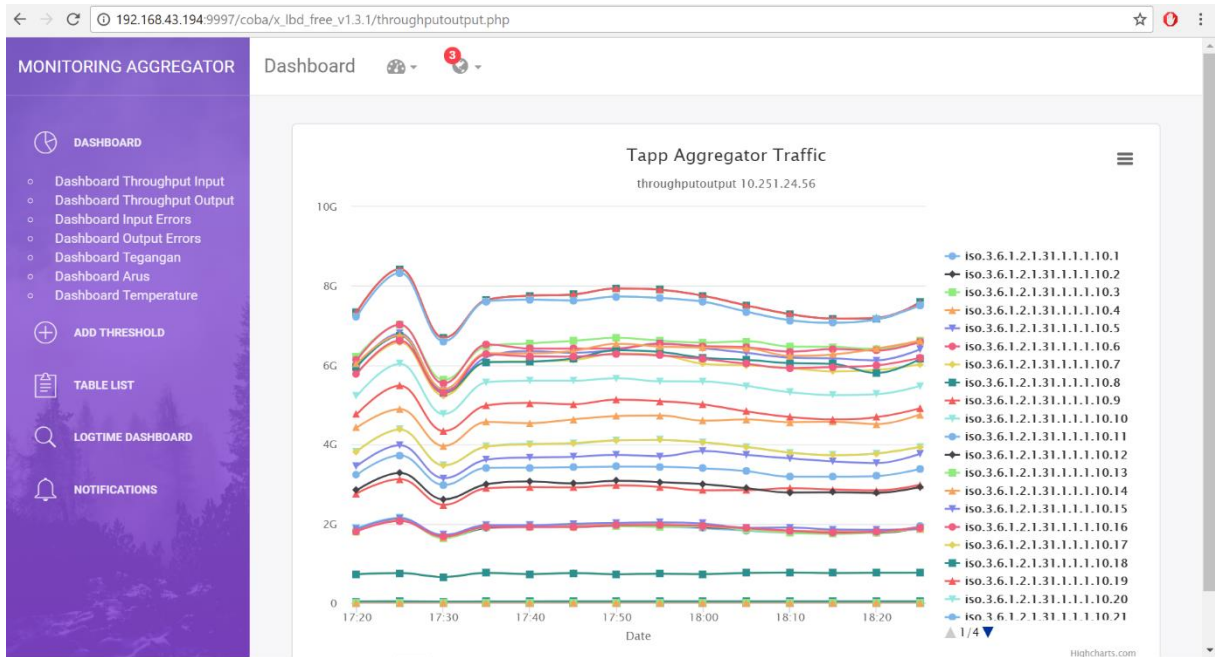
No	Community string	IP	Parameter	Alias
1	Public	10.251.24.56	iso.3.6.1.2.1.2.2.1.10	Throughput input
2	Public	10.251.24.56	iso.3.6.1.2.1.2.2.1.16	Throughput output
3	Public	10.251.24.56	iso.3.6.1.2.1.2.2.1.14	Inerrors
4	Public	10.251.24.56	iso.3.6.1.2.1.2.2.1.20	Outerrors
5	Public	10.251.24.56	iso.3.6.1.4.1.23022.100.2.3.1.4.1.1.22	Arus
6	Public	10.251.24.56	iso.3.6.1.4.1.23022.100.2.3.1.4.1.1.21	Tegangan

7	Public	10.251.24.56	iso.3.6.1.4.1.23022.100.2.3.1.4.1.1.20	Temperature
8	Public	10.251.24.56	iso.3.6.1.4.1.23022.100.12.1.1.4.1.3.3	Config



Gambar 4.2 Throughput input



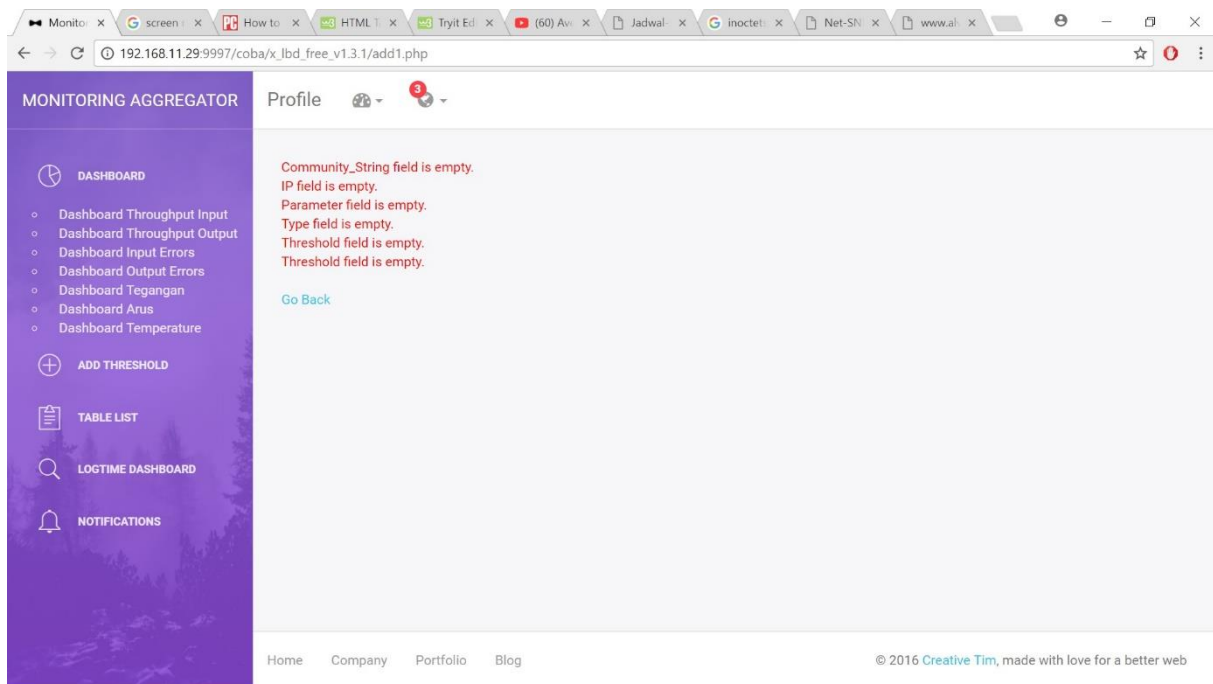


Gambar 4.3 Throughput output

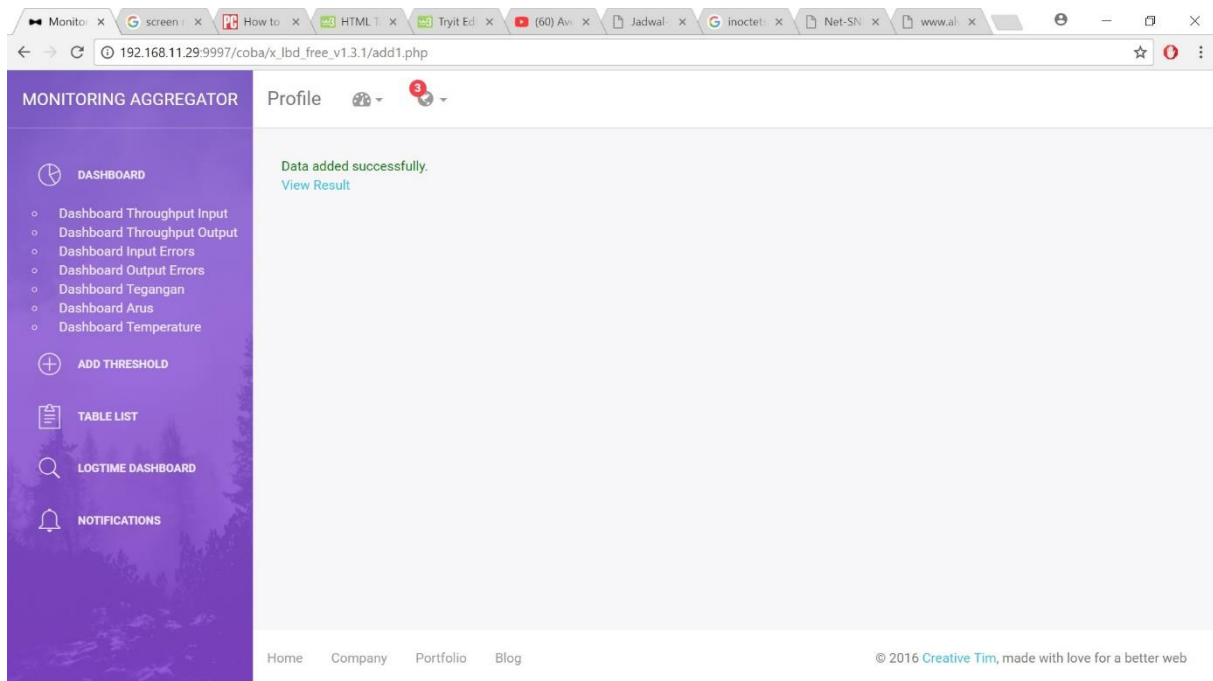
The screenshot shows a web browser window displaying a monitoring dashboard. The browser's address bar shows the URL `192.168.11.29:9997/coba/x_lbd_free_v1.3.1/add.php`. The dashboard has a purple sidebar with navigation options: DASHBOARD, ADD THRESHOLD, TABLE LIST, LOGTIME DASHBOARD, and NOTIFICATIONS. The main content area is titled 'Profile' and features a form titled 'Add Threshold'. The form contains the following fields: COMMUNITY_STRING (Community_string), IP (IP), PARAMETER (Parameter), TYPE (Type), THRESHOLD (Threshold), ALIAS (Alias), and HOSTNAME (Hostname). A blue 'Add' button is located at the bottom right of the form.

Gambar 4.4 add threshold

4.2.2 Pengujian fitur add



Gambar 4.6 jika ada field yang tidak diisi



Gambar 4.5 penambahan sukses

Untuk pengujian add threshold berjalan dengan lancar. Terdapat kondisi jika salah satu field kosong, maka akan di beritahukan seperti gambar 4.6. Setelah semua terisi, maka akan tertulis data sukses ditambah seperti gambar 4.5.

MONITORING AGGREGATOR

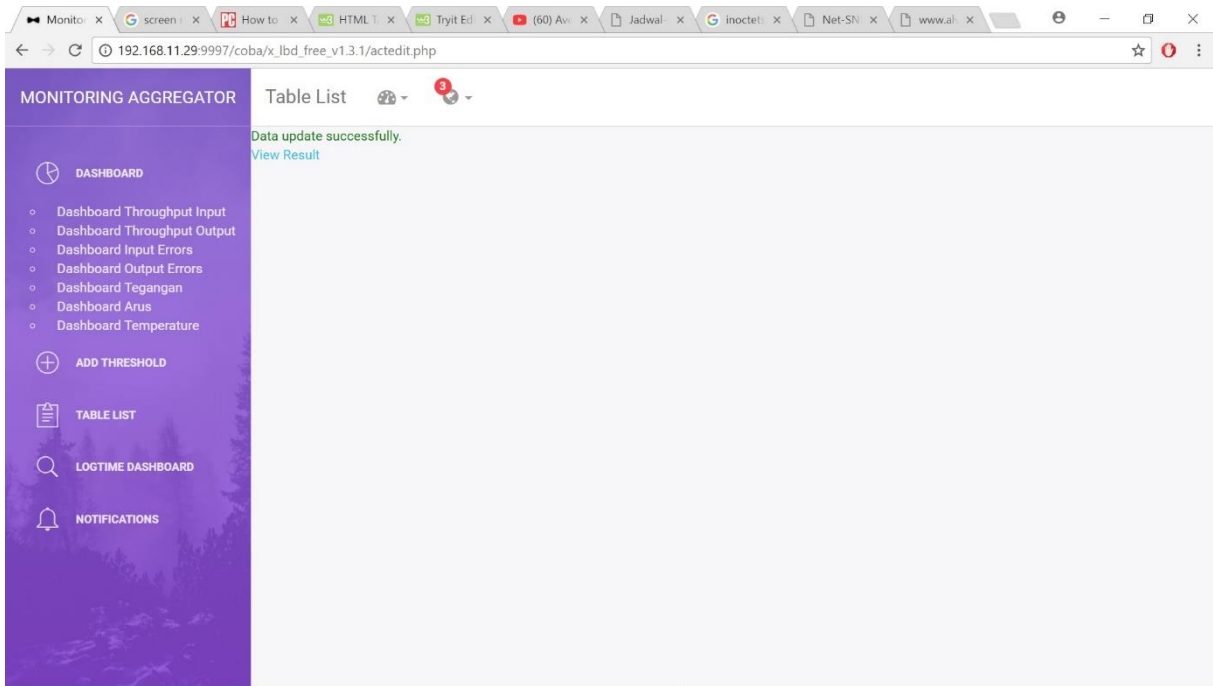
Table List

COMMUNITY_STRING	IP	PARAMETER	TYPE	THRESHOLD	ALIAS	HOSTNAME	
public	10.251.24.56	iso.3.6.1.2.1.2.2.1.10	Counter32:	3000000000	throughputinput	xStream40	Edit Delete
public	10.251.24.56	iso.3.6.1.4.1.23022.100.2.3.1.4.1.1.20	String:	40	temperature	xStream40	Edit Delete
public	10.251.24.56	iso.3.6.1.4.1.23022.100.2.3.1.4.1.1.21	String:	3.5	tegangan	xStream40	Edit Delete
public	10.251.24.56	iso.3.6.1.4.1.23022.100.2.3.1.4.1.1.22	String:	40	arus	xStream40	Edit Delete
public	10.251.24.56	iso.3.6.1.2.1.2.2.1.16	Counter32:	3000000000	throughputoutput	xStream40	Edit Delete
public	10.251.24.56	iso.3.6.1.2.1.2.2.1.14	Counter32:	1	inerrors	xStream40	Edit Delete
public	10.251.24.56	iso.3.6.1.2.1.2.2.1.20	Counter32:	1	outerrors	xStream40	Edit Delete
public	10.251.24.56	iso.3.6.1.4.1.23022.100.12.1.1.4.1.3.3	String:	1	config	xStream40	Edit Delete
public	10.251.24.56	iso.3.6.1.2.1.92.1.1.1	Gauge32:	1	power		Edit Delete

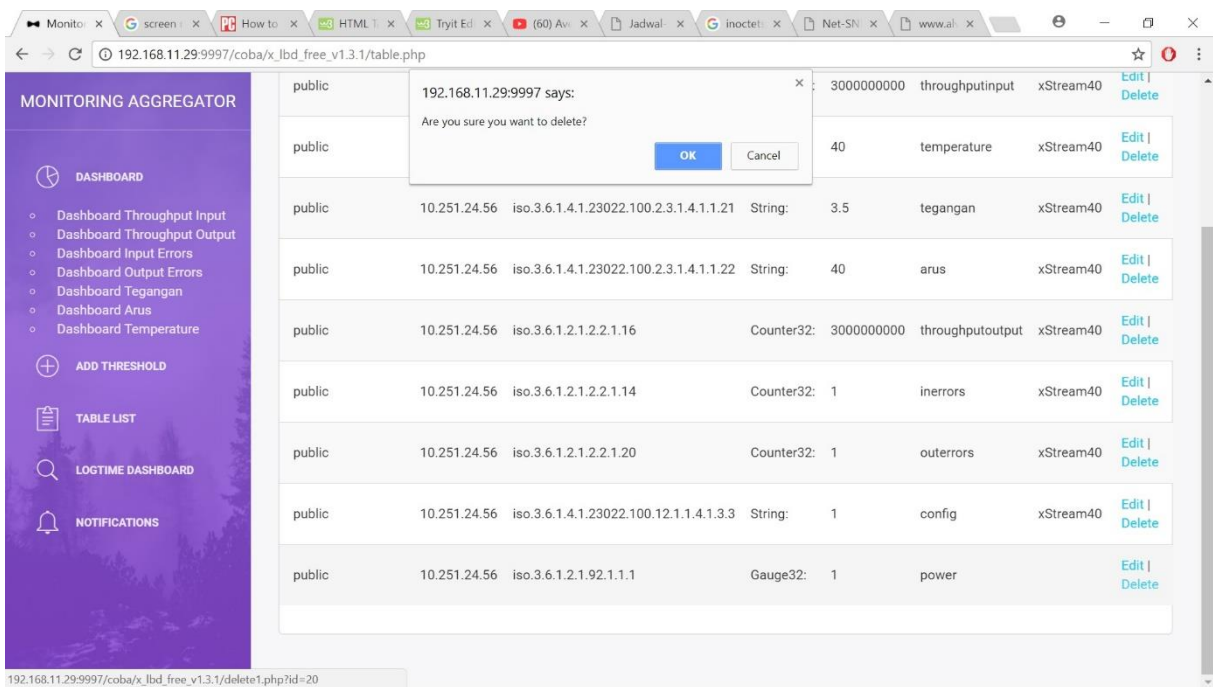
Gambar 4.7 table list

4.2.3 Pengujian table

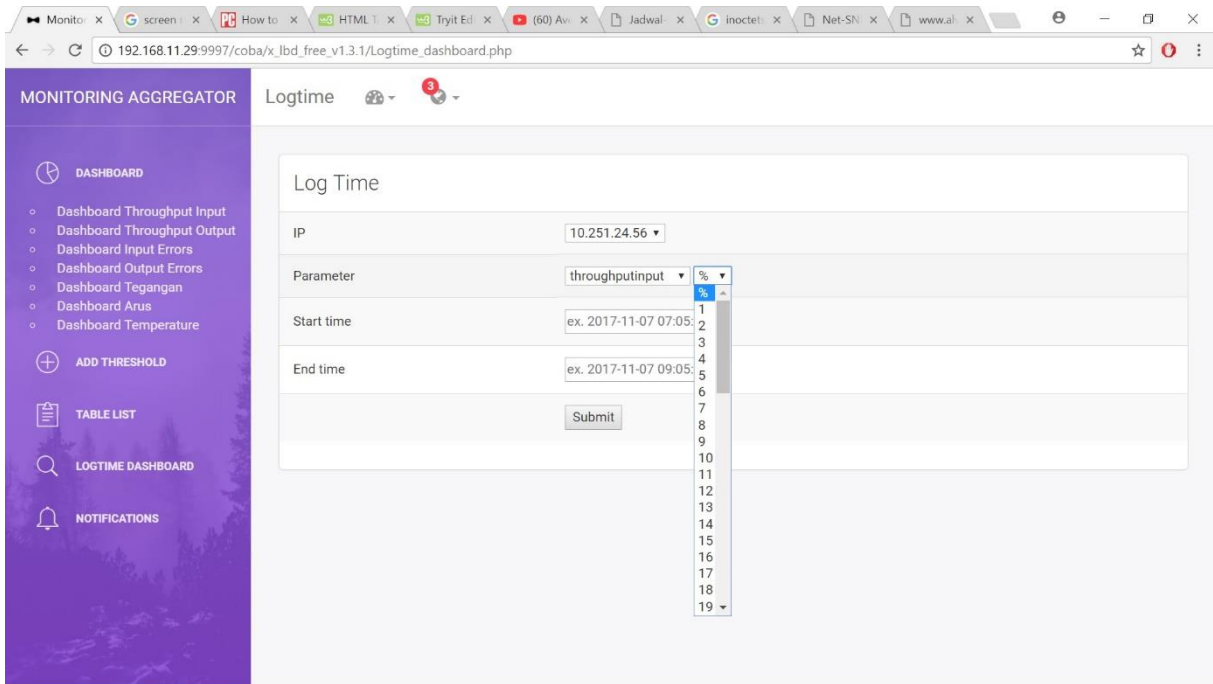
Fungsi tabel ini untuk membaca yang telah diisi pada add threshold. Selain untuk membaca, fungsi tabel ini adalah untuk mengedit dan mendelete data yang sudah ditambahkan.



Gambar 4.8 edit sukses



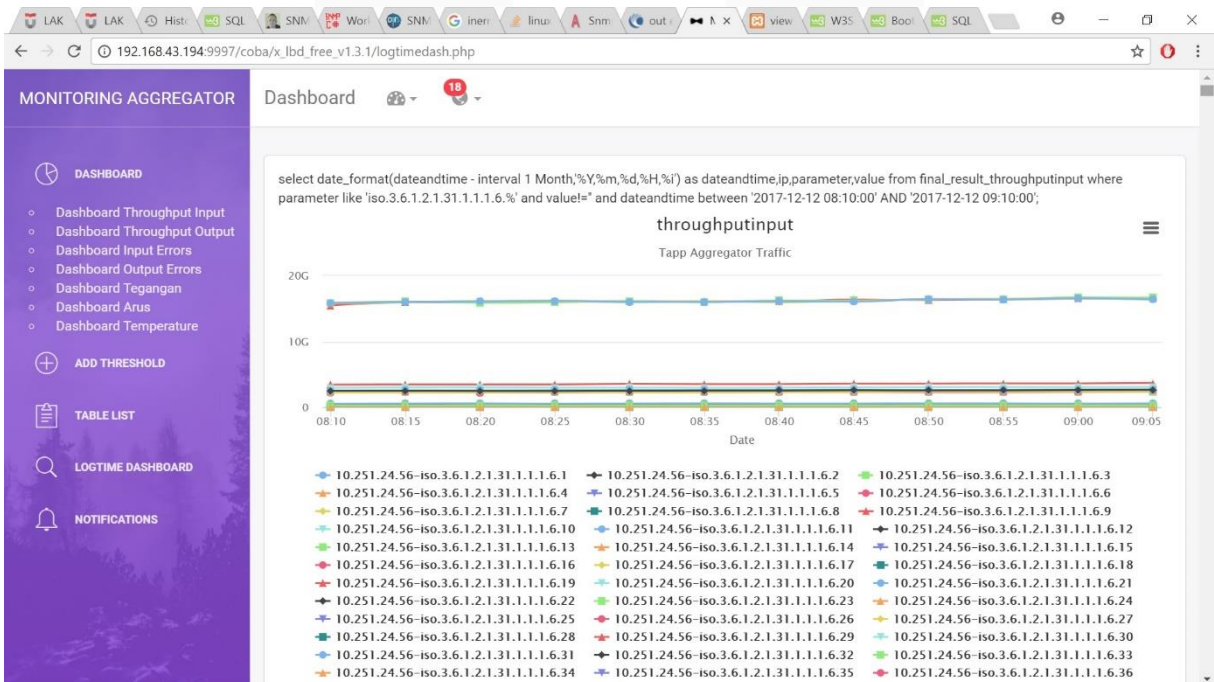
Gambar 4.9 delete



Gambar 4.10 logtime dashboard

4.2.4 Pengujian Logtime

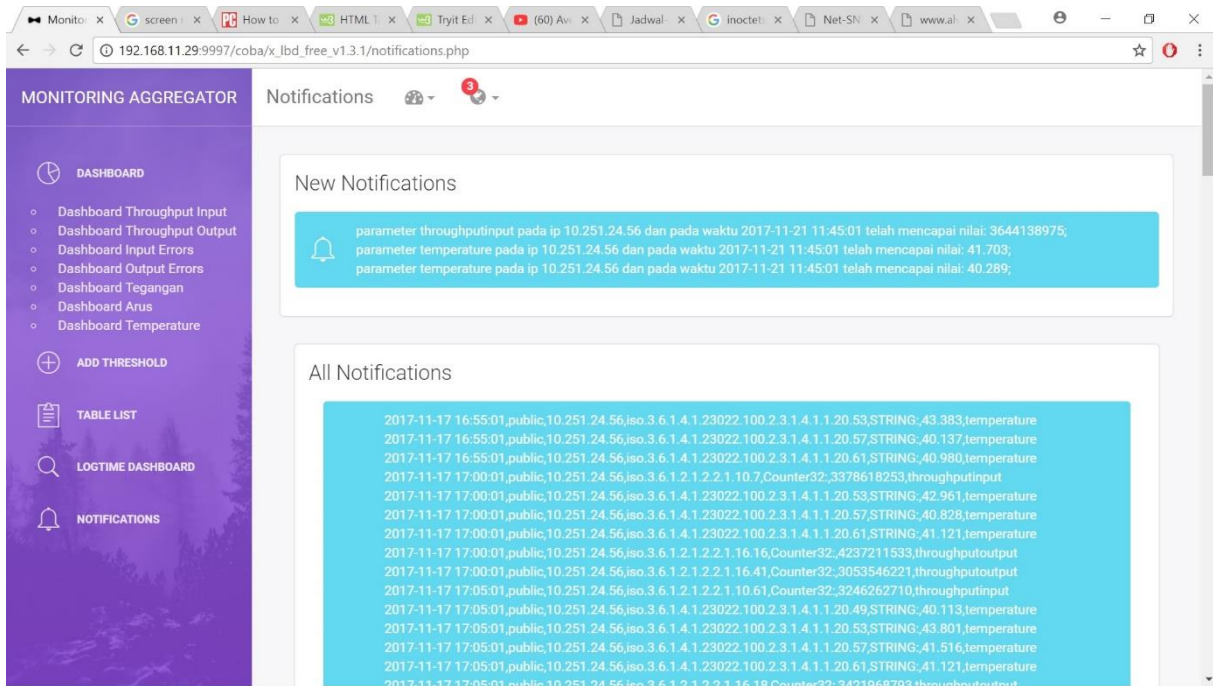
logtime dashboard berfungsi untuk melihat dashboard lebih detail dengan waktu dan port yang diinginkan.



Gambar 4.11 hasil logtime

4.2.5 Pengujian Notifikasi

New notifications merupakan notifikasi terbaru yang jumlahnya seperti pada bagian header. All notifications merupakan notifikasi yang terdapat pada database pada hari ini ada 24 jam sebelumnya,



Gambar 4.13 notifikasi performance.

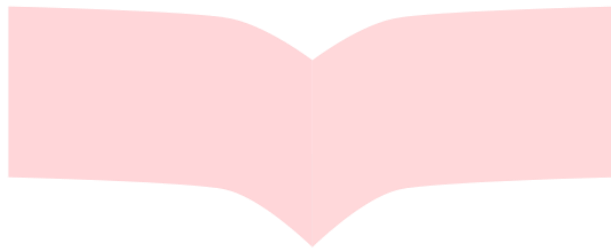


Gambar 4.12 notifikasi perubahan konfigurasi.

5. Kesimpulan

Berdasarkan perancangan dan pengujian pada proyek akhir ini, maka dapat disimpulkan bahwa:

1. Dapat memonitoring performance tap aggregator.
2. Dapat memonitoring perubahan konfigurasi.
3. Di ketahui nilai throughput sebesar 1Gbps-20Gbps karena penggunaan Tap Aggregator 40G.
4. Didapatkan nilai inerrors dan outerrors 0.
5. Dapat melihat perubahan konfigurasi dari nilai in_ports 1 221.132.211.192 ip4_src_mask 255.255.255.240 action redir redit_ports 10,38,39 menjadi wwwwww.
6. Model yang sama bisa digunakan untuk monitor semua perangkat yang memiliki SNMP agen.



Daftar Pustaka

- [1] Garrels, Machtelt. 2008. *Bash Guide for beginners*. Fultus Corporation.
- [2] MySQL. <https://id.wikipedia.org/wiki/MySQL>. 7 Mei 2012.
- [3] Highcharts. <https://www.highcharts.com/products/highcharts/>.
- [4] Prihandika, Lukman. (2010). <https://brokenz1.wordpress.com/2010/08/17/gprs-core-network/>. GPRS Core Network. Jakarta.
- [5] Ltd, Frame, Communications. NetOptics xStream40. <http://www.frame.co.uk/m/taps-and-aggregation-devices/netoptics-xstream%E2%84%A2-40.aspx>.
- [6] Saputra, R. S. (2016). PERANCANGAN DAN IMPLEMENTASI APLIKASI SISTEM MONITORING JARINGAN BERBASIS WEB. Bandung.
- [7] Telegram. <https://core.telegram.org/bots/api>.
- [8] Herlangga, Kresna, Galuh. 2015. Bootstrap. <https://www.codepolitan.com/tutorial/cara-menggunakan-bootstrap-3-untuk-membuat-web>.
- [9] PHP. W3school: <https://www.w3schools.com/php/>.
- [10] Hill, Lindsay. 2014. SNMP Counters-Collection Interval Variation. <https://lkhill.com/snmp-counters-collection-interval-variation/>.
- [11] SQL. W3school: <https://www.w3schools.com/sql/>.
- [12] Bootstrap. W3school: <https://www.w3schools.com/bootstrap/>.
- [13] Prastiawan Adrian M. 2008. Tahapan membuat program dengan baik. <http://wa2n.staff.uns.ac.id/2008/09/14/tahapan-membuat-program-dengan-baik/#comments>.