Volume 2, No. 1 2021

JISE

Journal of Internet and Software Engineering

ISSN 2797-9016

https://ugm.id/jise

The journal published by
Department of Electrical Engineering and Informatics
Vocational College, Universitas Gadjah Mada

JISE
Journal of Internet and Software Engineering
Volume 2, No. 1 2021

EDITORIAL BOARD

Journal of Internet and Software Engineering (JISE)

https://ugm.id/jise

The journal published by
Department of Electrical Engineering and Informatics
Vocational College, Universitas Gadjah Mada
Sekip unit III, Caturtunggal, Terban,
Kec. Gondokusuman, Kab. Sleman, D.I Yogyakarta 55281

JISE
Journal of Internet and Software Engineering
Volume 2, No. 1 2021

https://ugm.id/jise
The journal published by
Department of Electrical Engineering and Informatics
Vocational College, Universitas Gadjah Mada

PEMANTAUAN DAN ANALISIS PERFORMA SISTEM *HONEYPOT* DENGAN *SIMPLE NETWORK MANAGEMENT PROTOCOL* (SNMP)

Imron Kadafi Hariri, Alif Subardono
Departemen Teknik Elektro dan Informatika
Universitas Gadjah Mada
imron.kadafi.h@mail.ugm.ac.id, alif@ugm.ac.id

Abstract – Installation of the honepot sensor on Raspberry Pi device integrated with Modern Honey Network (MHN) is a DSSDI UGM action to see the threats of existing public Network, so it can be preventive action. But the installed Honeypot sensors sometimes have problems in the form of overload on the System so that data attacks are not recorded. It is necessary to develop a monitoring System on the Honeypot System to monitor and manage System resources and to analyze Honeypot data needs. System monitoring which is one of the Network Management System (NMS) functions can be implemented with Simple Network Management Protocol (SNMP) to monitor the performance of Honeypot sensor devices. From monitoring the Honeypot System shows the Honeypot sensor device overloaded in memory. In addition, from the monitoring is seen the influence of the number of attacks on System performance.

Keywords - Honeypot, Modern Honey Network (MHN), Network Management System (NMS), Simple Network Management Protocol (SNMP)

Intisari – Pemasangan sensor honepot pada perangkat Raspberry Pi yang terintegrasi dengan Modern Honey *Network* (MHN) merupakan tindakan DSSDI UGM guna melihat ancaman-ancaman yang ada jaringan publik, sehingga dapat dilakukan tindakan pencegahan. Namun sensor *Honeypot* yang terpasang kadang mengalami masalah berupa overload pada sistem sehingga data serangan tidak tercatat. Perlu dikembangakan sistem pemantauan pada sistem *Honeypot* tersebut untuk memantau dan mengelola sumber daya sistem juga untuk kebutuhan analisis data *Honeypot*. Pemantauan sistem yang merupakan salah satu fungsi *Network Management System* (NMS) dapat dierapkan dengan *Simple Network Management Protocol* (SNMP) untuk memantau performa dari perangkat sensor *Honeypot*. Dari pemantauan sistem *Honeypot* menunjukkan perangkat sensor *Honeypot* mengalami overload pada memori. Selain itu dari pemantauan tersebut terlihat adanya pengaruh jumlah serangan terhadap performa sistem.

Kata Kunci - *Honeypot*, Modern Honey *Network* (MHN), *Network Management System* (NMS), *Simple Network Management Protocol* (SNMP)

I. PENDAHULUAN

Serangan terhadap sistem informasi dan jaringan saat ini semakin banyak dan bermacam-macam seiring dengan jumlah penggunaan teknologi komputer yang terus meningkat. Adanya ancaman-ancaman yang ada pada sistem informasi membuat Direktorat Sistem dan Sumber Daya Informasi Universitas Gadjah Mada (DSSDI UGM) sebagai penyedia sistem dan layanan informasi di lingkungan UGM harus tetap menjaga sistem jaringan di UGM dari ancaman tersebut. Untuk menjaga kinerja sistem yang berjalan agar bekerja secara semestinya, diperlukan tindakan pencegahan, pemeliharaan dan penanganan terhadap segala ancaman yang ada pada sistem informasi.

DSSDI UGM telah memasang sensor Honepot yang terintegrasi dengan *Modern Honey Network* (MHN) guna melihat ancaman-ancaman yang ada jaringan publik, sehingga dengan data yang diperoleh dari *Honeypot* administrator jaringan dapat melakukan tindakan pencegahan. *Honeypot* merupakan sebuah sistem yang menyerupai sistem asli di mana pemasangan sistem palsu tersebut bertujuan untuk menjebak pengguna yang bertujuan buruk [1]. Kemudian MHN digunakan untuk mengumpulkan data yang telah tercatat pada tiap sensor *Honeypot* yang sudah di integrasikan.

Sensor *Honeypot* yang digunakan oleh DSSDI UGM dipasang pada perangkat Raspberry Pi yang merupakan mini PC. Karena sumber daya yang dimiliki oleh Raspberry Pi ini relatif kecil mengakibatkan perangkat mudah mengalami *overload* sehingga data serangan yang masuk ke sistem *Honeypot* tidak tercatat oleh sensor. Selain masalah pada perangkat sensor *Honeypot*, *server* MHN juga bisa mengalami masalah pada penerimaan data yang tidak tersimpan pada basis data MHN yang disebabkan *storage* pada *server* MHN penuh.

Adanya masalah yang berkaitan dengan sumber daya pada sistem perlu dilakukan manajemen dan pemantauan sumber daya yang digunakan untuk menjaga sistem tetap berjalan sebagaimana mestinya. *Network Management System* (NMS) yang bermanfaat untuk memantau dan mengelola jaringan bisa menjadi solusi untuk masalah tersebut. Ada banyak cara untuk menerapkan NMS, salah satunya adalah dengan menggunakan protokol *Simple Network Management Protocol* (SNMP). Protokol tersebut bisa memenuhi kebutuhan pemantauan sistem. Selain berguna dalam pemantauan dan manajemen sistem, NMS dengan SNMP juga bisa digunakan untuk kebutuhan analisis data performa sistem. Penerapan NMS dengan SNMP ini akan sangat bermanfaat bagi administrator jaringan karena dapat membantu dan mempermudah dalam mengelola sistem jaringan.

II. TEORI PENDUKUNG

2.1. Keamanan Jaringan

Keamanan jaringan adalah segala aktivitas yang ditujukan untuk melindungi fungsi dan integritas data dan jaringan. Hal tersebut mencakup segala teknologi perangkat keras dan perangkat lunak yang terpasang. Pengelolaan akses ke jaringan dengan menargetkan berbagai ancaman dan mencegah serangan masuk atau menyebar pada jaringan merupakan tujuan dari keamanan jaringan [2].

Keamanan jaringan pada suatu jaringan dapat diuji dengan metode *Information Systems Security Assessment Framework* (ISSAF). Metode tersebut memiliki struktur yang jelah dan intuitif sehingga dapat digunakan untuk menguji keamanan sistem jaringan secara optimal. Dari hasil pengujian keamanan sistem jaringan administrator dapat mengetahui celah keamanan yang ada dan kemudian

melakukan tindakan antisipasi untuk menghadapi ancaman yang ada [3].

Open Source Security Information Management (OSSIM) merupakan sistem yang menggabungkan *tool* keamanan dalam satu paket. OSSIM yang dipasang pada jaringan digunakan untuk mengolah dan menganalisa data *traffic* atau aktivitas yang berbahaya pada jaringan. Dari hasil pengolahan data tersebut, didapat informasi yang bisa membantu admin dalam mengamankan jaringan [4].

2.2. *Honeypot*

Honeypot adalah suatu sistem palsu atau layanan palsu yang sengaja dibentuk untuk menjebak pengguna yang mempunyai tujuan buruk atau mendeteksi adanya usahausaha yang dapat merugikan sistem atau layanan. Biasanya *Honeypot* terdiri dari komputer, aplikasi, dan data yang menyerupai perilaku sistem nyata yang tampak menjadi bagian dari jaringan tetapi sebenarnya terisolasi dan terpantau [1].

Honeypot dipasang dengan tujuan mencatat setiap serangan yang masuk ke dalam sistem *Honeypot*. Kegiatan tersebut pasti menghasilkan *file log*, *file log* tersebut pasti akan semakin bertambah banyak seiring dengan semakin banyaknya serangan yang masuk dan akan memenuhi storage dari sistem *Honeypot*. Apabila log sampai memenuhi *resource* maka sistem tidak bisa berjalan sebagaimana mestinya. *File log* yang berukuran besar juga sulit untuk dianalisis dan memakan banyak waktu. Untuk mengatasi hal tersebut diperlukan pengelolaan file log yang efektif dan efisien [5].

2.2.1. Low Interaction *Honeypot*

Honeypot ini adalah *Honeypot* yang dibuat menyerupai sistem atau layanan pada *server* nyata, biasanaya hanya meyerupai layanan tertentu. Peretas atau penyerang hanya bisa memeriksa satu atau beberapa bagian saja pada jaringan tersebut dan tidak dapat berinteraksi langsung dengan sistem operasi yang digunakan, namun dengan interaksi secara tidak lansung maka informasi yang didapat cukup terbatas. Sistem ini bersifat seperti IDS yang hanya mendeteksi serangan masuk. Contoh dari jenis *Honeypot* ini misalnya Dionaea, Honeyd, Kippo, Glastopf, Snort dan lain-lain.

2.2.1.1. Dionaea

Dionaea merupakan salah satu *Honeypot* yang sering digunakan dan termasuk dalam tipe low interaction *Honeypot*, *Honeypot* ini bertujuan untuk mendeteksi serngan berupa malware yang disusupkan oleh penyerang.

Dionaea memiliki log yang mencatat aktivitas serangan pada jaringan seperti informasi asal alamat serangan, port tujuan yang diserang protokol yang layanan yang diserang. Selain itu Dionaea juga akan menyimpan berkas malware yang disusupkan oleh penyerang. Berkas malware yang didapat dapat dianalisa tool lain untuk mengetahui jenis dan tujuan malware tersebut [6]. Dionaea juga dapat digunakan untuk mendeteksi port scanning pada suatu jaringan, dari pola port scanning yang diperoleh administrator dapat memberikan tindakan keamanan lebih efisien [7].

2.2.1.2. Kippo

Honeypot yang khusus digunakan untuk mendeteksi serangan pada protokol SSH. Biasanya Kippo digunakan untuk mendeteksi adanya brute force pada suatu sistem. Tidak hanya mencatat serangan, Kippo juga dapat mempelajari pola serangan dengan bantuan tool lain. *Honeypot* ini dapat diterapkan menggunakan perangkat berspesifikasi rendah seperti Raspberry Pi. Kippo akan mencatat alamat IP sumber serangan dan juga mencatat user dan password yang digunakan untuk percobaan akses ke layanan SSH Kippo [8].

2.2.1.3 Glastopf

Ancaman yang ada pada layanan web dapat dideteksi dengan memanfaatkan Glastopf. Glastopf adalah *Honeypot* dengan tinkat interaksi rendah [9]. Implementasi Glastopf yang dikombinasikan dengan HIHAT dapat dimanfaatkan untuk mengetahui tujuan dan parameter pada HTTP *request*. Halaman web dan sistem informasi palsu yang dimiliki Glastopf dan HIHAT dugunakan untuk menjebak penyerang dan melihat *request* yang dikirim ke sistem palsu tersebut oleh penyerang [10].

2.2.2. High Interactions *Honeypot*

Honeypot yang dibuat agar penyerang berinteraksi langsung dengan sistem operasi atau layanan serta tidak ada batasan dalam interaksi tersebut. Karena interaksi langsung antara penyerang dengan sistem menyebabkan *Honeypot* ini memiliki risiko yang sangat tinggi, namun diisamping risiko tersebut bisa didapat banyak informasi tentang serangan yang masuk. Dari risiko yang ada pada penggunaan *Honeypot* ini maka perlu perlakuan ekstra dalam pengelolaanya. Hal tersebut bisa diatasi dengan subuah jail, sanbox atau VMware box karena dengan s*oft*ware ini akan mengisolasi *Honeypot* tersebut. Contoh dari *Honeypot* ini adalah Honeynet.

2.3. Modern Honey *Network*

Modern Honey *Network* (MHN) merupakan *server* terpusat untuk manajemen dan pengumpulan data *Honeypot*. MHN mempermudah dan mempercepat proses pemasangan sensor *Honeypot* karena didalam MHN sendiri sudah terdapat sekrip untuk pemasangan sensor *Honeypot* seperti Snort, Dionaea, Glastopf, Kippo dan lain-lain [11].

MHN diintegrasikan dengan beberapa sensor *Honeypot* seperti Dionaea, Kippo, Snort, Glastopf dan lain sebagainya. Sensor *Honeypot* dipasang pada perangakat RasberryPi yang terhubung dengan jaringa publik. Data serangan yang didapat dari sensor kemudian ditampilkan pada *interface* MHN. Dari pemasangan MHN tersebut telah didapat data serangan seperti asal serangan, tujuan serangan, port tujuan serangan, user dan password yang digunakan untuk menyerang serta jumlah serangan yang dilakukan [12]

2.4. Raspberry Pi

Raspberry Pi merupakan mini PC seukuran kartu kredit yang dapat langsung di hubungkan dengan monitor dan perangkat keyboard dan mouse. Selain ukurannya yang kecil, harga dari perangkat ini juga lebih murah dibanding dengan komputer pada umumnya. Sudah banyak proyek digital yang menggunakan perangkat ini. Salah satu seri yang banyak digunakan Raspberry Pi 3 Model B. Seri tersebut memiliki sepesifikasi CPU Quad Core 1.2 GHz, 1

GB RAM, 1 port FastEthernet, port HDMI dan beberapa fitur lain [13].

Perangkat Raspberry Pi dapat dimanfaatkan untuk penggunaan komputasi kecil sampai sedang. Seperti penggunaan Raspberry Pi sebagai IDS pada sebuah jaringan. Perangkat Raspberry Pi digunakan untuk membandingkan performa anatara Snort dan Suricata. Dari pengujian yang dilakukan menujukkan bahwa perangkat Raspberry Pi model B+ dapat menangani lebih dari 10000 rules [14].

2.5. *Network Management System* (NMS)

Network Management System (NMS) merupakan aplikasi atau sistem yang memungkinkan administrator jaringan mengelola komponen independen jaringan di dalam kerangka kerja manajemen jaringan yang lebih besar. NMS dapat digunaakan untuk memantau baik perangkat lunak maupun komponen perangkat keras yang ada dalam suatu jaringan. Biasanya digunakan untuk mencatat data dari jaringan yang dipantau kemudian data tersebut diteruskan ke sistem administrator [15].

Manfaat utama dari NMS adalah memungkinkan pengguna untuk memantau atau mengelola seluruh sistem operasi jaringan menggunakan perangkat komputer yang terpusat. NMS sangat berguna untuk kebutuhan deteksi perangkat, pemantauan, analisis performa, manajemen dan notifikasi pada jaringan.

International Organizationn for Standardization (ISO) mendefinisikan bahwa terdapat lima area fungsional dalam manajemen jaringan. Lima area tersebut manajement tersebut adalah sebagai berikut [16] :

1. *Fault Management* – manajemen untuk deteksi, isolasi, peringatan dan perbaikan kesalahan yang ada pada jaringan.
2. *Configuration Management* – manajemen pada aspek konfigurasi perangkat jaringan seperti manajemen file konfigurasi, manajemen inventaris, dan manajemen perangkat lunak.
3. *Performance Management* — Pantau dan ukur berbagai aspek kinerja sehingga keseluruhan kinerja dapat dipertahankan pada tingkat yang dapat diterima.
4. *Security Management* — Menyediakan akses ke perangkat jaringan dan sumber daya perusahaan untuk individu yang berwenang.
5. *Accounting Management* — Informasi penggunaan sumber daya jaringan.

Monitoring pada sistem jaringan sangatlah penting untuk dilakukan karena pentingnya menjaga kualitas dan ketersediaan jaringan. Salah satu cara melakukan hal tersebut adalah dengan monitoring pada jaringan adalah dengan menggunakan protokol *Simple Network Management Protocol* (SNMP), seperti penelitian yang berjudul "Implementasi dan Analisis Sistem Monitoring Performance Jaringan dengan Parameter *Quality Of Service* (QoS)". Pada penelitian tersebut dilakukan pemantauan pada jaringan untuk melihat kualitas performa jaringan berdasarkan parameter QoS, pengambilan data besar *troughput* jaringan, *jitter*, *latency* dan packet loss digunakan untuk menentukan kualitas dari QoS jaringan [17]. Yuli Sholikatin juga melakukan penelitian tentang pemantaua jaringan dengan menggunakan SNMP guna kebutuhan *Fault Management*, dari penelitian tersebut didapat kesimpulan bahwa NMS dengan SNMP dapat menampilkan informasi penggunaan resource perangakat yang dipantau dan dapat memberi pemberitauan tentang kejadian error pada sistem [18].

2.6. *Simple Network Management System* (SNMP)

Simple Network Management Protocol (SNMP) merupakan sebuah protokol yang dugunakan untuk mengkoleksi dan mengatur informasi pada perangkat jaringan yang dikelola. SNMP sering digunakan pada manajemen jaringan untuk memantau jaringan. SNMP dapat digunakan pada jaringan yang besar dan kompleks. Secara manual dan individual pencatatan dan pemantauan banyak perangkat akan memakan waktu yang cukup lama, namun dengan menggunakan SNMP administrator jaringan dapat mengelola dan memantau sermua node jaringan dari satu antarmuka [19].

Pada SNMP terdapat tiga komponen pokok yang diperlukan yaitu sebagai berikut (Wilkins, 2011):

1. SNMP *Agent* - program ini berjalan pada node yang dipantau, mengumpulkan data tentang berbagai metrik seperti penggunaan *bandwidth* atau *memory*. Agen akan mengirimkan data pada SNMP manager ketika diminta. Agen juga dapat mengirim peringatan secara proaktif ke manajer ketika terjadi kesalahan.
2. SNMP Manager / *Network Management* Stations (NMS) – SNMP manager akan meminta kepada agen untuk mengirim informasi dari node melalui SNMP secara berkala. NMS akan mengumpulkan dan mengontrol data dari agen secara terpusat.
3. *Management* Information Base (MIB) – merupakan basis data yang berbentuk file teks (.mib) yang merinci dan menjelaskan semua objek yang digunakan oleh perangkat tertentu yang dapat diminta atau dikendalikan menggunakan SNMP. Basis data ini harus dimuat ke NMS sehingga dapat mengidentifikasi dan memantau status dari node pada jaringan. Setiap item MIB diberi pengidentifikasi objek (OID).

SNMP berjalan pada lapisan aplikasi. Semua pesan SNMP dikirim melalui UDP. SNMP *agent* akan menerima *request* pada port 161 UDP. NMS bisa mengirim *request* ke agen melalui port tersebut, kemudian agen akan merespon dengan mengirim informasi sesuai dengan permintaan NMS. NMS juga bisa menerima notifikasi (*Traps* dan *InformRequest*) dari *agent* melalui port 162.

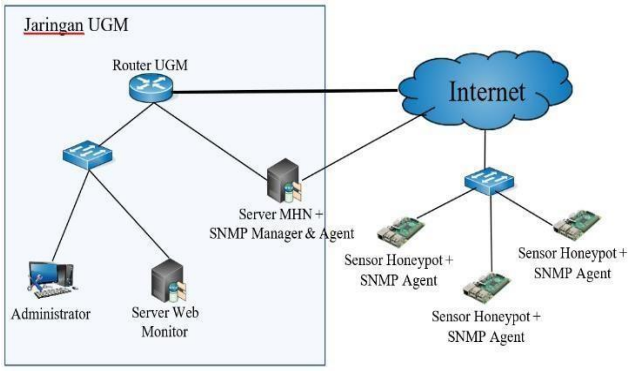
III. METODOLOGI DAN PERCOBAAN

Pada penelitian ini dilakukan dengan beberapa tahap. Tahap pertama pada penelitian ini adalah perancangan topologi dari sistem pemantauan. Topologi yang digunakan menyesuaikan dengan sistem *Honeypot* yang telah terpasang di DSSDI UGM. Selanjutnya adalah instalasi protokol SNMP yang akan digunakan sebagai protokol pemantau sistem. Setelah SNMP sudah dipasang dan diuji, dependensi yang diperlukan seperti web *server*, basis data dan Python konfigurasi pada *server* pemantauan. Kemudian merancang dan mengbangun sistem pemantauan pada *server* yang sudah dipersiapkan sebelumnya.

3.1. Perancangan Topologi

Sistem pemantauan pada penelitian ini dilakukan pada jairngan UGM untuk manajemen sistemnya, kemudian node yang dipantau berada pada jaringan publik seperti Gambar 1. Node yang dipantau adalah perangkat Raspberry Pi yang sudah terpasang sensor *Honeypot*. Dari sistem *Honeypot*

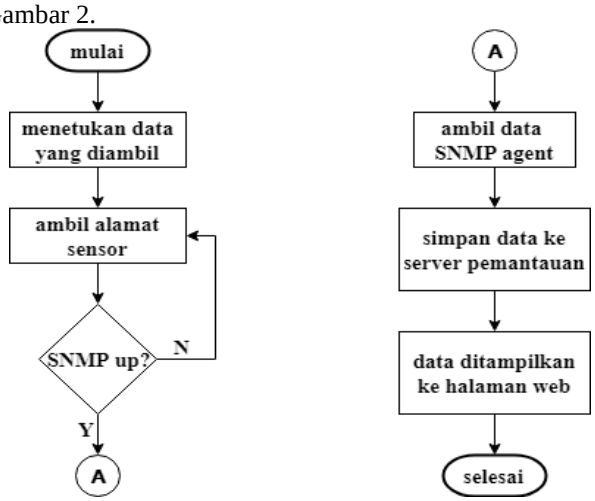
yang sudah ada di DSSDI UGM terdapat dua buah perngkat sensor *Honeypot*, masing-masing perangkat tersebut memiliki alamat IP publik xxx.xxx.92.50 dan xxx.xxx.92.54 sehingga sensor tersebut dapat menerima serangan dari jaringan publik. Dua perangkat sensor tersebut tepasang lebih dari satu *Honeypot*, *Honeypot* yang terpasang pada perangkat tersebut diantaranya Dionaea, Kippo, Glastopf pada sensor "madu-DSSDI-1-UGM" dan Dionaea, Kippo pada sensor "madu-DSSDI-2-UGM".



Gambar 1. Topologi sistem pemantauan

Server MHN yang telah terpasang memiliki dua antarmuka jaringan dimana satu antarmuka dengan alamat IP publik dugunakan untuk komunikasi antara MHN dengan sensor *Honeypot* dan satu antarmuka lainya dugunkan untuk manajemen *server* MHN yang berada pada jaringan UGM dengaan alamat IP xxx.xxx.245.52. Pada penelitian ini menggunakan tambahan *server* yang digunakan untuk *server* pemantau dikarenakan load pada *server* MHN sangat besar. *Server* pemantau tersebut digunakan untuk menampilkan data pemantauan performa sistem *Honeypot* dimana berada pada jaringan UGM dengan alamat IP xxx.xxx.109.68.

3.2. Pembuatan Sistem Pemantauan Informasi dari performa dari sensor *Honeypot* dikumpulkan *server* MHN yang terpasang yang menjadi *SNMP manager*. Manajer akan meminta informasi dari sensor *Honeypot* tiap 5 menit. Informasi-informasi yang telah dikumpulkan oleh *server* MHN akan disimpan ke dalam basis data, sebelum masuk ke basis data, informasi tersebut perlu disusun dan disesuaikan dengan kebutuhan agar dalam pemanfaatannya lebih mudah dan efisien. Alur dari sistem pemantauan dapat dilihat pada Gambar 2.



Gambar 2. Diagram alir sistem pemantauan

Server MHN sebagai manajer akan meminta informasi ke agen. Manajer menentukan OID atau MIB yang akan dimintakan datanya ke agen. Agen akan mengirimkan data sesuai dengan MIB atau OID yang diminta.

Daftar alamat sensor diambil dari database MHN. Semua sensor atau *host* yang ada pada daftar tesebut akan diambil informasi performanya dengan SNMP secara satu-persatu. Sebelum data diambil, tiap *host* akan diperiksa terlebih dahulu apakah host tersebut aktif atau tidak. Apabila *host* tidak aktif maka akan melakukan pemeriksaan ke *host* berikutnya dan apabila host tersebut aktif maka manajer akan meminta informasi performa *host*. Setelah data dari tiap *host* diperolah, data tersebut akan disimpan ke dalam basis data MongoDB.

Data yang telah dikumpulkan oleh manajer dan tersimpan akan diolah dan kemudian ditampilkan pada halaman *web* pemantauan. Dari pemantauan perangkat sensor *Honeypot* tersebut, data yang diperoleh akan dapat digunakan untuk untuk melakukan manajemen *resource*. Data pemantauan yang dikombinasikan dengan data serangan *Honeypot* dapt dianalisis secara lebih lanjut untuk mengetahui pengaruh jumlah serangan terhadap performa sistem *Honeypot*.

IV. ANALISIS DAN PEMBAHASAN

Hasil dari pemantauan sistem *Honeypot* menunjukkan sumberdaya yang dimiliki perangkat sensor *Honeypot* dan penggunaannya. Selain itu analisis data dapat dilakukan dengan informasi yang diperoleh dari hasil pemantauan tersebut.

4.1. Hasil Pemantauan Sistem *Honeypot*

Informasi tentang *resource* perangkat dapat dilihat pada sistem pemantauan ini. Pada sistem pemantauan ini akan memperlihatkan keadaan dan penggunaan *resource* dari tiap sensor *Honeypot* yang terpasang. Dari haril pemantauna yang telah dilakukan menunjukkan bahwa terdapat 9 sensor terintegrasi dengan MHN di DSSDI UGM yang terlihat pada halaman web pemantauan seperti pada Gambar 3. Dalam tampilan tersebut terlihat beberapa sensor yang dengan tampilan berwarna hijau yang berarti sensor tersebut aktif, sedangkan sensor yang berwarna merah menandakan bahwa sensor tersebut tidak aktif.

Gambar 3. Indikatr sensor

Tombol yang ada pada pemantauan tersebut memperlihatkan terdapat 2 buah sensor *Honeypot* aktif yang ditunjukkan dengan tombol yang berwarna hijau dan sedangkan lainnya tidak aktif yang ditunjukkan dengan tombol berwarna merah. Dari 2 buah sensor yang aktif masing-masing adalah sensor "madu-DSSDI-1-UGM" yang berada pada alamat IP xxx.xxx.92.50 dan sensor "maduDSSDI-2-UGM" yang berada pada alamat IP xxx.xxx.92.54. Pada Gambar 3 juga memunjukkan sensor-sensor yang tidak aktif, salah satunya "sesnsorFK-1-UGM" yang berada pada alamat IP xxx.xxx.93.214.

Informasi dari penggunaan memori, *storage* dan beban CPU dapat teramati secara detail. Pada pemantauan ini informasi *resource* tiap sensor dapat terpantau, dari hasil pemantauan yang telah diamati didapat informasi yang dipaparkan pada Tabel 1 dan Tabel 2. Tabel tersebut menunjukkan informasi penggunaan resource dari

perangankat sensor *Honeypot* "madu-DSSDI-1-UGM" dan "madu-DSSDI-2-UGM" yang diamati pada tanggal 26 Juli 2018 pukul 11.45.

Tabel 1. Pemantauan resource "madu-DSSDI-1-UGM"

Pengamatan		Pemakaian	Kapasitas	Persentase (%)
CPU (%)		1	100	1,0
Memori (MB)	Fisik	914,91	927,16	98,7
	Virtual	964,13	1027,15	93,9
	Buffer	70,71	927,16	7,6
	Cache	135,06	135,06	100,0
	Swap	4392,21	100	4392,2
	/	0	7458,86	0,0
	/Dev	2,37	114,85	2,1
	/Boot	0,75	6,99	10,7
Strorage (MB)		17568,8	29835,44	58,9

Tabel 2. Pemantauan resource "madu-DSSDI-2-UGM"

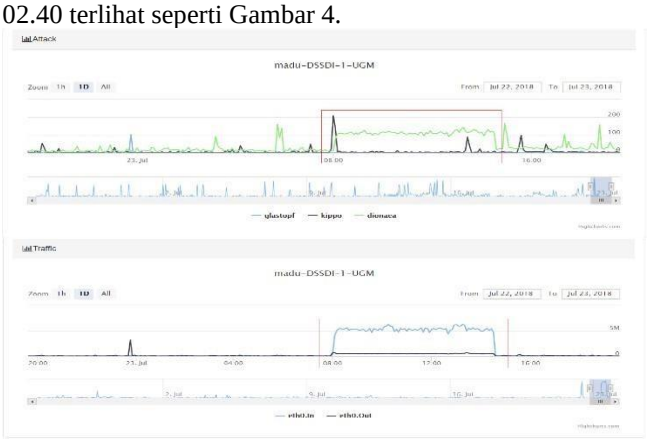
Pengamatan		Pemakaian	Kapasitas	Persentase (%)
CPU (%)		1	100	1,0
Memori (MB)	Fisik	901,91	927,16	97,3
	Virtual	910,47	1027,15	88,6
	Buffer	66,74	927,16	7,2
	Cache	101,05	101,05	100,0
	Swap	2340,41	100	2340,4
	/	0	7458,86	0,0
	/Dev	2,37	114,85	2,1
	/Boot	0,75	6,99	10,7
Strorage (MB)		9361,65	29835,44	31,4

Berdasarkan informasi pada Tabel 1 yang menampilkan informasi perangkat sensor "madu-DSSDI-1-UGM" tersebut, menunjukkan penggunaan memori fisik dan virtual pada perangkat sangat tinggi yang mencapai 98.7% dan 93.9%. Beban CPU pada perangkat sensor tidak begitu besar hanya mencapai 1%. Pada penggunaan *storage* masih tersisa banyak ruang, *storage* sudah terisi sebesar 58,9%.

Informasi resource pada perangkat sensor "madu-DSSDI2-UGM" yang ditampilkan pada Tabel 2 telihat penggunaan resource yang hampir sama dengan sensor "madu-DSSDI-1UGM". Hal tesebut ditunjukkan dengan nilai dari beban CPU sebesar 1%, memori fisik 97,3%. Penggunaan *resource* yang mencapai batas maksimal dapat mengakibatkan sistem berjalan lambat dan bahkan dapat menyebabkan kegagalan layanan. Apabila penggunaan *resource* mencapai batas maksimal secara terus menerus, perlu dilakukan pengelolaan *resource*.

4.2. Analisis Pengaruh Jumlah Serangan pada *Traffic*

Pengamatan pengaruh jumlah serangan terhadap *traffic* antarmuka jaringan sensor *Honeypot* dilakukan dengan melihat grafik jumlah serangan dan grafik *traffic*. Adanya peningkatan atau penurunan pada grafik diamati pada kedua grafik tersebut dengan melihat pada waktu yang sama. Grafik yang yang diamati pada tanggal 24 Juli 2018 pukul 02.40 terlihat seperti Gambar 4.



Gambar 4. Perbandingan grafik serangan dan *traffic* 24 juli 02.40

Pada Gambar 4 yang merupakan tampilan grafik jumlah serangan dan *traffic* dari sensor *Honeypot* "madu-DSSDI1UGM". Grafik tersebut menunjukan adanya kenaikan jumlah serangan yang signifikan dan kenaikan terjadi juga pada *traffic* antarmuka jaringan. Kenaikan jumlah serangan yang signifikan terjadi pada pukul 15.05 tanggal 23 Juli 2018. Kenaikan jumlah serangan tersebut juga diikuti kenaikan pada *traffic* sensor *Honeypot*. Kenaikan jumlah serangan tersebut bermula pada pukul 15.00 dimana terdapat jumlah serangan pada sensor Dionaea 11 serangan, Kippo 208 serangan dan Glastopf 0 serangan, dari serangan tersebut diikuti dengan kenaikan pada *traffic* antarmuka dengan nilai *traffic* masuk 635.068 Byte dan *traffic* keluar sebesar 286.438. detail dari kenaikan jumlah serangan dan *traffic* dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 3. Peningkatan jumlah serangan

Waktu	Dionaea	Kippo	Glastopf
14.50	9	3	0
14.55	9	20	0
15.00	11	208	0
15.05	77	136	0
15.10	104	5	0
15.15	109	9	0
15.20	110	2	1

Tabel 4. Peningkatan *traffic*

Waktu	*Traffic* Masuk (Byte)	*Traffic* Keluar (Byte)
14.50	18.899	30.128
14.55	37.934	75.388
15.00	286.438	635.065
15.05	3.965.887	714.208
15.10	4.962.856	454.885
15.15	5.399.166	506.529
15.20	5.433.565	528.003

Dari detail data tersebut bahwa adanya peningkatan jumlah serangan pada sesnsor Kippo dan Dionaea. Pada saat peningkatan jumlah serangan pada Kippo juga diikuti

peningkatan pada *traffic* masuk dan keluar yaitu pada pukul 14.55,15.00 dan 15.05. Peningkatan serangan yang signifikan terjadi pada sensor Dionaea mengakibatkan peningkatan juga pada traffic, kenaikan *traffic* terlihat jelas pada *traffic* masuk. Kenaikan *traffic* tertinggi terjadi pada pukul 15.05 yang mencapai angka 3.68 MB untuk traffic masuk, jumlah serangan yang meningkat drastis pada waktu tersebut adalah jumlah serangan dari sensor Dionaea yaitu dari 11 serangan menjadi 77 serangan dalam jangka waktu 5 menit. Peningkatan *traffic* keluar tertinggi tercatat pada pukul 15.00 dengan kenaikan *traffic* keluar sebesar 559.68 KB dimana jumlah serangan pada sensor Kippo mencapai 208 serangan yang sebelumnya hanya 20 serangan.

Traffic antarmuka jairngan tidak hanya dipengaruhi jumlah serangan saja, perlu diamati jumlah serangan pada tiap sensor secara seksama. Pada beberapa sampel yang diambil pada waktu tertentu dapat diamati tingkat pengaruh jumlah serangan terhadap traffic yang ada pada antarmuka jaringan. Perhitungan pengaruh tersebut dilakukan dengan cara membagi kenaikan yang terjadi pada traffic dengan kenaikan jumlah serangan. Perhitungan ini merupakan cara yang masih kasar dan perlu dilakukan penelitian yang lebih lanjut. Pada Tabel 5, Tabel 6 dan Tabel 7 akan menunjukkan tingkat pengaruh dari masing masing sensor *Honeypot* terhadap traffic antarmuka jaringan yang diambil pada tanggal 14 Juli 18.00-15 Juli 17.55.

Tabel 5. Pengaruh Glastopf pada Traffic

Serangan	*Traffic In*	*Traffic Out*	In (Byte / Serangan)	Out (Byte / Serangan)
41	71618	1502831	1219,20	35818,76
42	79439	1677929	7821,00	175098,00
34	67385	1156317	1506,75	65201,50
90	107966	2775807	927,96	30405,32
103	98333	3175909	677,55	30390,21
Rata-rata pengaruh			2430,49	67382,76

Dari Tabel 5 di atas dapat dilihat bahwa rata-rata pengaruh dari tiap serangan Glastopf akan mengakibatkan penambahan *traffic* data kurang lebih sebesar 67 KB dan pada traffic keluar dan 2 KB pada *traffic* masuk. Pengarunh Glastopf pada *traffic* keluar disebabkan karenaa Glastopf merupakan *Honeypot* yang dapat memberikan respon ke penyerang sehingga akan ada data yang dikirim keluar pada *traffic* jaringan sebagai bentuk respon dari Glastopf

Tabel 6. Pengaruh Kippo pada Traffic

Serangan	*Traffic In*	*Traffic Out*	In (Byte / Serangan)	Out (Byte / Serangan)
83	123411	255752	1221,21	2643,17
94	70412	141022	467,7	1040,09
100	254802	317965	25121,2	7901,6
76	152205	216140	1734,2	2383,42
39	58256	122011	949,17	2163,22
Rata-rata pengaruh			5898,70	3226,30

Dari perhitungan yang dilakukan menunjukkan setiap serangan pada sensor Kippo akan mempengaruhi traffic masuk sebesar 5 KB dan 3 KB pada *traffic* keluar berdasarkan rata-rata yang telah dihitung. Nilai tersebut lebih kecil dibanding dengan peangaruh dari serangan Glastopf.

Tabel 7. Pengaruh Dionaea pada Traffic

Serangan	*Traffic In*	*Traffic Out*	In (Byte / Serangan)	Out (Byte / Serangan)
56	47178	84106	360,46	175,17
163	226335	411510	3556,55	1163,14
35	79439	1677929	279,32	6253,5
Rata-rata pengaruh			1398,78	2530,60

Hasil perhitungan serangan pada sensor Dionaea pada tabel diatas tidak mempelihatkan pengaruh serangan yang cukup besar terhadap *traffic* antarmuka jaringan. Namun pada Tabel 7 terlihat peningkatan yang signifikan pada *traffic* antarmuka jaringan yang mengikuti kenaikan jumlah serangan sensor Dioanea. Hal dapat terjadi dikarenakan sernsor Dionaea yang berfungsi sebagai pendeteksi malware, biasanya Dionaea akan mendownload berkas malware yang dikirim penyerang untuk dianalisis.

4.3. Analisis Pengaruh Jumlah Serangan pada CPU

Serangan yang masuk ke *Honeypot* dilihat pada grafik kemudian dibandingkan dengan grafik penggunaan CPU. Kenaikan jumlah serangan yang masuk ke sensor *Honeypot* terlihat mempengaruhi penggunaan CPU perangkat. Terlihat pada Gambar 5 bahwa jumlah serangan mempengaruhi penggunaan CPU *user*. Dari grafik yang tertampil pada gambar tersebut merupakan grafik serangan pada sensor *Honeypot* "madu-DSSDI-1-
UGM" pada tanggal 25 Juli 2018, sebelum pukul 08.00 terdapat kenaikan jumlah serangan pada sensor Dionaea. Kenaikan tersebut diikuti dengan kenaikan penggunaan CPU pada waktu tersebut.

Gambar 5. Perbandingan grafik serangan dan *CPU* 25 Juli 2018 08.00

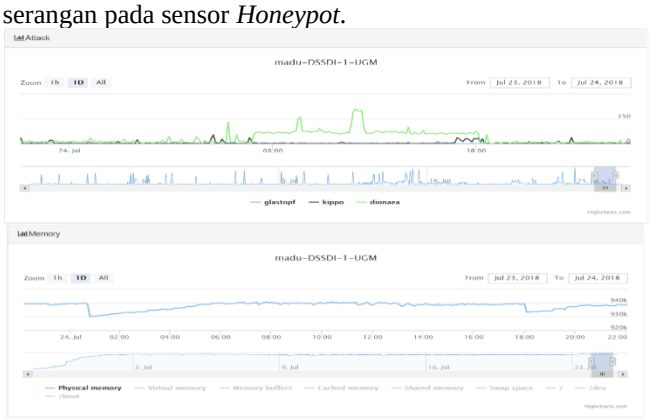
Pengkatan jumlah serangan yang mencapai 331 serangan pada sensor Dionaea diikuti dengan kenaikan penggunaan CPU user menjadi 8 %. Penggunan CPU sistem tidak mngalami perubahan yang tetap berada pada nilai 0. Jumlah serangan pada masing-masing sensor *Honeypot* terlihat memiliki pengaruh berbeda beda terhadap penggunaan CPU, seperti terlihat pada Gambar 6 yang merupakan grafik serangan dan penggunaan CPU dari sensor *Honeypot* "maduDSSDI-2-UGM".

Gambar 6. Perbandingan serangan dan *CPU* "madu- DSSDI - 2 - UGM"

Pada grafik yang diambil dari sensor "madu-DSSDI-2UGM" terlihat pada bagian yang ditandai terdapat peningkatan dari serangan dan penggunaan CPU. Pada sensor *Honeypot* tersebut terpasang dua buah sensor yaitu Kippo dan Dionaea. Dari grafik tersebut terlihat bahwa kenaikan penggunaan CPU pada tiap sensor berbeda.

4.4. Analisis Pengaruh Jumlah Serangan pada Memori

Penggunaan memori pada sistem sensor *Honeypot* dilihat pada grafik memori. Pada informasi memori terdapat beberapa jenis memori, yasitu diantaranya memori fisik, memori virtual, memori *swap*, memori *buffer*, memori *cache* dan lainya. Jumlah serangan yang masuk ke *Honeypot* dibandingkan dengan penggunaan memori dan dilihat memori mana yang berpengaruh terhadap jumlah serangan pada sensor *Honeypot*.



Gambar 7. Perbandingan serangan dan memori fisik

Dari jumlah serangan yang masuk ke *Honeypot* pada tanggal 24 terlihat adanya serangan yang cukup signifikan dan terjadi cukup lama. Pengamatan grafik dilakukan pada sensor "madu-DSSDI-1-UGM". Penggunaan memori fisik perangakat sensor tersebut mengalami kenaikan secara perlahan ketika ada peningkatan jumlah serangan kemudian turun setelah jumlah serangan turun. Pada grafik serangan terjadi peningkatan jumlah serangan pada pukul 6.00, pada saat yang sama tidak terjadi peningkatan penggunaan memori fisik pada perangkat. Tidak adanya peningkatan memori fisik pada perangakat dapat disebabkan karena memori fisik yang terpakai sudah mencapai batas maksimal. Hal ini terlihat dari grafik memori fisik yang sudah menigkat sebelumnya sampai batas maksimal, kanaikan grafik mulai berhanti pada pukul 5.30 dan setalah itu grafik terlihat stabil stabil. Hal tersebut menunjukkan bahwa penggunaan memori sudah mencapai batas maksimal. Kemudian grafik memori fisik turun pada saat beberapa jam setelah jumlah serangan turun. Hal tersebut juga terjadi pada grafik memori virtual.

Gambar 8. Perbandingan serangan dan memori *buffer*

Dari grafik memori *buffer* terjadi peningkatan secara drastis pada penggunaan memori *buffer* pada saat ada peningkatan jumlah serangan, penggunaan memori tersebut juga menurun secara drastis setelah jumlah serangan menurun. Hal tersebut terlihat pada grafik yang ditampilkan oleh Gambar 8. Dari grafik memori *buffer* tersebut menunjukkan pengkatan penggunaan memori terjadi bersamaan dengan peningkatan jumlah serangan pada perangkat sensor. Setiap terjadi peningkatan jumlah serangan yang signifikan, penggunaan memori *buffer* juga ikut meningkat. Penggunaan memori *buffer* turun pada saat satu jam setelah jumlah serangan menurun. Penggunaan memori *buffer* yang terpengaruh jumlah serangan pada *Honeypot* dikarenakan fungsi dari memori *buffer* yang berperan sebagai tempat penampungan sementara untuk data yang sedang dikirim atau diterima dari perangkat luar.

V. KESIMPULAN

Berdasarkan hasil pemantauan dan analisi performa dari sistem *Honeypot* dengan SNMP, didapat kesimpulan bahwa sistem pemantauan performa pada sistem *Honeypot* dapat dibangun dengan menggunakan SNMP untuk mengambil informasi sensor *Honeypot*. Dari sistem tersebut terlihat penggunaan memori pada perangkat sensor *Honeypot* telah melebihi kapasitas. Selain ituj umlah serangan yang masuk perangkat sensor *Honeypot* mempengaruhi performa dari perangkat sensor *Honeypot* yang terlihat dari peningkatan *traffic*, beban CPU dan penggunaan memori pada saat terjadi peningkatan jumlah serangan. Pengaruh jumlah serangan terhadap *traffic* antarmuka jaringan berbeda-beda tergantung dari jenis serangan yang masuk, serangan yang terdeksi pada sensor Glast*o*fp dapat menambah *traffic* data rata-rata 67 KB/serangan pada *traffic* keluar dan Kippo sebesar 6 KB/serangan, sedangakan pada sensor Dionaea tidak tentu karena sensor tersebut berfungsi deteksi *malware*.

DAFTAR PUSTAKA

[1] M. Rouse and M. Cobb, "*Honeypot*," Juni 2018. [Online]. Available: https://searchsecurity.techtarget.com/definition/ *Honeypot*. [Accessed 2 Juli 2018].

[2] Cisco, "Cisco," 2016. [Online]. Available: https://www.cisco.com/c/en/us/products/securit y/what -is-*Network*-security.html. [Accessed 13 Juli 2018].

[3] H. Bhagaskara, D. Adhipta and L. E. Nugroho, "Uji Penetrasi Sistem Keamanan jaringan Universitas Gadjah Mada dengan Information

System Security Assessment Framework (ISSAF)," Universitas Gadjah Mada,
Yogyakarta, 2015.

[4] B. Fredianto and Soedjatmiko, "Manajemen Keamanan Jaringan Informasi Menggunakan
OSSIM,"
Universitas Gadjah Mada, Yogyakarta, 2005.

[5] A. N. Singh and R. C. Joshi, "A *Honeypot System* for Efficient Capture and Analysis *of Network* Attack Traffic," IEEE, 2011.

[6] I. L. Pribadi, "Implementasi Sistem Keamanan Jaringan Menggunakan *Honeypot* Dionaea, IDS, Dan Cuckoo Sandbox," Universitas Telkom, Bandung, 2013.

[7] R. A. Habsoro and N. R. Rosyid, "Implementasi *Honeypot* Untuk Mengungkap Port Scanning Attacks dalam Jaringan," Universitas Gadjah Mada, Yogyakarta, 2015.

[8] J. V. Ho*of*, "Can a SSH *Honeypot* Be Used to
Attract
Attackers and Improve Security?," 29 September 2014. [Online].Available: https://securityintelligence.com/can-a-ssh*Honeypot*be-used-to-attract-attackers-andimprove-security/.
[Accessed 20 Juli 2018].

[9] L. Rist, "Know Your Tools: Glastopf - A dynamic, lowinteraction web application *Honeypot*," 15 November 2010. [Online].
Available:
https://www.honeynet.org/papers/KYT_glastop f. [Accessed 20 Juli 2018].

[10] I. Laksana and N. R. Rosyid, "Implementasi *Honeypot* Sebagai Pemantauan Parameter Pada Http *Request* Untuk Mengetahui Tujuan Serangan," Universitas Gadjah Mada,
Yogyakarta, 2017.

[11] Anomali, Inc., "Modern Honey *Network*,"
[Online]. Available: https://github.com/threatstream/mhn.
[Accessed 30 Juni 2018].

[12] W. Septian, W. Najib and S. Sumaryono, "Implementasi *Honeypot* Menggunakan Platform Modern Honey *Network* (studi Kasus Di Direktorat Sistem Dan Sumber Daya
Informasi, Universitas Gadjah Mada)," Universitas Gadjah Mada, Yogyakarta, 2017.

[13] RaspberryPi, "Raspberry Pi 3 Model B," 2016. [Online].

[14] J. Prakoso and A. K. Sari, "Perbandingan Performa Snort dan Suricata Sebagai Sistem Deteksi Intrusi pada Raspberry Pi," Universitas Gadjah Mada, Yogyakarta, 2018.

[15] Techopedia, "*Network Management System* (NMS)," 2018. [Online]. Available: https://www.techopedia.com/definition/11988/ networ k-*Management-System*-nms. [Accessed 30 Juni 2018].

[16] Cisco, "*Network Management System*: Best Practices White Paper," 11 Juli 2007. [Online].
Available:
https://www.cisco.com/c/en/us/support/docs/av ailabili ty/high-availability/15114-NMSbestpractice.html. [Accessed 30 Juni 2018].

[17] R. Lukitawati and A. Subardono, "Implementasi Dan Analisis Sistem Monitoring Performance Jaringan Dengan Parameter *Quality Of Service* (qos)," Universitas Gadjah Mada, Yogyakarta, 2017.

[18] Y. Sholikatin and N. R. Rosyid, "Implementasi *Fault Management* (manajemen Kesalahan)
Pada *Network Management System* (NMS) Berbasis SNMP," Universitas Gadjah Mada, Yogyakarta, 2017.

[19] M. Rouse, J. Scarpati, A. Ranjan, C. Karbinski and J. Mathew, "*Simple Network Management Protocol* (SNMP)," Januari 2018. [Online].
Available: https://search*Network*ing.techtarget.com/definit ion/SN MP. [Accessed 1 Juli 2018].

[20] S. Wilkins, "SNMP Concepts and Configuration," 20 Juli 2011. [Online]. Available:http://www.ciscopress.com/articles/ar ticle.asp?p=1730 888. [Accessed 2 Juli 2018].

ANALISIS PERFORMA BLUETOOTH PADA SISTEM DEVICE REMINDER BERDASARKAN PENGUKURAN JARAK DAN RECEIVED SIGNAL STRENGTH INDICATOR

Rafiqmia Khairuddin Nur Hammam, Hidayat Nur Isnianto
Departemen Teknik Elektro dan Informatika
Sekolah Vokasi
Universitas Gadjah Mada
rafiqmia.khairuddin@mail.ugm.ac.id, hnisnianto@ugm.ac.id

Abstract - Internet of Things (IoT) is one of technology that is currently being developed and widely applied in various sectors. That is supported by the development of IoT device infrastructure which is increasingly sophisticated and modern so its capable to meet various needs in its application. IoT can also be applied as a tool to assist one's activities in daily life within scope of a private network which are called Personal Area Network (PAN), one of them as a reminder tool. Reminders can be used to remind a person on an important condition such as stuff that drop behind. Sometimes a person forgets to put his stuff so it left behind in a place, and it will bring up a risk that is loss of stuff. To help the problem is required a reminder device that is expected to minimize the risk that may occur. This device applies a point to point communication consisting of one transmitter module that is Bluetooth Low Energy AT-09 and one receiver module that is Android smartphone. This reminder device will activate the alarm on Android smartphone when both modules are spaced more than 5 meters. Then, from the implementation of this system will do an analysis of quality of service based on the value of Received Signal Strength Indicator (RSSI) and the distance between the two modules. Quality of service's parameters which will analyze is delay and packet loss. Based on test results, signal strenght value (RSSI) and packet loss parameters are strongly influenced by barricade media, because when both modules are in unobstructed condition they are indicates a stronger signal strength value and few packet loss occurs. While for the delay parameter is influenced by the data rate transfer of bluetooth devices. *Keywords : RSSI, Distance, Bluetooth Low Energy, Quality of Service, Reminder*

Intisari - ***Internet of Things (IoT)*** **merupakan salah satu teknologi yang saat ini sedang dikembangkan dan banyak diterapkan di berbagai bidang. Hal tersebut didukung oleh perkembangan infrastruktur perangkat IoT yang semakin canggih dan modern sehingga mampu memenuhi berbagai macam kebutuhan dalam penerapannya. IoT juga dapat diterapkan sebagai alat untuk membantu aktifitas seseorang di kehidupan seharihari dalam lingkup jaringan personal yang disebut *Personal Area Network* (PAN), salah satunya yaitu sebagai alat pengingat *(reminder)*. *Reminder* dapat digunakan untuk mengingatkan seseorang pada suatu keadaan yang penting seperti tertinggalnya barang bawaan. Terkadang seseorang lupa meletakkan barang bawaannya sehingga tertinggal di suatu tempat, dan hal tersebut akan memunculkan suatu resiko yaitu kehilangan barang. Untuk membantu permasalahan tersebut maka dibutuhkan sebuah perangkat *reminder* yang diharapkan dapat meminimalisir resiko yang mungkin terjadi. Perangkat ini menerapkan komunikasi *point to point* yang terdiri dari 1 modul pemancar *(transmitter)* berupa *Bluetooth Low Energy* AT-09 dan 1 modul penerima *(receiver)* berupa *smartphone* Android. Perangkat *reminder* ini akan mengaktifkan alarm pada *smartphone* Android saat kedua modul berjarak lebih dari 5 meter. Kemudian, dari penerapan sistem ini dilakukan analisis terhadap *quality of service* berdasarkan nilai *Received Signal Strength Indicator* (RSSI) dan jarak antara kedua modul. Parameter *quality of service* yang akan dianalisis meliputi parameter *delay* dan *packet loss*. Berdasarkan hasil pengujian, nilai kuat sinyal (RSSI) dan parameter *packet loss* sangat dipengaruhi oleh media penghalang, karena saat kedua modul berada dalam kondisi tidak terhalang menunjukkan nilai kuat sinyal yang lebih stabil dan sedikit sekali terjadi *packet loss*. Sedangkan untuk parameter *delay* dipengaruhi oleh kecepatan transfer data dari perangkat bluetooth yang digunakan.**
Kata Kunci : RSSI, Jarak, *Bluetooth Low Energy, Quality of Service, Reminder*

I. PENDAHULUAN

Manusia merupakan makhluk yang diciptakan dengan dibekali akal. Dengan menggunakan akal inilah manusia dapat membentuk ataupun mengubah sifat dan kepribadian dalam dirinya, sehingga manusia mampu memiliki beragam sifat. Namun, dari beragam sifat manusia, ada satu sifat dasar yang pasti dimiliki oleh setiap orang yaitu sifat lupa. Lupa memiliki pengertian ketidakmampuan mengenal atau mengingat sesuatu yang pernah dialami atau dipelajari. Setiap orang tidak dapat terhindar dari sifat tersebut dan setiap orang pasti pernah lupa akan suatu hal, salah satunya yaitu lupa untuk membawa barang yang
penting ataupun lupa saat meletakkan barang penting tersebut sehingga tertinggal di suatu tempat. Baik barang yang berukuran kecil seperti kunci, dompet, handphone maupun barang yang berukuran besar seperti tas, yang mana barang-barang tersebut merupakan barang yang sering dibawa saat beraktifitas. Saat seseorang lupa meletakkan barang penting miliknya ataupun secara tidak sengaja meninggalkan barang tersebut, maka akan menimbulkan resiko dan potensi kehilangan [1].

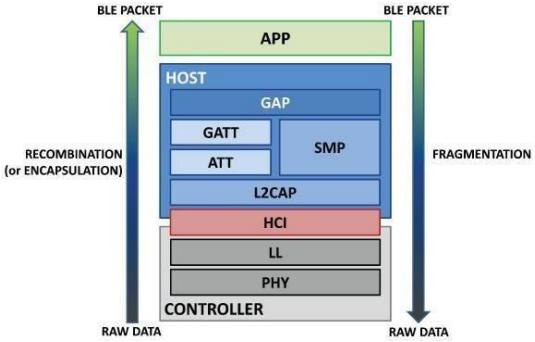
Berdasarkan permasalahan tersebut, adanya alat pengingat mungkin dapat menjadi solusi untuk meminimalisir resiko yang mungkin terjadi. Perangkat reminder dapat dipasangkan pada barang penting yang sering dibawa oleh si pemilik. Perangkat *device reminder* terdiri dari dua bagian utama yaitu modul pemancar (*transmitter*) dan modul penerima (*receiver*). Sisi pemancar perangkat *reminder* ini adalah smartphone Android pemilik barang yang akan mengaktifkan alarm saat barang tertinggal, sedangkan pada sisi penerima adalah *Bluetooth Low Energy* AT-09 yang terhubung dengan Arduino Nano dan akan dipasangkan pada barang bawaan. Smartphone pemilik barang akan di *install* aplikasi yang dibuat sebagai *reminder* untuk mengaktifkan alarm. Fitur bluetooth pada *smartphone* pemilik barang dihubungkan dengan BLE AT-09, dan saat jarak antara *smartphone* dan BLE AT-09 lebih dari 5 meter maka akan mengaktifkan alarm pada *smartphone* sebagai pengingat bahwa ada barang yang tertinggal, sehingga pemilik barang dapat segera mengambil barang tersebut guna meminimalisir resiko yang mungkin terjadi.

II. KAJIAN PUSTAKA

Internet of Things (IoT) merupakan perangkat fisik yang saling terhubung melalui suatu jaringan sehingga dapat berkomunikasi satu sama lain. Hal ini memungkinkan antar perangkat untuk mengumpulkan dan bertukar data baik pada perangkat itu sendiri maupun perangkat lain di sekitarnya. Perangkat cerdas IoT dapat terhubung dengan koneksi kabel atau nirkabel. Dengan kata lain, IoT adalah sebuah konsep komputasi dari *software* dan *hardware* yang terhubung ke jaringan dan kemudian memunculkan informasi yang berguna [2]. Objek IoT dapat diterapkan pada jaringan *Bluetooth Low Energy* (IEEE 802.15.4), Wi-Fi (IEEE 802.11), Ethernet (IEEE 802.3), atau standar komunikasi lainnya.

Objek IoT juga dapat diterapkan pada jaringan yang lebih bersifat pribadi seperti jaringan area personal yang sering disebut dengan *personal area network* (PAN). Penerapan jaringan PAN memiliki karakteristik berupa daya jangkau yang sangat terbatas dimana hanya meliputi jaringan yang ada disekitar pengguna dengan menggunakan berbagai macam perangkat yang dikonfigurasi secara pribadi. Saat seseorang menggunakan suatu perangkat yang terhubung dengan perangkat eletronik lainnya seperti laptop ataupun ponsel maka hal tersebut sudah dapat dikategorikan sebagai *Personal Area Network* [3]. Teknologi dan protokol yang umum diterapkan dalam jaringan PAN diantaranya adalah WiFi, *Wireless Application Protocol* (WAP), Bluetooth, ataupun Infrared. Salah satu teknologi yang sering digunakan dalam *Personal Area Network* adalah teknologi bluetooth.

Bluetooth adalah suatu teknologi komunikasi wireless yang memanfaatkan frekuensi radio ISM 2.4 GHz yang memungkinkan dua perangkat yang kompatibel untuk berkomunikasi dalam jarak dekat untuk membentuk suatu jaringan personal (PAN) dengan tingkat keamanan yang tinggi [4]. Bluetooth mampu menyediakan layanan komunikasi data secara *real time* antara *host to host* bluetooth dengan jarak jangkauan layanan yang terbatas. Pihak bluetooth mengembangkan sebuah protokol yang memiliki kemampuan untuk meminimalkan konsumsi daya yang disebut dengan *Bluetooth Low Energy* (BLE). BLE adalah protokol komunikasi yang dirancang dengan protokol terbaru Bluetooth 4.0 yang bekerja secara asinkron dimana komunikasi standar BLE memungkinkan pesan-pesan kecil dapat dikirim secara otomatis (tanpa diminta oleh penerima) dengan kecepatan *refresh* yang dapat diatur. BLE dirancang khusus untuk bekerja dengan sumber daya yang kecil dengan memiliki 40 *channel* yang ukuran masing-masing *channel* adalah 2 MHz. Sebanyak 37 *channel* dikenal sebagai *data channel* yang digunakan untuk koneksi saat mengirimkan data, sementara 3 *channel* lainnya disebut sebagai *advertising channel* yang digunakan untuk menyiarkan informasi dan juga untuk membuat sebuah koneksi dengan perangkat lain [5]. Protokol BLE terstruktur dalam 3 lapisan utama yaitu *App*, *Host*, dan *Controller* seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Struktur Protokol BLE

Perangkat *Bluetooth Low Energy* memiliki 2 versi pengembangan yaitu BLE Modul Serial dan BLE Beacon. Dalam penelitian ini, perangkat yang digunakan adalah salah satu produk BLE Modul Serial yaitu Modul Bluetooth 4.0 AT-09. AT-09 adalah salah satu perangkat BLE Modul Serial yang berisi chip BLE versi CC2540/CC2541. Modul ini memungkinkan untuk melakukan komunikasi serial dengan chip BLE berkat pin Rx dan pin Tx. Modul Bluetooth 4.0 AT-09 yang berukuran kecil ini memungkinkan perangkat untuk berkomunikasi dengan iOS, Android, Arduino, dan lainlain. AT-09 ini menggunakan *board* seri JDY-09 dan dapat diatur sebagai *peripheral* (*slave*) atau sebagai *central* (*master*).

Perangkat BLE AT-09 dapat berkomunikasi dengan *smartphone* yang memiliki sistem operasi iOS ataupun Android. Fitur bluetooth yang dimiliki oleh *smartphone* dapat menjadi media untuk komunikasi antara perangkat BLE AT-09 dengan perangkat *smartphone*. Selama proses komunikasi, kedua perangkat saling memancarkan dan menerima sinyal RF dari masing-masing perangkat. Nilai kuat sinyal yang diterima dapat dijadikan indikator untuk menentukan kualitas sinyal saat proses komunikasi seperti yang ditunjukkan pada Tabel 1.

Tabel 1. Indikasi Kuat sinyal

RSSI	Kategori Kualitas Sinyal
> -70 dBm	*Excellent*
-70 dBm to -85 dBm	*Good*
-86 dBm to -100 dBm	*Fair*
< -100 dBm	*Poor*
< -110 dBm	*No Signal*

Nilai kuat sinyal yang diterima sering disebut dengan *Received Signal Strength Indicator* (RSSI). RSSI adalah ukuran kekuatan sinyal saat mencapai perangkat penerima yang nilainya tergantung pada jarak dan kekuatan *broadcast*. RSSI adalah indikator relatif yang nilainya berfluktuasi/tidak tetap, tetapi jika nilai RSSI lebih besar, maka dapat dikatakan bahwa sinyal yang diterima semakin kuat [5]. Namun di sisi lain, nilai kuat sinyal yang diterima sangatlah rentan terhadap *noise*, *multi-path fading*, dan gangguan lainnya [6].

Semakin jauh perangkat dari suar, maka nilai RSSI yang didapat semakin tidak stabil. Nilai RSSI dapat dikonversi menjadi suatu nilai untuk memperkirakan jarak antara perangkat yang saling terhubung dengan menggunakan nilai *measured power* dan rumus yang didefinisikan oleh standar BLE Serial dan iBeacon. Rumus dalam perhitungan konversi jarak berdasarkan nilai RSSI ditunjukkan pada Persamaan (1).

Distance = 10 $(\frac{Measured\ Power - RSSI}{10\ \hat{}\ N})$ .....**(1)**

Measured Power : standar BLE & iBeacon untuk nilai daya ukur yaitu 69

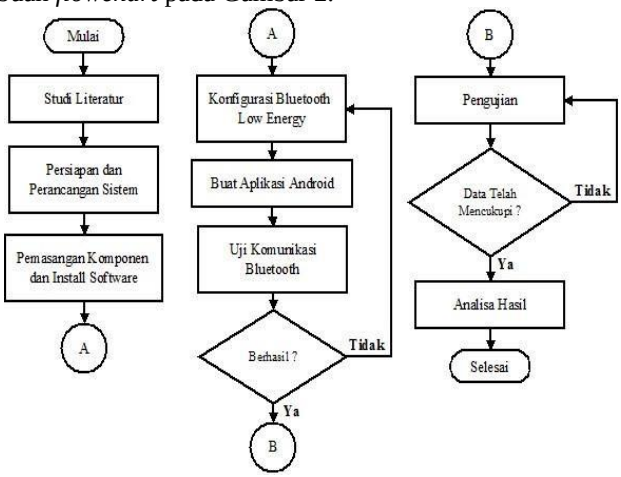
RSSI : nilai kuat sinyal yang diperoleh (contoh -64 dBm)

N : nilai konstanta antara 2 dan 4

Dalam komunikasi bluetooth, kekuatan sinyal yang diterima sangat mempengaruhi kinerja dan performa (QoS) dari perangkat. Dari hal tersebut, maka dapat dilakukan sebuah analisis terhadap performa (QoS) bluetooth berdasarkan faktor-faktor yang mungkin dapat mempengaruhi performa perangkat. Adapun beberapa faktor yang dimaksud antara lain faktor jarak antar perangkat, nilai kuat sinyal yg diterima, faktor lokasi, faktor interferensi dengan perangkat nirkabel lain, ataupun faktor adanya media penghalang saat komunikasi bluetooth berlangsung.

III. METODE PENELITIAN

Penelitian ini dilakukan dengan beberapa tahap yaitu: analisis kebutuhan, perancangan topologi, perancangan sistem, pengujian dan pengambilan data, dan analisis. Adapun alur penelitian yang dilakukan, ditunjukkan dengan sebuah *flowchart* pada Gambar 2.



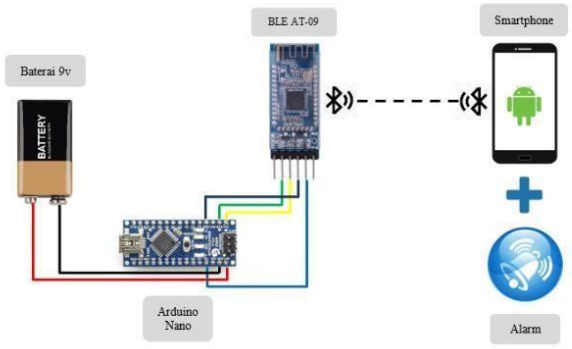
Gambar 2. Bagan Alir Metode Penelitian

A. Analisis Kebutuhan

Adapun kebutuhan pada penelitian ini meliputi beberapa perangkat keras (*hardware*) dan perangkat lunak (*software*). Kebutuhan *hardware* antara lain 1 buah laptop, *smartphone*, Arduino Nano v3, Modul Bluetooth 4.0 AT-09, 4 buah kabel jumper *female to female*, 1 kabel USB Serial, baterai 9 volt, dan konektor. Sedangkan untuk kebutuhan *software* meliputi Android Studio v2.3.3, Arduino IDE v1.8.5, Sublime Text 3, dan Docklight v1.6.

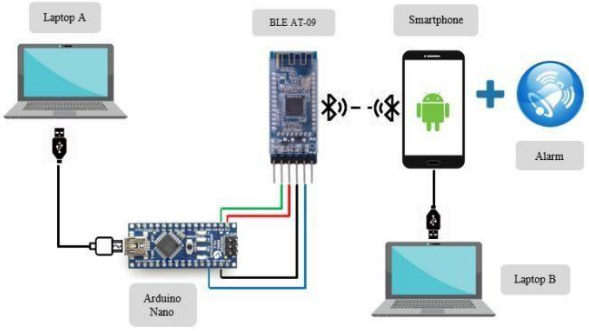
B. Perancangan Topologi

Pada penelitian ini topologi dibagi menjadi 2 tahap perancangan yaitu perancangan topologi untuk penerapan perangkat dan perancangan topologi untuk pengujian perangkat. Topologi penerapan perangkat ditujukan untuk penggunaan perangkat secara *real* seperti yang ditunjukkan pada Gambar 3.



Gambar 3. Topologi Penerapan Perangkat

Sedangkan untuk topologi pengujian perangkat ditujukan untuk proses pengambilan data dan pengujian guna melakukan analisis performa (QoS) bluetooth yang digunakan dalam penelitian ini. Topologi pengujian ditunjukkan pada Gambar 4.



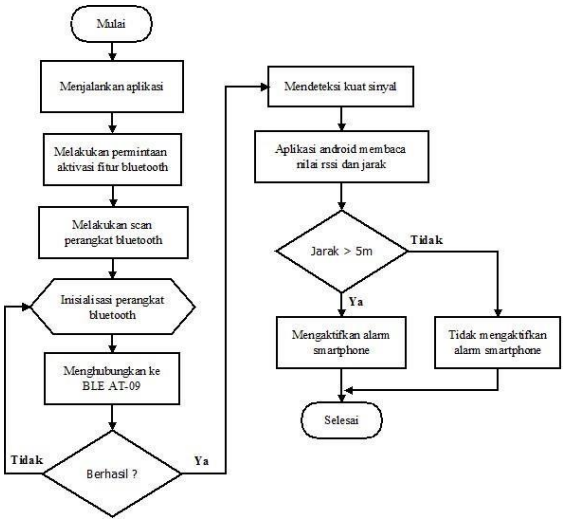
Gambar 4. Topologi Pengujian Perangkat

C. Perancangan Sistem

Sistem yang telah dirancang memiliki proses kerja sebagai berikut :

1. Saat menjalankan aplikasi *device reminder*, sistem akan secara otomatis melakukan request terhadap layanan bluetooth pada smartphone.
2. Sistem pada aplikasi akan melakukan scanning terhadap perangkat bluetooth yang ada di sekitar.
3. Menginisialisasi perangkat bluetooth yang terdeteksi.
4. Menghubungkan *smartphone* dengan perangkat Bluetooth Low Energy AT-09 untuk memulai komunikasi.
5. Sistem akan mendeteksi kuat sinyal yang di dapat dari komunikasi antar kedua perangkat tersebut kemudian menampilkan nilai rssi dan jarak melalui tampilan aplikasi yang kemudian sistem akan mengaktifkan fitur alarm saat perangkat BLE AT09 dan smartphone berjarak lebih dari 5 meter.

Diagram alir untuk menunjukkan proses kerja sistem *device reminder* dapat dilihat pada Gambar 5.



Gambar 5. Diagram Alir Sistem

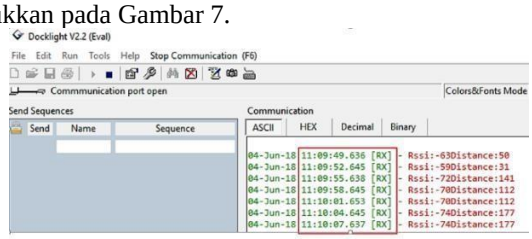
IV. PENGUJIAN DAN ANALISIS

Pada proses pengujian sistem ini, nilai kuat sinyal yang diterima dapat dilihat pada *logcat* aplikasi Android Studio melalui metode *debugging* perangkat *smartphone* dengan laptop via port USB. Pembacaan secara *real time* untuk nilai kuat sinyal yang diterima oleh *smartphone* android dapat dilihat pada Gambar 6.

```
D/BluetoothGatt: readRssi() - device: 98:7B:F3:59:20:1C
D/BluetoothGatt: onReadRemoteRssi() - Device=98:7B:F3:59:20:1C rssi=-76
D/BluetoothGatt: readRssi() - device: 98:7B:F3:59:20:1C
D/BluetoothGatt: onReadRemoteRssi() - Device=98:7B:F3:59:20:1C rssi=-79
D/BluetoothGatt: readRssi() - device: 98:7B:F3:59:20:1C
D/BluetoothGatt: onReadRemoteRssi() - Device=98:7B:F3:59:20:1C rssi=-80
D/BluetoothGatt: readRssi() - device: 98:7B:F3:59:20:1C
D/BluetoothGatt: onReadRemoteRssi() - Device=98:7B:F3:59:20:1C rssi=-77
D/BluetoothGatt: readRssi() - device: 98:7B:F3:59:20:1C
D/BluetoothGatt: onReadRemoteRssi() - Device=98:7B:F3:59:20:1C rssi=-81
```

Gambar 6. Pembacaan Kuat Sinyal

Kemudian nilai kuat sinyal yang diterima akan dikonversi menjadi nilai jarak dan dikemas oleh aplikasi *device reminder* menjadi paket yang akan dikirimkan ke perangkat BLE melalui komunikasi serial. Adapun hasil konversi dapat dilihat pada halaman aplikasi Docklight seperti yang ditunjukkan pada Gambar 7.



Gambar 7. Hasil Konversi

Pada penelitian ini pengujian dilakukan dengan menjalankan 4 skenario pengujian. Skenario pengujian yang dilakukan meliputi penempatan perangkat pada lokasi *indoor* dan *outdoor* dalam kondisi kedua perangkat tidak terhalang (*Line of Sight*) ataupun terhalang suatu media (*Non Line of Sight*).

A. *Received Signal Strength Indicator* (RSSI)

a) Pengujian *Indoor* Kondisi Tidak Terhalang Pengujian pada skenario ini dilakukan di dalam satu ruangan dengan panjang ± 11 meter dan lebar ±10 meter. Pengukuran jarak pada pengujian ini dilakukan di bagian ruangan yang dapat ditarik lurus tanpa terhalang benda-benda yang ada di ruangan tersebut. Pengujian dilakukan mulai dari jarak 1 meter hingga 10 meter.

Hasil pengujian skenario ini menunjukkan nilai RSSI yang bergerak stabil pada tiap jarak pengujian. Nilai kuat sinyal yang diterima pada skenario indoor dengan kondisi tidak terhalang (*line of sight*) ditunjukkan pada Tabel 2.

Tabel 2. Nilai RSSI Pengujian *Indoor* Kondisi Tidak Terhalang

Jarak	RSSI	Kategori Kualitas Sinyal
1 meter	-67 dBm s/d -70 dBm	Sangat Bagus
2 meter	-72 dBm s/d -76 dBm	Bagus
3 meter	-75 dBm s/d -79 dBm	Bagus
4 meter	-76 dBm s/d -80 dBm	Bagus
5 meter	-76 dBm s/d -81 dBm	Bagus
6 meter	-78 dBm s/d -80 dBm	Bagus
7 meter	-80 dBm s/d -83 dBm	Bagus
8 meter	-85 dBm s/d -87 dBm	Sedang
9 meter	-87 dBm s/d -89 dBm	Sedang
10 meter	-88 dBm s/d -90 dBm	Sedang

Tabel 2 menunjukkan hasil nilai RSSI yang bergerak stabil karena tidak adanya perubahan nilai yang semakin kecil secara signifikan pada tiap jarak pengujian. Stabilnya perubahan nilai RSSI ditandai dengan besar nilai yang semakin kecil yang hanya memiliki selisih -5 dBm antara pembacaan nilai kuat sinyal tertinggi dengan nilai kuat sinyal terendah pada jarak pengujian.

b) Pengujian *Indoor* Kondisi Terhalang Pengujian pada skenario ini dilakukan di dalam dua ruangan yang bersebelahan. Pemisah dari kedua ruangan tersebut adalah sebuah dinding dengan material berupa batu bata. Pengujian dilakukan dengan menempatkan secara terpisah perangkat *smartphone* android dan perangkat BLE pada tiap ruangan. Pengujian kuat sinyal dan performa perangkat dimulai dari jarak 1 meter hingga 10 meter dengan cara pengukuran ditarik lurus sejajar antara kedua perangkat melewati penghalang sebuah dinding. Saat proses pengujian jarak 11 meter komunikasi antara kedua perangkat sering terputus atau *disconnect* sehingga dapat diketahui jarak efektif yang dapat dijangkau perangkat pada pengujian skenario ini hanya 10 meter. Nilai kuat sinyal yang diterima pada skenario *indoor* dengan kondisi terhalang (*Non Line of Sight*) dapat dilihat pada Tabel 3.

Tabel 3. Nilai RSSI Pengujian Indoor Kondisi Terhalang

Jarak	RSSI	Kategori Kualitas Sinyal
1 meter	-68 dBm s/d -72 dBm	Bagus
2 meter	-74 dBm s/d -78 dBm	Bagus
3 meter	-76 dBm s/d -80 dBm	Bagus
4 meter	-78 dBm s/d -82 dBm	Bagus
5 meter	-78 dBm s/d -83 dBm	Bagus
6 meter	-80 dBm s/d -84 dBm	Bagus
7 meter	-79 dBm s/d -85 dBm	Bagus
8 meter	-83 dBm s/d -89 dBm	Sedang
9 meter	-86 dBm s/d -90 dBm	Sedang
10 meter	-88 dBm s/d -92 dBm	Sedang
11 meter	*disconnect*	Jelek/ *no signal*

Hasil pengujian skenario ini menunjukkan nilai yang lebih kecil dan kurang stabil dibandingkan dengan nilai

RSSI pada skenario dengan kondisi tidak terhalang. Kurang stabilnya perubahan nilai RSSI dapat diketahui dari besar nilai yang semakin kecil secara signifikan hingga mencapai selisih -6 dBm antara pembacaan nilai kuat sinyal tertinggi dengan nilai kuat sinyal terendah jarak pengujian. Hal tersebut mungkin saja dikarenakan pancaran sinyal RF bluetooth yang kurang maksimal karena terhalang oleh dinding.

c) Pengujian *Outdoor* Kondisi Tidak Terhalang Pengujian pada skenario ini dilakukan di luar ruangan pada sebuah lorong terbuka. Kondisi lorong yang dipilih yaitu lorong kosong yang tidak ada benda seperti kursi, meja, ataupun benda lainnya. Namun, pengujian pada skenario ini memiliki kondisi yang sedikit berbeda dengan kondisi pada pengujian skenario lain, yang mana di lorong ini terdapat perangkat nirkabel lain berupa perangkat *access point*. Pada skenario ini dilakukan pengujian mulai dari jarak 1 meter hingga 13 meter, namun daya jangkau efektif dari perangkat bluetooth pada skenario ini hanya 12 meter. Pada jarak 13 meter perangkat BLE masih terdeteksi dan dapat terhubung dengan aplikasi, namun sering kali terputus atau *disconnect*. Dan pada jarak 14 meter perangkat BLE sudah tidak terdeteksi oleh aplikasi.

Nilai RSSI yang diterima pada pengujian outdoor dengan kondisi tidak terhalang ini ditunjukkan pada Tabel 4.

Tabel 4. Nilai RSSI Pengujian Outdoor Kondisi Tidak Terhalang

Jarak	RSSI	Kategori Kualitas Sinyal
1 meter	-68 dBm s/d -72 dBm	Bagus
2 meter	-72 dBm s/d -76 dBm	Bagus
3 meter	-74 dBm s/d -79 dBm	Bagus
4 meter	-77 dBm s/d -83 dBm	Bagus
5 meter	-78 dBm s/d -85 dBm	Bagus
6 meter	-79 dBm s/d -83 dBm	Bagus
7 meter	-82 dBm s/d -87 dBm	Sedang
8 meter	-84 dBm s/d -90 dBm	Sedang
9 meter	-86 dBm s/d -91 dBm	Sedang
10 meter	-89 dBm s/d -93 dBm	Sedang
11 meter	-88 dBm s/d -94 dBm	Sedang
12 meter	-90 dBm s/d -94 dBm	Sedang
13 meter	*disconnect*	Jelek/ *no signal*

Nilai RSSI yang ditunjukkan pada Tabel 4 menunjukkan hasil nilai yang bergerak tidak stabil karena adanya perubahan nilai RSSI yang semakin kecil secara signifikan pada beberapa jarak pengujian. Hal tersebut dapat diketahui dari perubahan nilai RSSI dengan besar nilai yang semakin kecil hingga mencapai selisih -7 dBm antara pembacaan nilai kuat sinyal tertinggi dengan nilai kuat sinyal terendah pada jarak pengujian. Tidak stabilnya kuat sinyal yang diterima mungkin saja disebabkan adanya interferensi dari perangkat nirkabel lain yang ada di sekitar tempat pengujian. Hal tersebut dapat terjadi karena mungkin saja perangkat nirkabel yang ada di sekitar tempat pengujian menggunakan perangkat yang beroperasi pada frekuensi/band yang sama dengan perangkat bluetooth yang digunakan yaitu frekuensi 2,4 GHz.

d) Pengujian *Outdoor* Kondisi Terhalang Pengujian pada skenario ini dilakukan di 2 lorong terbuka yang berdekatan di gedung yang sama dengan tempat pengujian skenario outdoor kondisi tidak terhalang. Hanya saja, kondisi pada tempat ini berbeda dengan skenario sebelumnya yang mana di kedua lorong ini tidak terdapat perangkat nirkabel lain. Pengujian dilakukan pada jarak 1 meter hingga 10 meter dengan kondisi terhalang sebuah dinding. Nilai kuat sinyal yang diterima pada pengujian skenario *outdoor* dengan kondisi terhalang (*Non Line of Sight*) dapat dilihat pada Tabel 5.

Tabel 5. Nilai RSSI Pengujian Outdoor Kondisi Terhalang

Jarak	RSSI	Kategori Kualitas Sinyal
1 meter	-69 dBm s/d -72 dBm	Bagus
2 meter	-71 dBm s/d -75 dBm	Bagus
3 meter	-75 dBm s/d -80 dBm	Bagus
4 meter	-78 dBm s/d -83 dBm	Bagus
5 meter	-84 dBm s/d -87 dBm	Sedang
6 meter	-83 dBm s/d -89 dBm	Sedang
7 meter	-85 dBm s/d -91 dBm	Sedang
8 meter	-88 dBm s/d -93 dBm	Sedang
9 meter	-89 dBm s/d -94 dBm	Sedang
10 meter	-89 dBm s/d -97 dBm	Sedang
11 meter	*disconnect*	Jelek/ *no signal*

Nilai RSSI yang ditunjukkan pada Tabel 5 menunjukkan hasil nilai yang bergerak kurang stabil karena adanya perubahan nilai RSSI yang semakin kecil secara signifikan pada beberapa jarak pengujian. Hal tersebut dapat diketahui dengan kurang stabilnya perubahan nilai RSSI dengan besar nilai yang semakin kecil hingga selisih -8 dBm antara pembacaan nilai kuat sinyal tertinggi dengan nilai kuat sinyal terendah pada setiap jarak pengujian. Dari hasil pengujian tersebut menunjukkan bahwa media penghalang dan faktor lokasi penggunaan alat sangat mempengaruhi kuat sinyal yang dapat diterima oleh perangkat.

B. *Packet Loss*

a) Pengujian *Indoor* Kondisi Tidak Terhalang Hasil pengujian QoS pada skenario ini menunjukkan uji kualitas parameter *packet loss* yang dapat dikategorikan sangat bagus karena tidak adanya paket yang hilang dari 90 paket yang dikirimkan pada setiap jarak pengujian. Berdasarkan hasil tersebut, maka penghitungan *packet loss ratio* tidak perlu dilakukan karena parameter *packet loss* dapat diketahui bernilai 0% pada tiap jarak pengujian seperti yang ditunjukkan pada Tabel 6.

Tabel 6. Parameter *Packet Loss* Pengujian *Indoor* Kondisi Tidak Terhalang

Jarak	Paket Terkirim	Paket Diterima	Packet Loss	Kategori Kualitas Uji
1 meter	90	90	0%	Sangat Bagus
2 meter	90	90	0%	Sangat Bagus
3 meter	90	90	0%	Sangat Bagus
4 meter	90	90	0%	Sangat Bagus
5 meter	90	90	0%	Sangat Bagus
6 meter	90	90	0%	Sangat Bagus
7 meter	90	90	0%	Sangat Bagus

8 meter	90	90	0%	Sangat Bagus
9 meter	90	90	0%	Sangat Bagus
10 meter	90	90	0%	Sangat Bagus

b) Pengujian *Indoor* Kondisi Terhalang Hasil pengujian QoS parameter *packet loss* pada skenario ini menunjukkan adanya paket yang hilang dari 90 paket yang dikirimkan pada beberapa jarak pengujian seperti yang ditunjukkan pada Tabel 7.

Tabel 7. Parameter *Packet Loss* Pengujian *Indoor* Kondisi Terhalang

Jarak	Paket Terkirim	Paket Diterima	Packet Loss	Kategori Kualitas Uji
1 meter	90	90	0%	Sangat Bagus
2 meter	90	90	0%	Sangat Bagus
3 meter	90	89	1,1%	Bagus
4 meter	90	89	1,1%	Bagus
5 meter	90	90	0%	Sangat Bagus
6 meter	90	90	0%	Sangat Bagus
7 meter	90	90	0%	Sangat Bagus
8 meter	90	90	0%	Sangat Bagus
9 meter	90	89	1,1%	Bagus
10 meter	90	88	2,2%	Bagus

Berdasarkan hasil tersebut, uji kualitas parameter *packet loss* pada kondisi ini dapat dikategorikan bagus karena menunjukkan nilai *packet loss* kurang dari 3%. Adanya paket yang hilang pada pengujian skenario ini mungkin saja dikarenakan oleh kegagalan dalam penerimaan paket akibat dari *overflow* yang terjadi pada *buffer* perangkat BLE AT-09. Hal tersebut dapat disebabkan karena adanya media penghalang berupa dinding yang menyebabkan tidak maksimalnya sinyal yang diterima sehingga pembacaan paket pada *buffer* penerima (*buffer RX*) menjadi tidak maksimal atau terjadi *error*.

c) Pengujian *Outdoor* Kondisi Tidak Terhalang Hasil pengujian QoS parameter *packet loss* pada skenario ini menunjukkan adanya paket yang hilang dari 90 paket yang dikirimkan pada beberapa jarak pengujian seperti yang ditunjukkan pada Tabel 8.

Tabel 8. Parameter *Packet Loss* Pengujian *Outdoor* Kondisi Tidak Terhalang

Jarak	Paket Terkirim	Paket Diterima	Packet Loss	Kategori Kualitas Uji
1 meter	90	90	0%	Sangat Bagus
2 meter	90	90	0%	Sangat Bagus
3 meter	90	90	0%	Sangat Bagus
4 meter	90	90	0%	Sangat Bagus
5 meter	90	90	0%	Sangat Bagus
6 meter	90	90	0%	Sangat Bagus
7 meter	90	90	0%	Sangat Bagus
8 meter	90	90	0%	Sangat Bagus
9 meter	90	90	0%	Sangat Bagus
10 meter	90	89	1.1%	Bagus
11 meter	90	90	0%	Sangat Bagus
12 meter	90	88	2,2%	Bagus

Berdasarkan hasil tersebut, uji kualitas parameter *packet loss* pada kondisi ini masih dapat dikategorikan bagus karena menunjukkan nilai *packet loss* kurang dari 3%. Adanya paket yang hilang pada pengujian skenario ini mungkin saja dikarenakan oleh kegagalan dalam penerimaan paket akibat adanya gangguan atau interferensi sinyal dari perangkat nirkabel di sekitar yang menyebabkan terganggunya komunikasi bluetooth. Sehingga selama proses komunikasi, paket yang dikirimkan menumpuk pada *buffer* penerima dan terjadi *error* pada proses pembacaan paket oleh *buffer* penerima. Selain itu, *packet loss* terjadi pada jarak pengujian yang semakin jauh yaitu 10 meter dan 12 meter, hal tersebut membuktikan bahwa jarak sangat mempengaruhi performa dalam komunikasi perangkat bluetooth.

d) Pengujian *Outdoor* Kondisi Terhalang Hasil pengujian QoS parameter *packet loss* pada skenario ini menunjukkan adanya paket yang hilang dari 90 paket yang dikirimkan pada beberapa jarak pengujian seperti yang ditunjukkan pada Tabel 9.

Tabel 9. Parameter *Packet Loss* Pengujian *Outdoor* Kondisi Terhalang

Jarak	Paket Terkirim	Paket Diterima	Packet Loss	Kategori Kualitas Uji
1 meter	90	90	0%	Sangat Bagus
2 meter	90	90	0%	Sangat Bagus
3 meter	90	89	1,1%	Bagus
4 meter	90	90	0%	Sangat Bagus
5 meter	90	88	2,2%	Bagus
6 meter	90	90	0%	Sangat Bagus
7 meter	90	90	0%	Sangat Bagus
8 meter	90	88	2,2%	Bagus
9 meter	90	89	1,1%	Bagus
10 meter	90	86	4,4%	Sedang

Berdasarkan hasil tersebut, uji kualitas parameter *packet loss* pada kondisi ini dapat dikategorikan sedang karena menunjukkan adanya nilai *packet loss* lebih dari 4%. Adanya paket yang hilang pada pengujian skenario ini mungkin saja dikarenakan oleh kegagalan dalam penerimaan paket akibat dari *overflow* yang terjadi pada *buffer* perangkat BLE AT09. Hal tersebut dapat disebabkan oleh adanya media penghalang berupa dinding ataupun gangguang lain pada lingkungan *outdoor*. Hal tersebut yang menyebabkan tidak maksimalnya sinyal yang diterima sehingga pembacaan paket pada *buffer* penerima (*buffer RX*) menjadi tidak maksimal atau terjadi *error*.

C. *Delay*

a) Pengujian *Indoor*

Pengujian parameter *delay* pada skenario *indoor* ini menunjukkan waktu *delay* yang cukup besar jika dibandingkan dengan standarisasi nilai *delay* seperti yang ditunjukkan pada Tabel 10.

Tabel 10. Parameter *Delay* Pengujian *Indoor* Kondisi Tidak Terhalang dan Kondisi Terhalang

Jarak	Delay	
	Kondisi Tidak Terhalang	Kondisi Terhalang
1 meter	1,82 detik	1,93 detik
2 meter	1,93 detik	1,92 detik
3 meter	1,93 detik	2,09 detik
4 meter	1,92 detik	2,07 detik
5 meter	2,08 detik	2,14 detik
6 meter	1,92 detik	1,93 detik
7 meter	1,95 detik	2,25 detik
8 meter	1,93 detik	2,09 detik
9 meter	2,25 detik	2,42 detik
10 meter	2,38 detik	2,83 detik

Besarnya waktu *delay* yang diperoleh dimungkinkan karena rendahnya kecepatan transfer data (*RF data rate*) dari perangkat BLE AT-09 yang hanya sebesar 6 kbps. Waktu *delay* pengiriman dan penerimaan paket antara kedua perangkat pada kondisi terhalang menunjukkan nilai *delay* yang lebih besar dibandingkan nilai *delay* pada skenario dengan kondisi tidak terhalang. Hal ini menunjukkan bahwa objek penghalang sangatlah mempengaruhi performa dan kinerja dari perangkat bluetooth. Selain itu, faktor penghalang dengan jarak yang semakin jauh antara kedua perangkat menimbulkan anomali atau gangguan sehingga menjadikan semakin besarnya nilai *delay* yang hampir mencapai 0,5 detik.

b) Pengujian *Outdoor*

Pengujian parameter *delay* pada skenario ini menunjukkan waktu *delay* yang lebih besar dari pengujian di lokasi *indoor* seperti yang ditunjukkan pada Tabel 11.

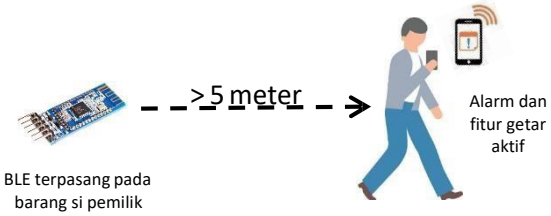
Tabel 11. Parameter *Delay* Pengujian *Outdoor* Kondisi Tidak Terhalang dan Kondisi Terhalang

Jarak	Delay	
	Kondisi Tidak Terhalang	Kondisi Terhalang
1 meter	2,14 detik	2,08 detik
2 meter	2,25 detik	2,28 detik
3 meter	2,28 detik	2,36 detik
4 meter	2,17 detik	2,32 detik
5 meter	2,57 detik	2,25 detik
6 meter	2,23 detik	2,52 detik
7 meter	2,46 detik	2,52 detik
8 meter	2,61 detik	2,81 detik
9 meter	2,38 detik	2,88 detik
10 meter	2,85 detik	3.06 detik
11 meter	2,90 detik	-
12 meter	2,96 detik	-

Pada hasil pengujian yang ditunjukkan pada Tabel 11 menunjukkan nilai waktu *delay* yang mengalami perubahan yang signifikan di beberapa jarak pengujian. Perubahan waktu *delay* secara signifikan terjadi pada kondisi tidak terhalang di pengujian jarak 5 meter hingga 10 meter. Hal tersebut mungkin terjadi karena adanya interferensi pada saat proses pengujian di jarak 5 hingga 10 meter oleh perangkat nirkabel lain yang berada di tempat pengujian. Interferensi pada komunikasi bluetooth akan mempengaruhi performa perangkat bluetooth berupa penurunan kecepatan transfer data sehingga dapat memungkinkan terjadinya waktu *delay* yang semakin besar. Dengan terjadinya penurunan kecepatan transfer data maka akan semakin lama waktu yang dibutuhkan dalam jeda proses antara pengiriman dengan penerimaan paket.

Sedangkan untuk pengujian saat kondisi terhalang menunjukkan nilai waktu *delay* yang lebih besar dibandingkan dengan hasil pengujian pada skenario lain. Media penghalang berupa dinding dan faktor lingkungan *outdoor* sangatlah mempengaruhi kinerja dan performa dari perangkat bluetooth itu sendiri. Selain itu media penghalang juga dapat menimbulkan gangguan pada komunikasi bluetooth. Terlebih pada pengujian jarak yang semakin jauh menunjukkan waktu *delay* yang semakin besar secara signifikan mencapai *delay* sebesar 3 detik. Semakin jauh jarak antara kedua perangkat, adanya media penghalang, dan kemungkinan adanya gangguan lain di lingkungan *outdoor* menjadikan kurang maksimalnya penerimaan sinyal bluetooth sehingga memungkinkan terjadinya anomali dalam komunikasi perangkat bluetooth yang menyebabkan penurunan kualitas performa perangkat.

D. Pengujian Alarm Pada Sistem Device Reminder Pada sistem *device reminder* ini, alarm dan fitur getar (*vibrate*) akan diaktifkan saat perangkat *smartphone* bergerak menjauhi perangkat BLE AT-09 dengan jarak lebih dari 5 meter. Seperti yang diilustrasikan pada Gambar 8.



Gambar 8. Ilustrasi Penerapan *Device Reminder*

Alarm dan fitur getar akan aktif secara terus menerus sampai si pemilik barang menyadari bahwa ada barang bawaannya yang tertinggal. Selain mengaktifkan alarm dan fitur getar, perangkat *smartphone* juga menampilkan suatu tampilan peringatan. Melalui tampilan peringatan ini si pemilik barang dapat mematikan bunyi alarm dan fitur getar melalui sebuah tombol. Tampilan peringatan pada sistem *device reminder* ini ditunjukkan pada Gambar 9.

Gambar 9. Tampilan Peringatan

Pengujian ini dilakukan untuk mengetahui akurasi dari aktivasi alarm dengan membandingkan antara aktivasi alarm berdasarkan jarak pada teori dan aktivasi alarm pada jarak sebenarnya. Berdasarkan teori, alarm akan diaktifkan pada jarak lebih dari 5 meter. Hal tersebut dibandingkan dengan aktivasi alarm pada jarak sebenarnya dalam keadaan perangkat smartphone bergerak menjauhi modul BLE AT-09.

Untuk pengujian ini dilakukan 10 kali percobaan untuk mengetahui alarm akan aktif pada jarak berapa meter berdasarkan pengukuran jarak sebenarnya. Berdasarkan dari 10 kali percobaan tersebut diperoleh hasil alarm aktif pada jarak 4,5 meter – 5,4 meter. Hasil yang diperoleh adalah sebagai berikut :

- Alarm aktif di jarak 4,5 meter pada 1x percobaan
- Alarm aktif di jarak 4,6 meter pada 1x percobaan
- Alarm aktif di jarak 4,8 meter pada 2x percobaan
- Alarm aktif di jarak 5 meter pada 4x percobaan
- Alarm aktif di jarak 5,2 meter pada 1x percobaan
- Alarm aktif di jarak 5,4 meter pada 1x percobaan

Berdasarkan dari hasil tersebut dapat diketahui aktivasi alarm tidak selalu berada di jarak lebih dari 5 meter. Hal tersebut memang mungkin dapat terjadi mengingat pembacaan jarak pada aplikasi sesuai dengan nilai kuat sinyal yang diterima (RSSI) yang bersifat fluktuatif. Namun, dari 10 kali percobaan tersebut didapatkan sebanyak 6x percobaan yang menunjukkan aktivasi alarm pada jarak lebih dari 5 meter. Jadi dapat disimpulkan akurasi aktivasi alarm pada sistem ini cukup akurat sesuai dengan teori pada penelitian ini.

V. KESIMPULAN

Penggunaan perangkat Bluetooth Low Energy Modul Serial AT-09 tidak begitu cocok dalam penerapan komunikasi data cepat pada sistem *device reminder* ini karena memiliki kecepatan transfer data yang rendah sehingga menimbulkan waktu delay cukup besar yang mencapai ± 2 detik.

Hasil pengujian dari skenario dengan media penghalang menunjukkan bahwa faktor media penghalang sangatlah mempengaruhi performa perangkat bluetooth yang memungkinkan resiko kehilangan paket dalam komunikasi bluetooth. Hal tersebut dibuktikan pada skenario pengujian dengan kondisi terhalang yang menunjukkan cukup banyaknya paket yang hilang pada beberapa jarak pengujian dengan nilai *packet loss* sebesar 1,1% sampai dengan 4,4%. Hal ini berkaitan dengan menurunnya sinyal RF yang diterima karena faktor jarak yang semakin jauh dan adanya media penghalang antara perangkat pengirim dan perangkat penerima.

Nilai kuat sinyal yang diterima (RSSI) pada kondisi terhalang (*Non Line of Sight*) lebih tidak stabil dibandingkan dengan nilai kuat sinyal yang diterima pada kondisi tidak terhalang (*Line of Sight*). Hal ini dibuktikan oleh terjadinya penurunan nilai yang signifikan mencapai selisih -8 dBm saat kondisi terhalang. Faktor jarak juga sangat mempengaruhi besar kecilnya waktu *delay*, dengan semakin jauhnya jarak antara perangkat pengirim dengan perangkat penerima maka waktu *delay* dalam komunikasi perangkat bluetooth akan semakin besar.

DAFTAR PUSTAKA

[1] Yudiansyah, "Perancangan Dan Realisasi Wireless Device Reminder Multi User Menggunakan Teknik Modulasi Digital Pada Modul XBee," Universitas Telkom, Bandung, 2015.

[2] K. J. Singh and D. S. Kapoor, "Create Your Own Internet of Things: A survey of IoT platforms," *IEEE Consumer Electronics Magazine*, pp. 57-68, 2017.

[3] R. Rashid and R. Yusoff, "Bluetooth Performance Analysis in Personal Area Network (PAN)," *2006 International RF and Microwave Conference*, pp. 393 - 397, 2006.

[4] A. Nurcahyana, I. Wijayanto and J. Andjarwirawan, "Development of Mobile Indoor Positioning System Application Using Android and Bluetooth Low Energy with Trilateration Method," *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT)*, pp. 185-189, 2017.

[5] B. Soewito, Y. Agses and G. Fergyanto, "Increasing Accuracy of Bluetooth Low Energy for Distance Measurement Applications," *11th Information and Creativity Support Systems (KICSS)*, pp. 1-5, 2016.

[6] M. Botta and M. Simek, "Adaptive Distance Estimation Based on RSSI in 802.15.4 Network," *Radio Engineering*, vol. 22, pp. 11631168, 2013.

[7] S. Bertuletti, A. Cerreati, U. Della, M. Caldara and M. Galizzi, "Indoor Distance Estimated from Bluetooth Low Energy Signal Strength: Comparison of Regression Models," *2016 IEEE Sensors Applications Symposium (SAS)*, pp. 1-5, 2016.

[8] L. Grezelda, "Studi Penerapan Teknologi Bluetooth untuk Monitoring Sensor pada Kendaraan Roda Empat," Universitas Gadjah Mada, Yogyakarta, 2014.

[9] J. I Wayan, "Analisis Parameter QoS Terhadap Pengaruh Pertambahan Jarak dan Interferensi Wi-Fi Pada Jaringan Bluetooth," Universitas Jember, Jember, 2015.

[10] S. Onofre, P. Miguel, J. Paulo and P. Sousa, "Surpassing Bluetooth Low Energy Limitations on Distance Determination," *2016 IEEE International Power Electronics and Motion Control Conference (PEMC)*, pp. 843-847, 2016.

[11] H. Hoshi, H. Ishizuka, A. Kobayashi and A. Minamikawa, "An Indoor Location Estimation Using BLE Beacons Considering Movable Obstructions," *2017 Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, pp. 1-2, 2017.

J. Neburka, Tlamsa, Benes, Pollak, Kaller, Bolecek, Sabesta and Kratochvil, "Study of the Performance of RSSI based Bluetooth Smart Indoor Positioning," Ko˘sice, Slovak Republic , 2016.

Implementasi dan Analisa QoS pada *Smart Door* yang Terintegrasi dengan Aplikasi Telegram

Aidah Khuriatul Mujtahidah[1], Unan Yusmaniar Oktiawati[2]
[1]Departemen Teknik Elektro dan Informatika,
Sekolah Vokasi, Universitas Gadjah Mada,
[2]Departemen Teknik Elektro dan Informatika,
Sekolah Vokasi, Universitas Gadjah Mada,
[1] aidahkhm@gmail.com, [2] unan_yusmaniar@ugm.ac.id

Abstract - Home security is a mandatory requirement for every homeowner. With home security, homeowners will always feel safe leaving the house for a long time. In this final project, the author made a smart door with its main components are the HC-SR04 ultrasonic sensor and web camera. The function of the ultrasonic sensor and webcam is to detect a movement in the area around the access of people. In this system, not only can it detect movement, but also send notifications in the form of text and videos through the Telegram application so that homeowners can find out if someone tries to enter the house even though the owner of the house is away from his house. The author also adds dual security in the form of a remote control system, where when the homeowner knows of theft, the homeowner simply sends a "close" message through the Telegram application, so the door automatically locks and keeps the thief for a while until the homeowner arrives at location. This remote control system can also be used when the homeowner forgets to lock the door so that when the owner of the house is away from home, he simply controls the door through the telegram application.
Keywords: ***smart door, ultrasonic, web camera, raspberry pi, solenoid door lock***

Intisari - ***Home security*** **adalah kebutuhan wajib setiap pemilik rumah. Dengan adanya** ***home security*****, pemilik rumah akan selalu merasa aman meninggalkan rumah dalam jangka waktu yang lama. Pada proyek akhir ini, penulis membuat sebuah** ***smart door*** **dengan komponen utamanya adalah sensor ultrasonik HC-SR04 dan** ***web camera*****. Fungsi sensor ultrasonik dan** ***webcam*** **tersebut untuk mendeteksi suatu pergerakan di area sekitar akses masuk-keluarnya orang. Pada sistem ini, tidak hanya dapat mendeteksi gerakan, tetapi juga mengirimkan notifikasi berupa teks dan video melalui aplikasi Telegram sehingga pemilik rumah dapat mengetahui jika ada orang yang mencoba masuk ke dalam rumah walaupun si pemilik rumah sedang berada jauh dari rumahnya. Penulis juga menambahkan keamanan ganda berupa sistem kendali jarak jauh, dimana ketika pemilik rumah telah mengetahui adanya pencurian, si pemilik rumah cukup mengirimkan pesan "*close*" melalui aplikasi Telegram, sehingga pintu otomatis terkunci dan membuat pencuri akan tertahan untuk beberapa saat sampai pemilik rumah tiba di lokasi. Sistem kendali jarak jauh ini juga dapat digunakan ketika si pemilik rumah lupa mengunci pintu sehingga ketika si pemilik rumah sudah berada jauh dari rumah, dia cukup mengontrol pintu melalui aplikasi telegram tersebut.**
Kata kunci: ***smart door, ultrasonic, web camera, raspberry pi, solenoid door lock***

I. PENDAHULUAN

Keamanan memiliki peranan yang sangat penting dalam mengamankan suatu aset berharga dari ancaman pihak luar. Sistem keamanan rumah yang dibangun pada umumnya hanya dapat memberikan peringatan di tempat kejadian berupa sebuah alarm dan tidak terdapat peringatan secara *online* yang terhubung dengan perangkat ponsel pintar seperti notifikasi *email* ataupun panel pemantauan jarak jauh yang dapat diakses setiap saat.

Berdasarkan data Polda Metro Jaya, di wilayah DKI Jakarta selama Mei 2018 telah terjadi 1.447 kasus kejahatan. Antara lain, pencurian dengan pemberatan sebanyak 253 kasus, pencurian dengan kekerasan 19 kasus, perampasan 27 kasus, perampokan empat kasus, pembajakan satu kasus, pencurian kendaraan sepeda motor 154 kasus, mobil 56 kasus. Kemudian pembunuhan delapan kasus, penganiayaan berat 146 kasus, pemerasan 38 kasus, pemerkosaan enam kasus, dan narkotika 547 kasus.

Permasalahan terkait kasus pencurian menjadikan suatu drongan bagi para peneliti untuk memberikan solusi dalam beragam model dan metode analisis. Pengendalian akses pintu ruangan dengan cara konvensional sangat rawan pencurian, karena kunci bisa diduplikasi, sehingga bisa saja semua orang menduplikatnya untuk suatu kejahatan. Untuk itu, perlu pembaharuan sistem, salah satunya adalah mengembangkan sistem pemantau dengan menggunakan program motion. Program *motion* menangani deteksi gerak dan *streaming*. Selain itu sistem juga dibuat agar dapat mengirimkan notifikasi saat terjadi gerakan melalui *email*. Untuk alasan keamanan dan *backup* data, sistem juga akan mengunggah video hasil rekaman ke *google drive*. Sistem terdiri dari 1 server dan 1 klien yang saling berhubungan menggunakan jaringan WLAN [1].

Penelitian ini bertujuan untuk membangun sebuah sistem keamanan rumah otomatis yang dapat mendeteksi suatu gerakan serta memberikan notifikasi berupa teks dan video ketika terdeteksi suatu gerakan, untuk mengamankan aset yang berada di dalam rumah terhadap tindakan pencurian. Serta memberikan keamanan ganda berupa kunci pintu dengan kendali jarak jauh menggunakan aplikasi telegram, agar pengguna atau pemilik rumah dapat mengakses pintu dari jarak jauh ketika pengguna atau pemilik rumah lupa mengunci pintu ataupun ada seseorang yang tidak dikenal masuk ke dalam rumah.

II. TINJAUAN PUSTAKA

Konsep *smart door* merupakan bagian dari konsep *pervasive computing* yang telah menjadi isu hangat dari berbagai penelitian di berbagai sumber seperti artikel-artikel penelitian, majalah teknologi, dan internet. *Pervasive computing* adalah suatu lingkungan dimana sejumlah teknologi (terutama teknologi komputer) digunakan dan menyatu di dalam objek dan aktivitas manusia sehari-hari, sehingga kehadirannya tidak dirasakan sebagai sesuatu yang khusus [2]. *Pervasive computing* mengacu pada konsep dimana sistem dapat diakses di mana saja, kapan saja saat dibutuhkan dan melibatkan lebih dari satu perangkat seperti laptop, *handphone*, *personal digital assistance* dan lainlain. Dalam jurnal yang berjudul "Teknologi dan Teknik Sistem Terdistribusi Pervasif dalam Bidang Logistik: Studi Literatur Sistematis" [3] ditemukan beberapa teknologi seperti RFID, *Wireless Sensor Networks*, dan teknologi gabungan yang

digunakan untuk bidang logistik. Juga ditemukan beberapa teknik untuk meningkatkan kinerja seperti metode *smoothing*, teknik algoritma *multi-treshold*, dan teknik kalman *filter*. Berdasarkan literatur dapat disimpulkan bahwa beberapa teknik dapat meningkatkan kinerja teknologi.

Pengendalian akses pintu ruangan dengan cara konvensional sangat rawan pencurian, karena kunci bisa diduplikasi, sehingga bisa saja semua orang menduplikatnya untuk suatu kejahatan. Untuk itu, perlu pembaharuan sistem, salah satunya adalah memanfaatkan teknologi NCF/RFID. Selain itu, digunakan *XBee* sebagai protokol jaringan pada akses pintu untuk mendukung beberapa jaringan *node* yang berada di pintu agar dapat dikontrol dari sebuah koordinator. Dari penelitian yang dilakukan oleh Li Xizhi, Zhou Yu, Ai Changsheng, dan Qian Lijun, dapat diketahui bahwa RFID dapat digunakan untuk membuka setiap pintu masuk dari ruangan petugas keamanan. Dengan adanya *XBee* di dalam sistem tersebut, setiap informasi yang didapat dari akses pintu masuk dapat dikirim ke komputer pusat untuk dilakukan *monitoring*. Unit kontrol akses yang menggabungkan teknologi *XBee* dan frekuensi radio seperti RFID diidentifikasi memiliki keuntungan dibandingkan dengan unit kontrol akses konvensional yaitu dalam hal kenyamanan, efektivitas, dan keamanan [4].

Pada tahun 2014, dibuat sebuah penelitian, dimana dirancang sebuah sistem pemantau ruangan berbasis *Mini* PC (Raspberry Pi). Sistem pemantauan dibuat menggunakan program *motion*. Program *motion* menangani deteksi gerak dan *streaming*. Selain itu sistem juga dibuat agar dapat mengirimkan notifikasi saat terjadi gerakan melalui *email*. Untuk alasan keamanan dan *backup* data, sistem juga akan mengunggah video hasil rekaman ke *google drive*. Sistem terdiri dari 1 server dan 1 klien yang saling berhubungan menggunakan jaringan WLAN. Penelitian yang berjudul "Sistem Pemantau Ruangan Berbasis *Video Streaming* dengan *Server Raspberry Pi*" [5] ini, juga meninjau dari beberapa penelitian, salah satunya *system monitoring* berbasis *live video streaming* yang dilengkapi dengan notifikasi melalui SMS. Ero telah melakukan penelitian mengenai sistem pemantauan rumah yang menggunakan teknologi *video streaming* dan dalam sistem tersebut dimasukkan juga program, sehingga sistem hanya akan merekam apabila terdeteksi suatu gerakan, sehingga ukuran video yang terekam tidak terlalu besar karena tidak membutuhkan durasi yang panjang. Sistem milik Ero ini juga memberikan notifikasi melalui SMS kepada pemilik ketika terdeteksi gerakan ditempat yang dipantau [6].

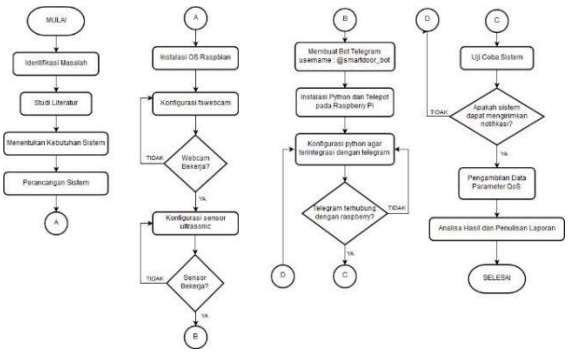
Pada tahun 2016, [7] membuat *prototype* kendali lampu rumah berbasis arduino yang terintegrasi dengan SMS *gateway*. Dimana sistem ini terdiri dari Arduino Uno, *module* GSM900, sensor LDR, *relay*, dan lampu. Prinsip kerja sistem adalah Arduino Uno digunakan untuk mengatur *relay* agar dapat menjalakan fungsi menghidupkan atau mematikan lampu dari jarak jauh.

Pada penelitian yang telah dilakukan [8] menjelaskan tentang sistem pemantau ruangan jarak jauh dengan sensor inframerah berbasis mikrokontroler AT89S51. Penelitian bertujuan untuk membuat sistem pengaman ruangan berbasis mikrokontroler AT89S51 dengan sensor PIR KC7783R sebagai *detector* inframerah yang dipancarkan ke tubuh manusia. Ketika sensor mendeteksi kehadiran seseorang memasuki ruangan, maka mikrokontroler akan mengaktifkan suara peringatan yang telah disimpan di dalam IC ISD2560 dan suara peringatan akan terdengar melalui *speaker*. Hasil dari penelitian ini berupa frekuensi suara yang dikeluarkan oleh *speaker* yang menandakan adanya seseorang yang masuk.

Kemudian pada tahun yang sama, Alexander Puliano [9] membuat suatu sistem pemantauan ruangan terintegrasi dengan *website* dan *email*, yang dapat diakses setiap saat. Perancangan ini menggunakan sensor PIR untuk membaca pergerakan manusia berupa perubahan radiasi inframerah, radiasi akan terbaca sensor hingga jarak 5 meter dengan sudut pembacaan 90° parabola. Jika terbaca perubahan radiasi lingkungan dengan adanya pancaran radiasi manusia, sensor akan mengirimkan sinyal 1 yang akan diterima dan diproses oleh *Raspberry Pi*. Sinyal akan diolah untuk mengambil gambar dengan USB *Camera* dan disimpan pada r*aspberry Pi*. Gambar akan dikirimkan ke *email* melalui jaringan internet sebagai notifikasi peringatan berupa teks, sedangkan gambar akan dikirimkan ke *website*.

III. METODE PENELITIAN

Tahapan penelitian yang dilakukan dalm proyek akhir ini pertama yaitu studi literatur dari beberapa jurnal dan buku, melakukan perancangan dan analisa sistem kebutuhan. Melakukan persiapan untuk perangkat keras yang dibutuhkan dan perangkat lunak. Dilanjutkan dengan tahap instalasi sistem operasi raspbian instalasi web camera, membuat Bot Telegram, Konfigurasi sensor ultrasonic, konfigurasi capture data, konfigurasi pengiriman data ke server telegram, dan konfigurasi sistem kendali pintu. Gambar 3.1 adalah diagram alir metode penelitian yang dilakukan oleh penulis.



Gambar 3.1 Diagram Alir Penelitian

3.1 Alat dan Bahan

Dalam melaksanakan penelitian ini, terdapat beberapa perangkat keras dan perangkat lunak yang dibutuhkan guna menunjang penelitian yang dilakukan. Adapun beberapa perangkat tersebut diantaranya sebagai berikut.

3.1.1 Perangkat Keras

1. Satu (1) Monitor
2. Satu (1) *Mini* PC/*Raspberry Pi*
3. Satu (1) *Smartphone*

4. Satu (1) Sensor Ultrasonik HC-SR04
5. Satu (1) *Web Camera* Sturdy
6. Satu (1) *Solenoid Door Lock* 12V

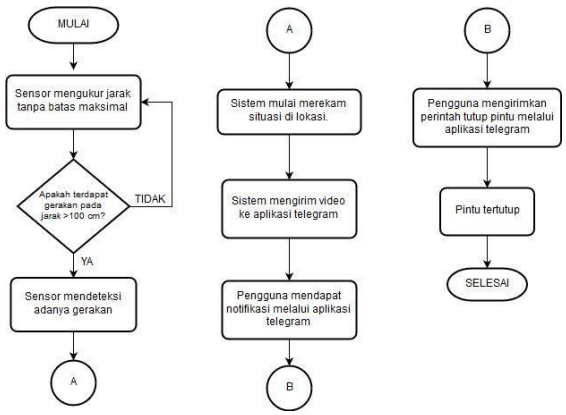
3.1.2 Perangkat Lunak

1. Sistem Operasi Raspbian
2. *Python*
3. Telegram

3.2 Cara Kerja Sistem

Gambar 3.2 merupakan cara kerja dari *smart door*. Sensor ultrasonik yang diletakkan pada daerah pintu akan memancarkan gelombang lurus dengan jarak maksimal 400 cm. Sistem yang dibuat oleh penulis hanya dapat mendeteksi objek pada jarak maksimal 100 cm. Ketika terdapat objek yang berada pada jarak kurang dari 100 cm, maka gelombang pantulan dari objek akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima. Apabila sensor telah mendapatkan nilai jarak objek, *Raspberry Pi* memerintahkan *web camera* untuk merekam keadaan di daerah pintu. Ketika video sudah mencapat durasi yang diinginkan, *Raspberry Pi* mulai mengirimkan video tersebut ke Telegram.

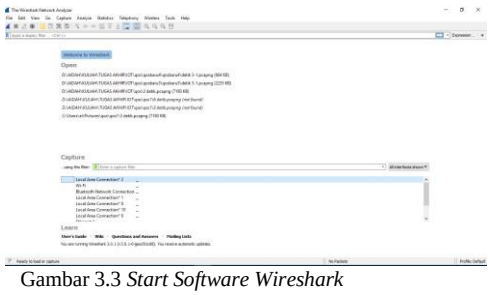
Dimana Telegram yang akan mendapat notifikasi,video hanya Telegram yang *user* ID nya sudah didaftarkan pada sistem di *Raspberry Pi*. Sehingga tidak semua pengguna Telegram akan mendapatkan notifikasi tersebut. Setelah pengguna mendapatkan notifikasi, pengguna kemudian mengirimkan perintah "*close*" untuk menutup atau mengunci pintu.



Gambar 3.2 Diagram Alir Cara Kerja Sistem

3.3 Cara pengambilan data

a) Buka Software Wireshark untuk melihat paket data yang masuk Pada Jaringan Wifi dan Klik jaringan yang sudah Terhubung yaitu Wireless Network Connection selanjutnya pilih Capture Lalu Klik Start. Seperti pada Gamar 3.3.

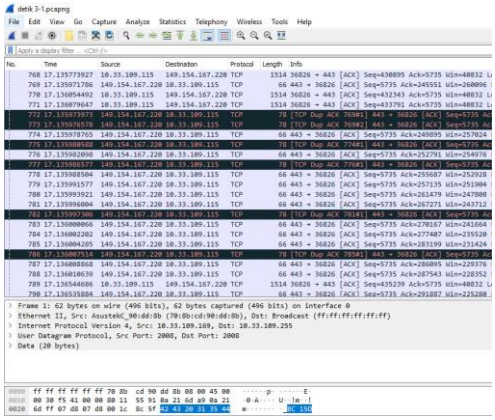


Gambar 3.3 *Start Software Wireshark*

b) Pada Raspberry Pi, menjalankan program yang sudah dibuat. Ketika terapat objek berjarak kurang dari 100 cm, maka sistem akan menampilkan pesan "mendeteksi gerakan" kemudian memulai proses rekam video seperti pada Gambar 3.4. Pada Gambar 3.4 dapat dilihat, setelah sistem menampilkan pesan jarak yang dibaca dan pesan "Mendeteksi Gerakan", sistem membuka perangkat untuk merekam yaitu /dev/video0. Kemudian sistem mulai melakukan *capture frame* yang dimulai dari detik ke-0. Hasil *capture* akan disimpan pada folder /home/pi/camera/.

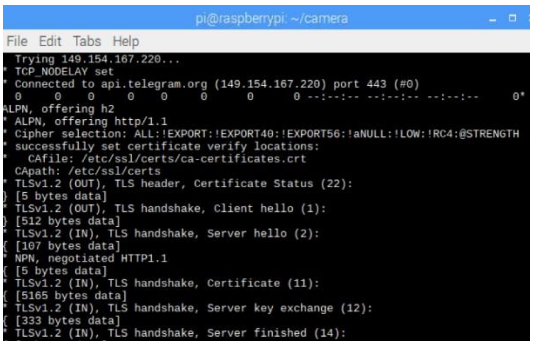
Gambar 3.4 Menjalankan program *smart door*

c) Lalu Wireshark akan langsung menangkap paket paket data yang berada pada jaringan LAN, biarkan software wireshark berjalan sampai waktu yang akan kita tentukan seperi pada Gambar 3.5.

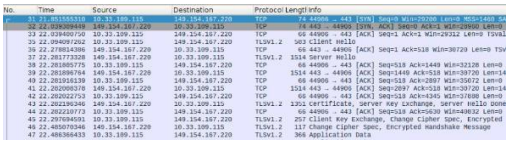


Gambar 3.5 Capture Paket Data *Wireshark*

Gambar 3.6 merupakan proses yang terjadi pada *Raspberry Pi* saat data dikirim ke *server* telegram. Adanya pesan "Trying 149.154.167.220" menunjukkan bahwa *Raspberry Pi* sedang meminta koneksi ke server telegram (*ip server* telegram : 149.154.167.220). Sedangkan Gambar 3.7 merupakan proses monitoring *wireshark* ketika pengiriman data dari *Raspberry Pi* ke *server* telegram. Pada *wireshark* akan menampilkan pesan [SYN] yang berarti meminta koneksi SSL TCP ke port 443 milik protokol HTTPS. TCP dikatakan terhubung ketika *flag* nya berubah menjadi [ACK]. Ketika protokol berubah menjadi TLSv1.2, berarti sertifikat SSL (*Secure Sockets Layer*) sudah berhasil diverifikasi**.** Tanda bahwa *Raspberry Pi* sudah terhubung dengan *server* telegram adalah ketika sudah muncul pesan "Server Hello Done" di baris ke-43, yang artinya *server* dari Telegram sudah dapat berkomunikasi dengan *Raspberry Pi*. Sehingga *Raspberry Pi* mulai mengirimkan data video nya pada detik ke22 seperti pada Gambar 3.7.

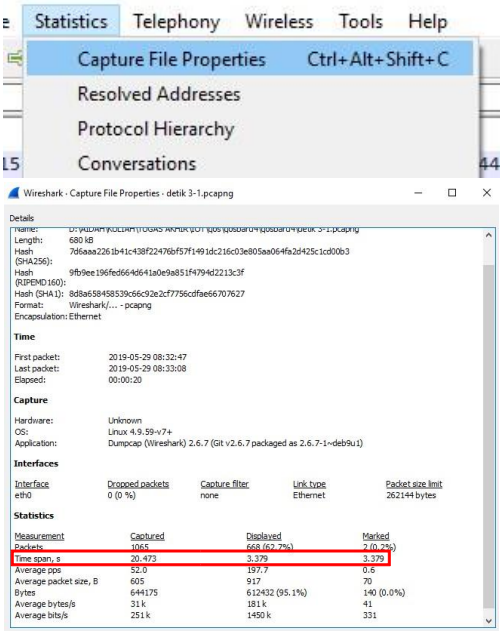


Gambar 3.6 Proses untuk Dapat Berkomunikasi dengan *Server* Telegram pada *Raspberry Pi*



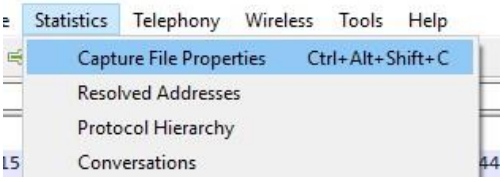
Gambar 3.7 Proses untuk Dapat Berkomunikasi dengan *Server* Telegram pada *Wireshark*

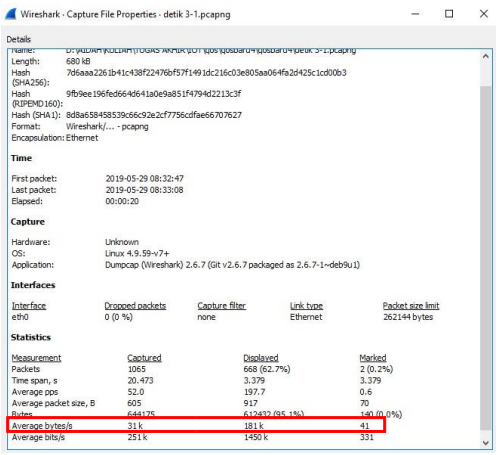
d) Untuk mengetahui data *delay*, masuk ke menu *statistic* ❼ *Capture File Properties* di bagian *Time Span* seperti pada Gambar 3.8.



Gambar 3.8 Capture Data *Delay*

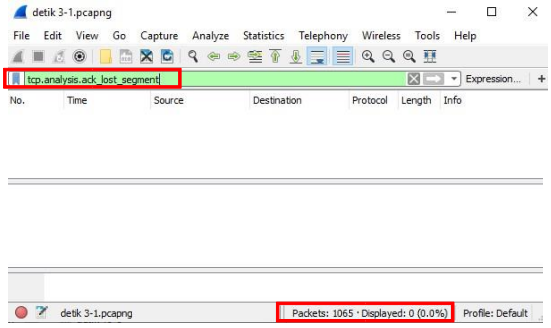
e) Untuk mengetahui data *throughput*, masuk ke menu *statistic* ❼ *Capture File Properties* di bagian *Averages* seperti pada Gambar 3.9.





Gambar 3.9 Capture Data *Throughput*

f) Untuk mengetahui data *packet loss*, dapat diambil dari aplikasi *wireshark* dengan cara melakukan filterisasi "tcp.analysis.ack_lost_segment".
seperti pada Gambar 3.10.

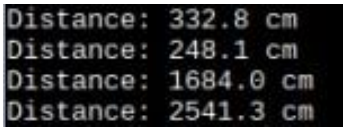


Gambar 3.10 Capture Data *Packet Loss*

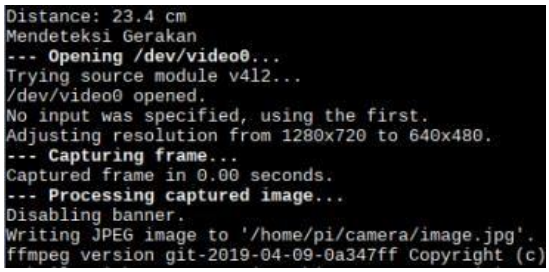
IV. HASIL DAN PEMBAHASAN

4.1 Pengujian Notifikasi Telegram

Pada saat sensor tidak terhalang objek, maka pada Raspberry Pi hanya akan menampilkan informasi jarak baca sensor seperti pada Gambar 4.1.



Gambar 4.1 Pembacaan Sensor Ultrasonik Diatas 100 cm Namun, apabila jarak objek kurang dari 100 cm, maka sistem akan menampilkan pesan "mendeteksi gerakan" kemudian memulai proses rekam video seperti pada Gambar 4.2. Pada Gambar 4.2 dapat dilihat, setelah system menampilkan pesan jarak yang dibaca dan pesan "Mendeteksi Gerakan", sistem membuka perangkat untuk merekam yaitu /dev/video0. Kemudian sistem mulai melakukan *capture frame* yang dimulai dari detik ke-0. Hasil *capture* akan disimpan pada folder
/home/pi/camera/.



Gambar 4.2 Membuka Perangkat Video pada *Raspberry Pi*

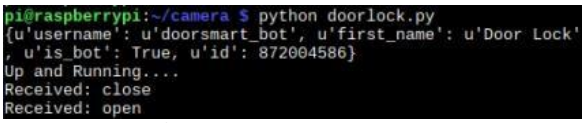
Setelah *capture* selesai, maka sistem akan mengirimkan video dengan format GIF. Penulis menggunakan format GIF karena tujuan dari sistem hanya ingin melihat objek saja tanpa perlu menggunakan suara. Sistem juga mengirimkan pesan "Terdeteksi" seperti pada Gambar 4.3.

Gambar 4.3 Notifikasi pada aplikasi Telegram

3.2 Pengujian Sistem Kendalo Pintu Jarak Jauh Menggunakan Solenoida

Untuk mengaktifkan solenoid, ditambahkan *relay* pada rangkaian sebagai saklar atau elektromagnetik *switch* yang kinerjanya dikendalikan oleh magnet listrik. Gambar 4.4 merupakan perintah membuka atau menutup pintu yang dikendalikan melalui aplikasi Telegram. Ketika pengguna mengirimkan sebuah perintah *close* atau *open*, pada *Raspberry Pi* akan menampilkan informasi bahwa pengguna mengirim pesan *close* atau *open* seperti pada Gambar 4.5. Apabila dikirimkan perintah "*close*" maka *solenoid* akan memanjang (merupakan cara kerja solenoid NC / *Normally Close*). Dan apabila dikirimkan perintah "*open*" maka *solenoid* akan memendek (merupakan cara kerja solenoid NO / *Normally Open*)

Gambar 4.4 Perintah Buka / Tutup Pintu melalui Telegram



Gambar 4.5 Informasi pada *Raspberry Pi*

4.3 Pengujian *Akurasi*

Akurasi yang diuji dalam penelitian ini adalah akurasi sensor HC-SR04 dalam membaca jarak, dimana nilai akurasi didapatkan dengan perhitungan *relative error*, dimana nilai membaca jarak dari sensor HC-SR04 dibandingkan dengan nilai jarak nyata yang dihitung menggunakan penggaris seperti pada Gambar 4.6. Untuk jarak yang diuji adalah 25 cm, 50 cm, 75 cm, dan 100 cm.

Gambar 4.6 Objek berjarak 25 cm dari sensor

Tabel 4.1 Hasil Pengujian *Relative Error* Akurasi

Percobaan ke- n	Jarak ke - n			
	25 cm	50 cm	75 cm	100 cm
1	24,5	50,0	74,4	87,2
2	24,5	50,0	75,7	98,5
3	24,6	50,4	74,8	98,0
4	24,8	50,1	74,8	94,6
5	24,8	50,0	74,3	99,0
6	25,0	50,2	75,0	98,8
7	25,0	50,0	75,0	100,0
8	24,5	50,0	75,0	99,8
9	25	50,2	75,5	100,0
10	25,1	50,0	75,2	97,5
Rata-rata	24,82	50,09	74,97	97,34
***Relative Error* Akurasi (%)**	0,72	0,18	0,04	2,66

Dapat dilihat dari Tabel 4.1, bahwa sensor tidak selalu bernilai sama dengan jarak sebenarnya namun mendekati. Untuk mendapat nilai *relative error*, digunakan rumus *Relative Error* (%) = $\frac{|AE|}{AV} X\ 100$, dimana AE adalah *Absolute Error* atau kesalahan pembacaan sensor yang bersifat mutlak, dan AV adalah *Actual Value* atau nilai sebenarnya.Berikut contoh menghitung menghitung nilai *relative error* pada jarak 25 cm.

$$Absolute\ Error\ (AE) = 25 - 24{,}82$$

$$= 0{,}18$$

$$Relative\ Error\ (\%) = \frac{|AE|}{AV}\ X\ 100$$

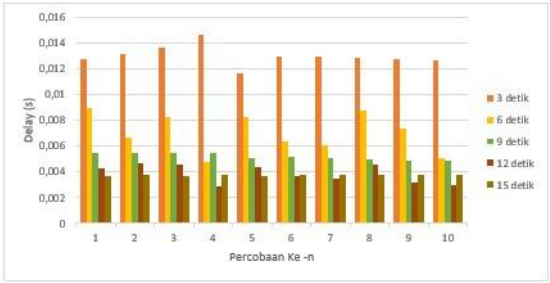
$$= \frac{0{,}18}{25}\ x\ 100$$

$$= 0{,}72\ \%$$

4.2 Pengujian *Delay*

Tabel 4.2 Hasil Pengujian *Delay*

Percobaan Ke- n	Delay Pada Durasi n Detik				
	3	6	9	12	15
1	0,012723	0,008888	0,005425	0,004222	0,003641
2	0,013127	0,006666	0,005455	0,004637	0,003781
3	0,013598	0,008237	0,005393	0,004493	0,00363
4	0,014568	0,004748	0,005414	0,002845	0,003773
5	0,011664	0,008199	0,005048	0,004316	0,003653
6	0,012932	0,006358	0,005108	0,003621	0,003791
7	0,012865	0,006029	0,005029	0,003461	0,003702
8	0,012797	0,008699	0,004949	0,004519	0,003703
9	0,012729	0,007374	0,004874	0,00314	0,003705
10	0,012662	0,00504	0,004795	0,002979	0,003706
Rata-rata *Delay* (s)	0,0129665	0,0070238	0,005149	0,0038233	0,0037085



Gambar 4.7 Grafik Data *Delay*

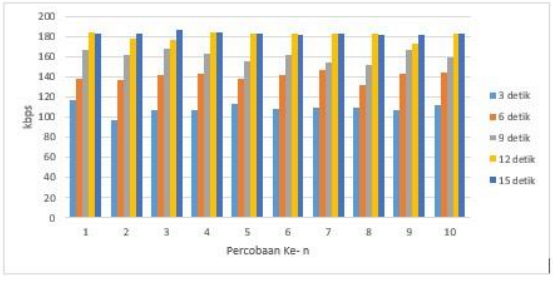
Dapat dilihat pada Tabel 4.2, terdapat lima kali percobaan dengan video berdurasi 3 detik, 6 detik, 9 detik, 12 detik, dan 15 detik. Rata-rata *delay* dalam detik secara urut untuk durasi 3 detik, 6 detik, 9 detik, 12 detik, dan 15 detik adalah 0,0129665 detik, 0,0070238 detik, 0,005149 detik, 0,0038233 detik, dan 0,0037085 detik. Sedangkan rata-rata *delay* dalam satuan milidetik secara urut adalah 12,9665 milidetik, 7,0238 milidetik, 5,149 milidetik, 3,8233 milidetik, dan 3,7085 milidetik.

Dari data tersebut, membuktikan bahwa *delay* pengiriman dari *Raspberry Pi* menuju ke *server* telegram termasuk kategori ***Excellent*** menurut kategori OWD pada Tabel 2.4 karena ratarata *delay* pada penelitian ini kurang dari 150 milidetik atau 0,15 detik.

4.3 Pengujian *Throughput*

Tabel 4.3 Hasil Pengujian *Throughput*

Percobaan Ke- n	Throughput Pada Durasi n Detik				
	3	6	9	12	15
1	116	138	166	184	183
2	97	136	161	178	182
3	107	142	167	176	186
4	106	143	163	184	184
5	113	138	155	183	182
6	108	141	161	182	181
7	109	146	154	182	183
8	109	132	152	183	183
9	106	143	166	172	181
10	111	144	159	183	188
Rata-rata *Throughput* (kbps)	108,2	140,3	160,4	180,7	182,5



Gambar 4.8 Grafik Data *Thoughput*

Dari Tabel 4.4 dapat dilihat bahwa nilai *throughput* saat terjadi pengiriman data dengan durasi 3 detik berkisar 97 kbps hingga 116 kbps. Pada pengiriman berdurasi 6 detik *throughput* yang dihasilkan berada pada angka kisaran 132 kbps hingga 146 kbps. Kemudian pada pengiriman data dengan durasi 9 detik *throughput* berkisar antara 152 kbps hingga 167 kbps. Lalu pada pengiriman data dengan durasi 12 detik *throughput* berkisar antara 172 kbps hingga 184 kbps. Dan pada pengiriman data dengan durasi 15 detik *throughput* berkisar antara 181 kbps hingga 186 kbps.

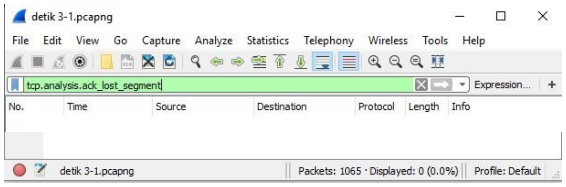
Apabila dilakukan perbandingan antara *delay* dengan *throughput*, nilai *throughput* mempengaruhi nilai *delay*. Dapat dilihat pada Tabel 4.3 dengan Tabel 4.4, Rata-rata dari seluruh percobaan *delay* dan *throughput* menghasilkan hasil yang berkebalikan. Pada durasi 3 detik *throughput* yang dihasilkan sebesar 108,2 kbps dan *delay* yang dihasilkan sebesar 0,0129665 detik, lalu pada pengiriman video berdurasi 6 detik mengalami kenaikan *throughput* menjadi 140,3 kbps dan nilai *delay* turun menjadi 0,0070238 detik, kemudian pada pengiriman video berdurasi 9 detik *throughput* mengalami kenaikan menjadi 160,4 kbps dan nilai *delay* turun menjadi 0,005149 detik. Pada pengiriman video berdurasi 12 detik *throughput* mengalami kenaikan menjadi 180,7 kbps sedangkan *delay* turun menjadi 0,0038233 detik. Dan pada pengiriman video berdurasi 15 detik, *throughput* mengalami kenaikan lagi menjadi 182,5 kbps sedangkan *delay* turun menjadi 0,0037085 detik.

Dari data tersebut, dapat disimpulkan bahwa semakin besar *throughput*, maka semakin kecil *delay* yang dihasilkan. Yang artinya, semakin besar *throughput*, maka pengiriman data pun semakin cepat.

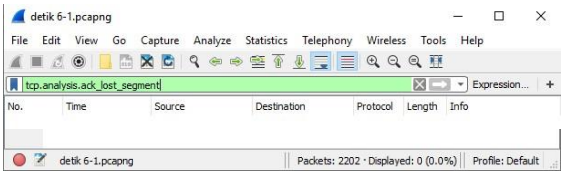
4.4 Pengujian *Packet Loss*

Penentuan kualitas baik atau buruk parameter *packet loss* dalam penelitian ini menggunakan standar dari *Telecommunications Protocol Harmonization Over Network* (TIPHON). Dalam pengambilan data *packet loss* dilakukan variasi perubahan durasi video dalam pengambilan data, variasi tersebut adalah 3 detik, 6 detik, 9 detik, 12 detik, dan 15 detik.

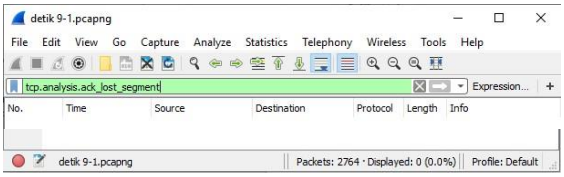
Untuk mendapatkan data *packet loss*, dapat diambil dari aplikasi *wireshark* dengan cara melakukan filterisasi "tcp.analysis.ack_lost_segment" pada bagian kolom filter seperti pada Gambar 4.9 untuk durasi video 3 detik, Gambar 4.10 untuk durasi video 6 detik, Gambar 4.11 untuk durasi video 9 detik, Gambar 4.12 untuk durasi video 12 detik, dan Gambar 4.13 untuk durasi video 15 detik, yang diberi tanda merah. Apabila tidak ada paket yang ditampilkan, maka tidak ada paket yang hilang selama pengiriman paket data, atau *packet loss* nya bernilai 0%.



Gambar 4.9 Hasil *Capture Packet Loss* Durasi 3 detik Percobaan Pertama

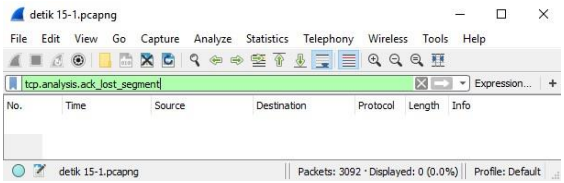


Gambar 4.10 Hasil *Capture Packet Loss* Durasi 6 detik Percobaan Pertama



Gambar 4.11 Hasil *Capture Packet Loss* Durasi 9 detik Percobaan Pertama

Gambar 4.12 Hasil *Capture Packet Loss* Durasi 12 detik Percobaan Pertama



Gambar 4.13 Hasil *Capture Packet Loss* Durasi 15 detik Percobaan Pertama

Setelah dilakukan percobaan 10 kali untuk tiap durasi, packet loss yang didapat bernilai 0% atau dapat dikatakan tidak ada paket yang hilang selama pengiriman berlangsung. Sehingga dapat disimpulkan bahwa packet loss untuk pengiriman data dari *publisher* ke *subscriber t*ermasuk kategori **sangat bagus**.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari keseluruhan kegiatan penelitian yang telah dilakukan pada tugas akhir ini, dapat diambil beberapa kesimpulan sebagai berikut:

1. Durasi video mempengaruhi QoS dalam mengirimkan data ke *server* telegram pada sistem *smartdoor*. Semakin lama durasi video, maka ukuran paket data semakin besar, hal tersebut berpengaruh pada nilai *throughput* dan *delay.*
2. *Delay* yang dihasilkan untuk durasi 3 detik hingga 15 detik mengalami penurunan dari 0,012 detik menjadi 0,003 detik. Sedangkan *throughput* untuk durasi 3 detik hingga 15 detik mengalami kenaikan dari 108,2 kbps menjadi 182,5 kbps. Sedangkan *packet loss* bernilai 0% selama pengujian berlangsung baik pada durasi 3 detik, 6 detik, 9 detik, 12 detik, dan 15 detik.
3. Nilai *throughput* mempengaruhi kecepatan pengiriman data. Ketika *throughput*nya besar, *delay* yang dihasilkan kecil sehingga data yang dikirim lebih cepat sampai.
4. Keakurasian Sensor ultrasonik HC-SR04 yang digunakan penulis masih belum stabil atau masih belum baik. Sensor dapat membaca jarak dengan stabil dalam rentang 3 cm – 99 cm.
5. Sistem yang dibuat berupa alat deteksi gerakan berupa sensor ultrasonik yang dilengkapi *web camera* untuk merekam objek yang berada pada akses pintu masuk rumah. Dimana ketika sensor mendeteksi adanya gerakan dengan jarak maksimal 100 cm dari sensor, *Raspberry Pi* bekerja dengan mengirimkan notifikasi berupa video dan pesan "Terdeteksi" ke aplikasi telegram.
6. *Solenoid Door Lock* bekerja ketika pengguna atau pemilik rumah mengirimkan perintah "*close*" atau "*open*" melalui aplikasi telegram.

5.2 Saran

Berikut adalah beberapa saran yang dapat digunakan untuk mengembangkan penelitian berikutnya:

1. Menambahkan konfigurasi agar sistem dapat melakukan rekognisi wajah.
2. Melakukan perubahan pada sesnsor dengan menggunakan sensor yang keakurasiannya lebih stabil daripada sensor HC-SR04.

DAFTAR PUSTAKA

[1] E. N. Wijatsongko, "Sistem Pemantau Ruangan Berbasis Video Streaming dengan Server Raspberry Pi," Universitas Gadjah Mada, Yogyakarta, 2014.
[2] A. A. Arman, 9 April 2008. [Online]. Available: https://kupalima.wordpress.com/2008/04/09/pervasivecomputing/.
[3] E. R. Subhiyakto, D. W. Utomo and P. W. Adi, "Teknologi dan Teknik Sistem Terdistribusi Pervasif dalam Bidang Logistik: Studi LiteraturSistematis," *Jurnal Buana Informatika*, pp. 83-94, 2016.
[4] X. Li, Y. Zhou, C. Ai and L. Qian, "ICMTMA '14 Proceedings of the 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation," *IEEE Computer Society Washington, DC, USA ©2014*, pp. 589-592, 2014.
[5] E. N. Wijatsongko, "Sistem Pemantau Ruangan Berbasis Video Streaming dengan Server Raspberry Pi," Universitas Gadjah Mada, Yogyakarta, 2014.
[6] J. Ero, "Sistem Monitoring Berbasis Live Video Streaming dan Dilengkapi Notifikasi SMS," Universitas Gadjah Mada, Yogyakarta, 2009.
[7] P. P. Rahmanto, "Prototype Kendali Lampu Ruma Berbasis Arduino dengan Menggunakan SMS (Short Message Service)," Universitas Gadjah Mada, Yogyakarta, 2016.
[8] S. Ahadiah, Muharnis and Agustiawan, "Implementasi Sensor Pir Pada Peralatan Elektronik Berbasis Microcontroller," *JURNAL INOVTEK POLBENG*, pp. VOL. 07, NO. 1, 2017.
[9] A. Puliano, "Sistem Pemantauan Ruangan Menggunakan Raspberry Pi Berbasis IoT," Universitas Gadjah Mada, Yogyakarta, 2017.

IMPLEMENTASI DAN ANALISIS PERFORMA PROTOKOL *MESSAGE QUEUING TELEMETRY TRANSPORT (MQTT) PROTOCOL* JARINGAN *SMART FARMING* PADA BUDIDAYA JAMUR TIRAM DENGAN MEMANFAATKAN *INTERNET OF THINGS*

Sidiq Rilo Pambudi and Alif Subardono
Departemen Teknik Elektro dan Informatika
Sekolah Vokasi
Universitas Gadjah Mada
sidiq.rilo.p@mail.ugm.ac.id, alif@ugm.ac.id

Abstract – The development of technology, gives many effects for human life. Many things can be controlled automatically and remotely. Internet of Things is as one of technology development which can be a solution to make human life easier. Internet of Things can also be used in agriculture sector and smart farming.

Oyster growth is very affected by temperature and humidity. The temperature needed for growing of oyster is 24-28 Celsius degree. The use of Internet of Things is needed to monitor, take care and increase the productivity. Implementation of Smart Farming in agriculture sector, such as the using of protocol Message Queuing Telemetry Transport (MQTT) as a media of data traffic among censor, server and receiver of data. Message Queuing Telemetry Transport (MQTT) as protocol which can work as real time and work in low bandwidth, so the traffic of data works smoothly although in low bandwidth

Keywords: Internet Of Things , Smart Farming, MQTT, QoS

Intisari – Perkembangan teknologi memberikan banyak dampak pada kehidupan manusia. Banyak hal dapat dikontrol otomatis dari jarak jauh. *Internet of Things (IoT)* sebagai teknologi yang dapat menjadi solusi untuk mempermudah pekerjaan manusia. *Internet of Things (IoT)* dapat digunakan pula di bidang pertanian atau perkebunan pintar (*Smart Farming).*

Tanaman Jamur Tiram sangat dipengaruhi oleh temperatur dan kelembaban. Kondisi udara saat pertumbuhan Jamur Tiram berada pada suhu 24-28 derajat Celcius. Penggunaan *Internet of Thing* (IoT) di bidang pertanian dapat mendukung pemantauan, perawatan dan meningkatkan produktifitas. Penerapan Smart Farming di bidang pertanian salah satunya dengan menggunakan *protocol Message Queuing Telemetry Transport* (MQTT) sebagai sarana lalu lintas data antara sensor, server dan penerima data. *Message Queuing Telemetry Transport* (MQTT) merupakan *protocol* yang dapat bekerja secara realtime dan dapat bekerja pada bandwidth yang rendah, sehingga lalu lintas data tetap lancar walaupun menggunakan bandwidth yang rendah.

Kata Kunci : *Internet Of Things , Smart Farming, MQTT, QoS*

I. PENDAHULUAN

Berkembangnya teknologi dan perangkat jaringan yang lain tidak hanya sebagai alat untuk bekerja. Namun perangkat jaringan saat ini semakin berkembang dan dapat sebagai pengganti atau membantu kehidupan sehari-hari maupun dalam bidang pertanian, pendidikan, maupun di bidang yang lain, yaitu dengan perangkat internet pintar.

Dalam bidang pertanian juga harus didukung dengan teknologi terbaru supaya meningkatkan produktivitas dari hasil pertanian tersebut. Seperti tanaman Jamur Tiram yang membutuhkan keadaan suhu dan kelembaban ruang yang baik supaya tumbuh dengan sempurna. Karena pada saat ini untuk pemantauan suhu dan penyiraman tanaman Jamur Tiram masih belum otomatis, dengan sensor suhu diharapkan dapat memantau suhu dan kelembaban, serta penggunaan penyemprot tanaman otomatis.

II. KAJIAN PUSTAKA

2.1 Internet of Things

IoT saat ini sedang berkembang dalam penggunaan di kehidupan sehari-hari. Dengan Teknologi komunikasi nirkabel yang mampu menghasilkan jaringan skala besar, karena teknologi IoT dapat menghubungkan fungsi sistem keseluruhan dan komunikasi dalam seluruh objek dapat terhubung. [1].

2.2 Jamur Tiram

Jamur Tiram atau *Oyster Mushroom* merupakan tanaman yang dibudidayakan petani, karena bentuk dari jamur, seperti dengan cangkang tiram. Jamur Tiram rata-rata berdiameter 3-15 cm, sebagian jamur memiliki tangkai bercabang dan tubuhnya berwarna putih. [2]

2.3 Smart Farming

Smart Farming (Pertanian cerdas) menurut [3] adalah konsep yang muncul yang mengacu pada pengelolaan pertanian menggunakan teknologi modern. Bertujuan meningkatkan kuantitas dan kualitas produk pertanian serta mengoptimalkan tenaga kerja manusia yang dibutuhkan.

2.4 Protokol Message Queuing Telemetry Protocol

MQTT disebut *lightweight* protokol karena dalam penerapannya menggunakan pesan berukuran 2*bytes* pada jenis data MQTT. Protokol ini dapat diaplikasikan dengan lebar pita dan sumber daya yang kecil, MQTT memiliki kelebihan menjamin semua data tetap akan terkirim walaupun koneksi terputus sementara [4]. Komponen utama pada protokol MQTT adalah *broker*, komponen ini sangat penting dalam protkol yang memiliki arsitektur *Publisher/subsciber*. *Broker* berfungsi sebagai perantara pertukaran pesan antara *publisher* dan *subscriber*.

2.5 Quality of Service

Quality of Service digunakan untuk mengetahui kualitas dari suatu layanan dengan menggunakan beberapa parameter sebagai penilaian kualitas

2.5.1 Packet Delivery Loss

Packet Loss Ratio adalah jumlah paket yang tidak diterima dibandingkan dengan jumlah seluruh paket yang dikirimkan atau di transmisikan

$$PLR = \frac{\text{paket terkirim} - \text{paket diterima}}{\sum \text{paket terkirim}} \; x \; 100\%$$

2.5.2 Packet Delivery Ratio

Packet Delivery Ratio adalah hasil prosentase keberhasilan dari jumlah paket diterima oleh penerima. Parameter ini merupakan parameter penunjang parameter *packet loss*, sehingga standar yang digunakan adalah kebalikan dari standar paket loss.

$$PDR\ (\%) = \frac{\sum \text{Paket diterima}}{\sum \text{Paket terkirim}} x100\%$$

2.5.3 Delay

Delay adalah jumlah waktu keseluruhan dalam proses satu kali pengamatan, waktu yang dibutuhkan dalam satu kali pengiriman paket data dibagi dengan keseluruhan usaha pengiriman paket yang berhasil dalam pengamatan

$$delay = \frac{\sum \text{waktu pengiriman dalam satu kali pengamatan}}{\sum \text{usaha pengiriman paket berhasil}}$$

2.5.4 Throughput

Throughput adalah istilah yang menjelaskan banyak bit yang diterima dalam jangka waktu tertentu dengan satuan *bit per second* yang diperoleh dari nilai data sebenarnya.

$$throughput = \frac{\sum \text{paket yang berhasil dikirim x ukuran paket}}{\text{total waktu pengiriman}}$$

2.6 Sensor DHT 11

Sensor DHT 11 adalah modul yang digunakan untuk melakukan pembacaan suhu dan kelembaban. Modul DHT 11 dapat membaca suhu dalam waktu 1 detik

2.7 Nodemcu 8266

NodeMcu 8266 adalah perangkat pengontrol *system* yang memiliki kompabilitas dan memiliki kemudahan dalam pemrogramannya. *Nodemcu* adalah sebuah perangkat yang *open source*, perangkat ini merupakan mikrokontroler yang terintegrasi dengan wifi, sehingga tidak perlu menambahkan modul wifi tambahan. [5]

2.8 Raspberry Pi Model 3B

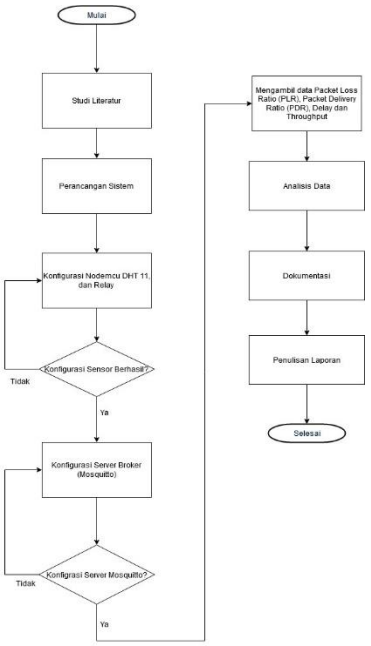
Raspberry Pi 3 merupakan generasi ketiga, memiliki ukuran yang kecil dan kuat. Memiliki prosessor yang lebih baik dari versi sebelumnya. Model B memiliki konektivitas *wireless* LAN, LAN dan *Bluethooth* yang hemat energi. Raspberry dapat digunakan untuk *Mini Server Cloud*, *Mini Server*, *Server* DNS dan lain sebagainya

2.9 Wireshark

Wireshark adalah salah satu alat untuk menganalisa jaringan, memfilter protokol, paket. Aplikasi ini banyak digunakan oleh *Network Engineer*, *Network Analyst*, *Network Administrator*.

III. METODE PENELITIAN

3.1 Tahapan Penelitian



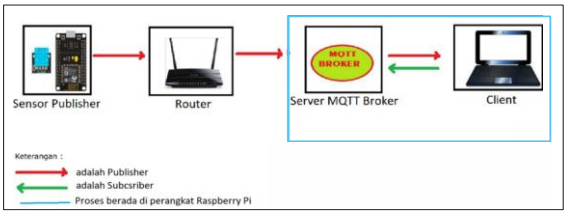
Gambar 3. 1 Tahapan penelitian

3.2 Alat dan Bahan

1. Perangkat Keras
   - Rasberry Pi 3 Model B
   - NodeMcu ESP 8266
   - Modul Relay
   - Sensor Suhu DHT 11
   - Router Tenda F3
   - Jamur Tiram
   - Lampu
   - Pompa air 5v
   - Kipas DC 12v

2. Perangkat Lunak
   - Raspbian OS
   - Arduino IDE
   - Mosquitto (MQTT Broker)
   - Wireshark

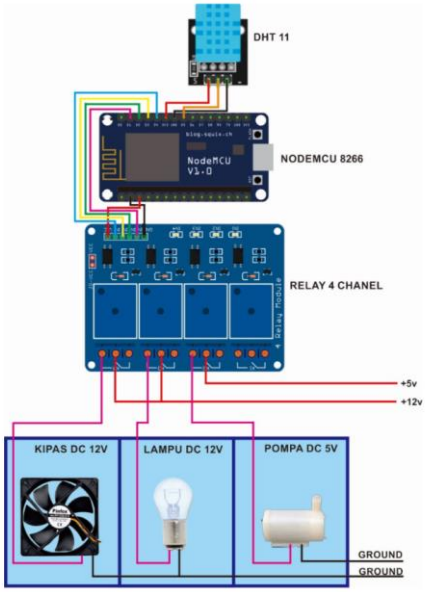
3.3 **Perancangan Topologi Alur Kerja Sistem**



Gambar 3. 2 Alur kerja sistem

Pada Gambar 3.3 menjelaskan alur kerja sistem dengan Sensor Publisher terdiri dari Nodemcu ESP 8266, sensor suhu DHT 11, Relay, Pompa, Kipas, dan lampu. Kemudian di tengah terdapat router Tenda F3 sebagai jembatan lalu lintas data. Kemudian pada kotak berwarna biru merupakan proses yang terjadi pada perangkat Raspberry Pi, yaitu Server MQTTBroker dan Client.

3.4 Perancangan Sistem Sensor



Gambar 3. 3 Perangkat pada *publisher*

Pada perangkat *publisher* Nodemcu ESP 8266 bertugas sebagai inti dari pengendalian perangkat sensor DHT 11, Relay (kipas, lampu, pompa), dan jaringan

3.5 Skenario Pengujian

1. Skenario 1 (Jarak pengujian 5 meter dengan umur tanaman Jamur Tiram yang berbeda)

Gambar 3. 4 Jarak antar perangkat 5 meter

2. Skenario 2 (Jarak pengujian 15 meter dengan umur tanaman Jamur Tiram yang berbeda)

Gambar 3. 5 Jarak antar perangkat 15 meter

3. Skenario 3 (Jarak pengujian 25 meter dengan umur tanaman Jamur Tiram yang berbeda)

Gambar 3. 6 Jarak antar perangkat 25 meter

3.6 Pengujian sistem

Pada pengujian sistem perangkat tediri dari 3 kotak berisi jamur dan pada kotak sudah terdapat rangkaian seperti pada Gambar 3.4, Kemudian pada bagian tengan merupaan router sebagai jembatan antara *publisher* dan *subscriber*. Pada bagian perangkat Raspberry Pi terdapat *server* MQTT dan *Client.*

IV. ANALISIS DAN PEMBAHASAN

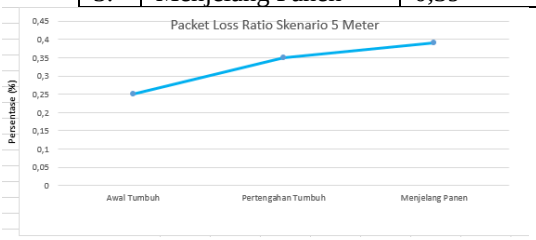
Analisis data dilakukan berdasarkan kumpulan data yang telah diambil berdasarkan parameter pada setiap scenario pengujian yang telah ditentukan. Adapun analisis yang didapatkan :

4.1 Skenario 1 (Jarak pengujian 5 meter dengan umur tanaman Jamur Tiram yang berbeda)

1. Packet Loss Ratio

Tabel 4. 1 Hasil Packet Loss Ratio Skenario 1

No	Umur Tanaman	Persentase (%)
1.	Awal Tumbuh	0,25
2.	Pertengahan Tumbuh	0,35
3.	Menjelang Panen	0,39



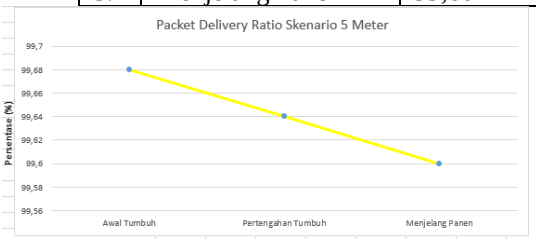
Gambar 4. 1 Grafik Packet Loss Ratio Skenario 1

Dari hasil pengamatan *Packet Loss Ratio* pada skenario 1 didapat hasil seperti pada Tabel 4.1 dan ditampilkan pada Gambar 4.1 didapat dengan standarisasi *TIPHON* sangat baik.

2. Packet Delivery Ratio

Tabel 4. 2 Hasil *Packet Delivery Ratio* Skenario 1

No	Umur Tanaman	Persentase (%)
1.	Awal Tumbuh	99,68
2.	Pertengahan Tumbuh	99,64
3.	Menjelang Panen	99,60



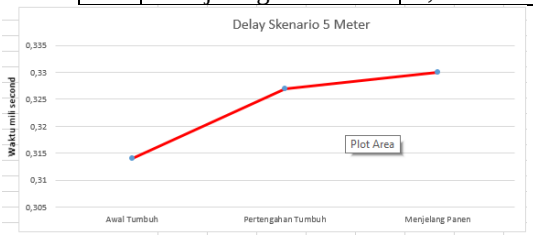
Gambar 4. 2 Hasil *Packet Delivery Ratio* Skenario 1

Dari hasil pengamatan *Packet Delivery Ratio* pada skenario 1 didapat hasil seperti pada Tabel 4.2 dan ditampilkan pada Gambar 4.2 didapat dengan standarisasi *TIPHON* sangat baik.

3. Delay

Tabel 4. 3 Hasil Delay Skenario 1

No	Umur Tanaman	*mili second*
1.	Awal Tumbuh	0,314
2.	Pertengahan Tumbuh	0,327
3.	Menjelang Panen	0,330



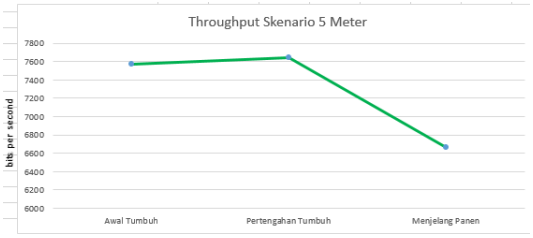
Gambar 4. 3 Hasil *Delay* Skenario 1

Dari hasil pengamatan *Delay* pada skenario 1 didapat hasil seperti pada Tabel 4.3 dan ditampilkan pada Gambar 4.3 didapat dengan standarisasi *TIPHON* sangat baik.

4. Throughput

Tabel 4. 4 Hasil Throughput Skenario 1

No	Umur Tanaman	*bits per second*
1.	Awal Tumbuh	7572
2.	Pertengahan Tumbuh	7648
3.	Menjelang Panen	6659



Gambar 4. 4 Hasil *Throughput* Skenario 1

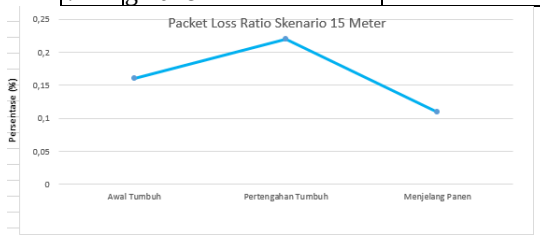
Dari hasil pengamatan *Throughput* pada skenario 1 didapat hasil seperti pada Tabel 4.4 dan ditampilkan pada Gambar 4.4.

4.2 Skenario 2 (Jarak pengujian 15 meter dengan umur tanaman Jamur Tiram yang berbeda)

1. Packet Loss Ratio

Tabel 4. 5 Hasil Packet Loss Ratio Skenario 2

lo	naman	entase (%)
.	mbuh	
.	an Tumbuh	
.	g Panen	



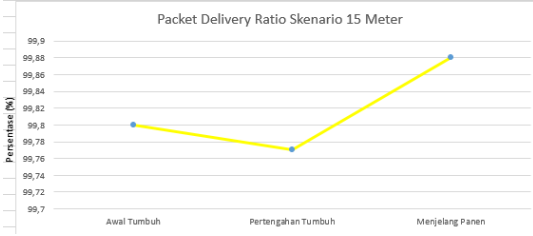
Gambar 4. 5 Grafik Packet Loss Ratio Skenario 2

Dari hasil pengamatan *Packet Loss Ratio* pada skenario 2 didapat hasil seperti pada Tabel 4.5 dan ditampilkan pada Gambar 4.5 didapat dengan standarisasi *TIPHON* sangat baik.

2. Packet Delivery Ratio

Tabel 4. 6 Hasil *Packet Delivery Ratio* Skenario 2

No	Umur Tanaman	Persentase (%)
1.	Awal Tumbuh	99,80
2.	Pertengahan Tumbuh	99,77
3.	Menjelang Panen	99,88



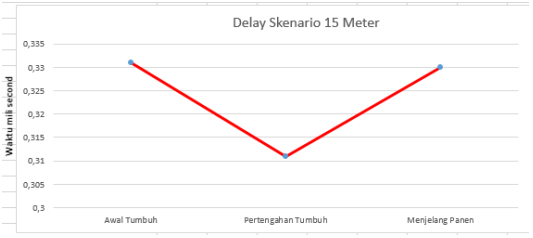
Gambar 4. 6 Hasil *Packet Delivery Ratio* Skenario 2

Dari hasil pengamatan *Packet Delivery Ratio* pada skenario 2 didapat hasil seperti pada Tabel 4.6 dan ditampilkan pada Gambar 4.6 didapat dengan standarisasi *TIPHON* sangat baik.

3. Delay

Tabel 4. 7 Hasil Delay Skenario 2

No	Umur Tanaman	*mili second*
1.	Awal Tumbuh	0,331
2.	Pertengahan Tumbuh	0,311
3.	Menjelang Panen	0,330



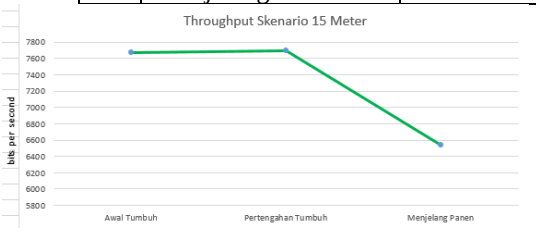
Gambar 4. 7 Hasil *Delay* Skenario 2

Dari hasil pengamatan *Delay* pada skenario 2 didapat hasil seperti pada Tabel 4.7 dan ditampilkan pada Gambar 4.7 didapat dengan standarisasi *TIPHON* sangat baik.

4. Throughput

Tabel 4. 8 Hasil Throughput Skenario 2

No	Umur Tanaman	*bits per second*
1.	Awal Tumbuh	7666
2.	Pertengahan Tumbuh	7694
3.	Menjelang Panen	6543



Gambar 4. 8 Hasil *Throughput* Skenario 2

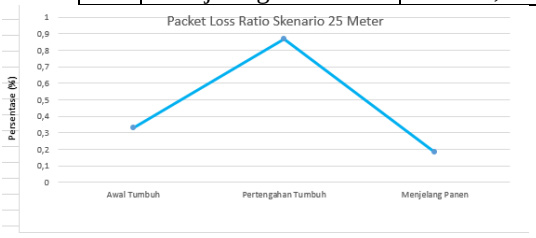
Dari hasil pengamatan *Throughput* pada skenario 2 didapat hasil seperti pada Tabel 4.8 dan ditampilkan pada Gambar 4.8.

4.3 Skenario 3 (Jarak pengujian 25 meter dengan umur tanaman

1. Packet Loss Ratio

Tabel 4. 9Hasil Packet Loss Ratio Skenario 3

No	Umur Tanaman	Persentase (%)
1.	Awal Tumbuh	0,33
2.	Pertengahan Tumbuh	0,87
3.	Menjelang Panen	0,18



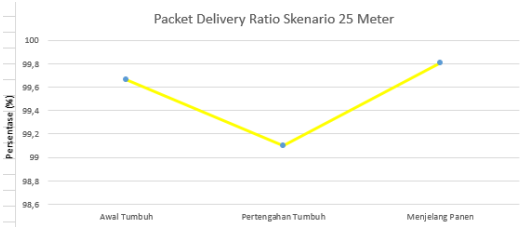
Gambar 4. 9 Grafik Packet Loss Ratio Skenario 3

Dari hasil pengamatan *Packet Loss Ratio* pada skenario 3 didapat hasil seperti pada Tabel 4.9 dan ditampilkan pada Gambar 4.9 didapat dengan standarisasi *TIPHON* sangat baik.

2. Packet Delivery Ratio

Tabel 4. 10 Hasil *Packet Delivery Ratio* Skenario 3

No	Umur Tanaman	Persentase (%)
1.	Awal Tumbuh	99,66
2.	Pertengahan Tumbuh	99,10
3.	Menjelang Panen	99,81



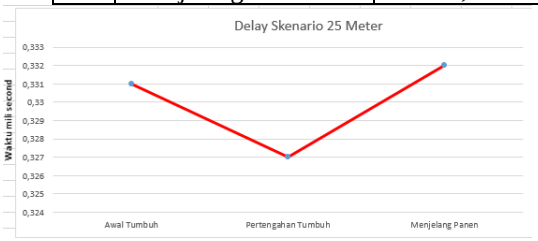
Gambar 4. 10 Hasil *Packet Delivery Ratio* Skenario 3

Dari hasil pengamatan *Packet Delivery Ratio* pada skenario 3 didapat hasil seperti pada Tabel 4.10 dan ditampilkan pada Gambar 4.10 didapat dengan standarisasi *TIPHON* sangat baik.

3. Delay

Tabel 4. 11 Hasil Delay Skenario 3

No	Umur Tanaman	*mili second*
1.	Awal Tumbuh	0,331
2.	Pertengahan Tumbuh	0,327
3.	Menjelang Panen	0,332



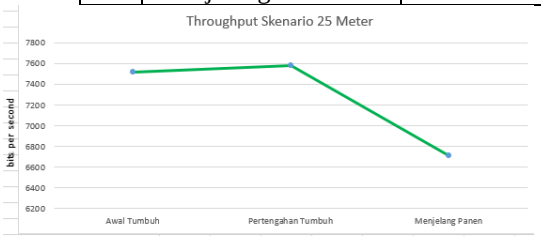
Gambar 4. 11 Hasil *Delay* Skenario 3

Dari hasil pengamatan *Delay* pada skenario 3 didapat hasil seperti pada Tabel 4.11 dan ditampilkan pada Gambar 4.11 didapat dengan standarisasi *TIPHON* sangat baik.

4. Throughput

Tabel 4. 12 Hasil Throughput Skenario 3

No	Umur Tanaman	*bits per second*
1.	Awal Tumbuh	7512
2.	Pertengahan Tumbuh	7580
3.	Menjelang Panen	6710



Gambar 4.12 Hasil *Throughput* Skenario 3

Dari hasil pengamatan *Throughput* pada skenario 3 didapat hasil seperti pada Tabel 4.12 dan ditampilkan pada Gambar 4.12.

V. KESIMPULAN

Berdasarkan hasil penelitian performa protokol MQTT pada tanaman Jamur Tiram dengan menerapkan jaringan *Internet Of Thing* dapat diambil kesimpulan :

1. *Packet Loss Delivery* yang dihasilkan dari proses pengambilan data juga dipengaruhi oleh jarak dan halangan dari *subscriber* ke *client*. Kemudian dari 3 skenario *Packet Loss Delivery* yang diperoleh hasil dibawah 1%, sehingga menurut standarisasi *TIPHON* memiliki hasil sangat bagus.
2. Packet Delivery Ratio yang dihasilkan dari proses pengambilan data juga dipengaruhi oleh jarak dan halangan dari *subscriber* ke *client*. Pada 3 skenario dihasilkan *Packet Delivery Ratio* yang bagus, sehingga keberhasilan yang diterima penerima memiliki hasil yang bagus.
3. Delay yang dihasilkan dari proses pengambilan data memiliki rata – rata yang bagus, sehingga menurut standarisasi TIPHON sangat bagus.
4. Throughput yang dihasilkan dari proses pengambilan data memiliki rata-rata yang hamper sama
5. Penggunaan Internet Of Things dalam tanaman Jamur Tiram sangat membantu petani dalam memantau suhu, kelembaban, dan relay berfungsi sebagai saklar otomatis untuk menghidupkan pompa, kipas dan lampu. Dapat diprogram sesui dengan keadaan iklim sekitar budidaya Jamur Tiram

DAFTAR PUSTAKA

[1] Y. Kang, K.-S. H. Mi-Ran Han and J.-B. Kim, "A Study on Internet of Things (IoT) Applications," 2015.

[2] S. Rahmat and Nurhidayat, Untung Besar dari Bisnis Jamur Tiram, Jakarta Selatan: AgroMedia Pustaka, 2011.

[3] Sciforce, "Smart Farming: The Future of Agriculture," https://www.iotforall.com/smart-farming-future-of-agriculture/, 2019.

[4] R. H. Abdur, "Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol Mqtt Pada Smarthome," http://repository.ub.ac.id/81/, 2017.

[5] M. M. Shekh, A. S.R, Hariprakash and Harshitha, "IoT Based Home Automation using Node MCU," International Journal of Engineering Science and Computing, 2018.

PEMANTAUAN *ROUTER* CPE PADA JARINGAN *METRO ETHERNET* MENGGUNAKAN ZABBIX BERBASIS RASPBERRY PI

Aris Hartono, Unan Yusmaniar Oktiawati
PT. Multipolar Technology Tbk, Jakarta
Departemen Teknik Elektro dan Informatika,
Sekolah Vokasi
Universitas Gadjah Mada, Yogyakarta
aris.hartono@multipolar.com, unan_yusmaniar@ugm.ac.id

Abstract – The development of information and communication technology, especially in this digital era, demands communication takes place quickly and the management of network connectivity efficiently and effectively. In line with this, PT Indonesia Comnets Plus have built the Metro Ethernet network which is an ethernet network technology implemented in a metropolitan area (big cities). The company provides an internet network due to providing the best quality network is compulsory. Hence, we need a Metro Ethernet network monitoring system with the aim of being able to know the performance and problems efficiently and in real time. One of the open source based applications used for network monitoring is Zabbix. This study aims to create and test the performance of Raspberry Pi in the Metro Ethernet network monitoring system at PT Indonesia Comnets Plus with the case study of DISKOMINFO in Garut Regency using the Zabbix application. In monitoring, the SNMP (Simple Network Monitoring Protocol) protocol is needed which will send network problems in the form of triggers. Trigger will evaluate the data collected and represent the current state of the system. Trigger contains two statuses, "OK" and "PROBLEM", to determine the threshold for what is "acceptable" data. Therefore, if the incoming data exceeds an acceptable state, the trigger status changes to "PROBLEM", the trigger status is recalculated every time the Zabbix server receives a new value. Zabbix server will send trigger status to electronic mail (e-mail) and Telegram application as notification for network administrators.

Keywords : Metro Ethernet, Zabbix, SNMP, Raspberry Pi, Telegram.

Intisari – Perkembangan teknologi informasi dan komunikasi terutama di era digital ini menuntut komunikasi berlangsung cepat serta pengelolaan konektivitas jaringan secara efisien dan efektif. Sejalan dengan hal tersebut, PT Indonesia Comnets Plus membangun jaringan *Metro Ethernet* yang merupakan teknologi jaringan *ethernet* yang diimplementasikan di sebuah area metropolitan (kota – kota besar). Perusahaan menyediakan jaringan internet, memberikan kualitas terbaik jaringan adalah suatu kewajiban. Maka diperlukan suatu sistem pemantauan jaringan *Metro Ethernet* dengan tujuan dapat mengetahui performa serta masalah secara efisien dan *real time*. Salah satu aplikasi berbasis *open source* yang digunakan untuk pemantauan jaringan adalah Zabbix. Penelitian ini bertujuan membuat dan menguji performa Raspberry Pi dalam sistem pemantauan jaringan *Metro Ethernet* pada PT Indonesia Comnets Plus dengan studi kasus DISKOMINFO Kabupaten Garut menggunakan aplikasi Zabbix. Dalam melakukan pemantauan dibutuhkan protokol SNMP (*Simple Network Monitoring Protocol*) yang akan mengirimkan permasalahan jaringan berupa *trigger*. *Trigger* akan mengevaluasi data yang dikumpulkan dan mewakili kondisi sistem saat ini. *Trigger* berisi dua status yaitu "OK" dan "*PROBLEM*", untuk menentukan ambang batas kondisi data apa yang "dapat diterima". Oleh karena itu, jika data yang masuk melampaui keadaan yang dapat diterima, *trigger* berubah status menjadi "*PROBLEM*", status *trigger* dihitung ulang setiap kali *server* Zabbix menerima nilai baru. *Server* Zabbix akan mengirimkan status *trigger* ke surat elektronik (surel) dan aplikasi Telegram sebagai notifikasi bagi administrator jaringan.

Kata kunci : Metro Ethernet, Zabbix, SNMP, Raspberry Pi, Telegram

I. PENDAHULUAN

Internet adalah kumpulan atau jaringan dari jaringan komputer yang ada di seluruh dunia. Dalam hal ini komputer yang sebelumnya *stand alone* kini dapat berhubungan langsung dengan *host-host* atau komputer lainnya (Janner Simarmata, 2006: 281). Pada zaman sekarang ini, pemanfaatan jaringan internet sudah menjadi salah satu kebutuhan masyarakat dari berbagai kalangan, mulai dari perusahaan perusahaan, kantor, universitas, sekolah, rumah tangga dan lain sebagainya. Oleh sebab itu dengan kebutuhan tersebut, maka perusahaan penyedia jaringan internet atau yang biasa disebut dengan *Internet Service Provider* (ISP).

PT Indonesia Comnets Plus (ICON+) merupakan perusahaan yang bergerak dalam layanan jasa yang berbasis teknologi jaringan internet dengan salah satu layanannya yaitu *Metro Ethernet*. Bagi suatu perusahaan yang bergerak di bidang penyedia jaringan internet, memberikan kualitas layanan jaringan yang terbaik adalah sebuah kewajiban. Maka dari itu, diperlukan *Network Monitoring System* yang digunakan untuk melakukan pemantauan perangkat jaringan berbasis *Metro Ethernet* di sisi *last mile*. Perusahaan dengan banyaknya sumber daya jaringan di berbagai wilayah yang tersebar di hampir seluruh provinsi di Indonesia membuat *monitoring* perangkat jaringan tidak perlu menggunakan sumber daya komputer apabila hanya beberapa perangkat jaringan saja yang di*monitoring*. Apabila *monitoring* dilakukan oleh komputer, dimensi yang besar menjadikan komputer sebagai *server* dinilai tidak praktis. Maka biaya yang dikeluarkan akan menjadi besar. Dengan spesifikasi komputer yang ada saat semakin canggih maka biaya yang diperlukan untuk mebangun *server monitoring* akan semakin besar. Untuk memenuhi kebutuhan tersebut, maka pada penelitian ini akan membahas sistem *monitoring* jaringan di DISKOMINFO Kabupaten Garut pada perangkat jaringan berbasis *Metro Ethernet* di sisi *last mile* sehingga dapat dipantau secara *real time* dengan menggunakan Raspberry Pi oleh administrator jaringan di PT Indonesia Comnets Plus (ICON+).

Network Monitoring System (NMS) merupakan sebuah sub sistem dalam manajemen jaringan (*Network Management System*) yang melibatkan penggunaan perangkat lunak dan perangkat keras.[1] Konsep manajemen jaringan adalah tentang adanya manager atau perangkat yang bertugas untuk melakukan manajemen dan *agent* atau *host* atau perangkat yang dimanajemen. Sistem dirancang dengan mikrokomputer Raspberry Pi sebagai *server* aplikasi Zabbix

yang terhubung ke jaringan perusahaan. Raspbian adalah sistem operasi gratis berbasiskan Debian yang dioptimalkan untuk perangkat keras Raspberry Pi. Mulai dari Zabbix 4.0 dan Raspbian Stretch, Zabbix LLC menyediakan repository resmi untuk pemasangan langsung Zabbix.[2]

Aplikasi Zabbix sebagai sebuah aplikasi yang dibangun berbasis web akan memberikan kelebihan dan kemudahan dalam menampilkan informasi jaringan. Tampilan berupa *Graphical User Interface* (GUI) yang dapat memudahkan administrator jaringan membaca kondisi jaringan dari nilai yang diberikan oleh SNMP. Permasalahan yang terjadi pada perangkat jaringan diinformasikan oleh *trigger*. *Trigger* akan mengevaluasi data yang dikumpulkan dan mewakili kondisi sistem saat ini. *Trigger* berisi dua status yaitu "OK" dan "*PROBLEM*", untuk menentukan ambang batas kondisi data apa yang "dapat diterima". Oleh karena itu, jika data yang masuk melampaui keadaan yang dapat diterima, *trigger* berubah status menjadi "*PROBLEM*", status *trigger* dihitung ulang setiap kali *server* Zabbix menerima nilai baru. *Server* Zabbix akan mengirimkan status *trigger* ke surat elektronik (surel) dan Telegram sebagai notifikasi bagi administrator jaringan.

Pada penelitian ini menggunakan aplikasi Zabbix yang ditanamkan pada perangkat Raspberry Pi dengan *interface* pemantauan berupa *website* dengan bantuan *database*. Pemetaan jaringan (*network mapping*) dan sistem peringatan dini berupa surat elektronik (surel) dan Telegram. Implementasi dan *monitoring* dilakukan dengan memantau perangkat jaringan milik PT Indonesia Comnets Plus (ICON+) berupa *router* CPE pada jaringan *Metro Ethernet*. Pengujian dilakukan untuk mengetahui performa sistem, sistem notifikasi dan fitur sistem manajemen dan pemantauan dengan lima macam model, yaitu: pengujian *fault, configuration, accounting, performance, dan security* (FCAPS).

A. Tinjauan Pustaka

Beberapa penelitian yang telah dilakukan diantaranya, melakukan penelitian tentang Implementasi dan Analisis Sistem *Monitoring* Jaringan *Supervisory Control And Data Acquisition* (SCADA) PLN Disjaya- shelter Gambir dengan menggunakan Aplikasi Zabbix Pada PT Indonesia Comnets Plus. Pada penelitian ini dijabarkan secara detail tentang *monitoring* sistem *Supervisory Control And Data Acquisition* (SCADA) PLN sehingga dibutuhkan suatu aplikasi untuk melakukan *monitoring* jaringan yaitu, Zabbix. Selain mengetahui performa jaringan, diperlukan juga pemantauan terhadap kualitas *Service Level Agreement* (SLA) dan Bandwidth yang digunakan. Dalam melakukan pemantauan SCADA, hanya dibutuhkan satu protokol yaitu *Internet Control Message Protocol* (ICMP) dimana hanya mengirim pesan-pesan kesalahan apabila terjadi kesalahan, protokol ini juga dapat menstabilkan kondisi jaringan untuk menjadikan normal kembali.[3]

Penelitian selanjutnya berjudul Penerapan metode UTAUT untuk memprediksi *behavioral intentions user* dalam menggunakan aplikasi Zabbix**.** Pengujian ini mengadopsi model untuk melihat niat pengguna Aplikasi UTAUT Zabbix di PT Media Nusantara Citra Tbk. Empat konstruk UTAUT digunakan sebagai penentu yang mempengaruhi perilaku niat (niat perilaku pengguna), yaitu harapan kinerja, harapan usaha, sosial mempengaruhi dan memfasilitasi kondisi. Data dalam Penelitian diuji dengan menggunakan SEM (Structural Equation Model). SEM adalah alat analisis statistik digunakan untuk menyelesaikan model penelitian bertingkat serentak. [4]

Penelitian selanjutnya tentang eksplorasi Zabbix untuk *monitoring* perangkat jaringan (Studi Kasus Teknik Informatika Universitas Pasundan). Pada penelitian ini dijelaskan secara detail tentang perancangan dan implementasi protokol *Simple Network Management Protocol* (SNMP) untuk manajemen jaringan yang diimplementasikan langsung di Program Studi Teknik Informatika Universitas Pasundan. Dalam operasionalnya, protokol *Simple Network Management Protocol* (SNMP) ini dibantu dengan aplikasi Zabbix. *Monitoring* jaringan diperlukan untuk mengevaluasi kinerja dan untuk memastikan efisiensi dan stabilitas operasional.[5]

Penelitian selanjutnya tentang *network* dan *service monitoring* menggunakan Nagios dan Zabbix pada Laboratorium Informatika UMM. Penelitian ini dilakukan dengan mengambil studi kasus di perguruan tinggi yang mempunyai segmen luas dalam segi jangkauan dan layanan yang disediakan oleh *server* sangatlah kompleks. Sedangkan pada topologi yang ada pada perguruan tinggi mendukung jaringan lokal pada dalam ruangan dan jaringan *wireless* pada jaringan luar ruangan. Implementasi pada kedua tipe jaringan tersebut pastinya membutuhkan perangkat sebagai alat *monitoring* jaringan pada berbagai macam aktifitas yang dijalankan agar dapat selalu termonitor.[6]

B. Landasan Teori

Internet merupakan koneksi jaringan komputer di seluruh dunia yang menggunakan protokol komunikasi yang sama untuk berkomunikasi satu dengan yang lain yang disebut *Transmission Control Protocol* / *Internet Protocol* (TCP/IP). Jaringan tersebut mentransmisikan sinyal digital dengan menggunakan kabel maupun tanpa kabel (*wireless*). Media transmisi internet berupa kabel dapat berupa kabel fiber optik dan *coaxial*. Media transmisi dengan menggunakan *wireless* bergantung pada spektrum gelombang elektromagnetik (Wallace, 2016).

Metro Ethernet merupakan teknologi jaringan *Ethernet* yang diimplementasikan di sebuah metropolitan area. Perusahaan - perusahaan besar dapat memanfaatkan teknologi tersebut untuk menghubungkan kantor- kantor cabang ke dalam sistem intranet yang ada di dalam perusahaan tersebut. Jaringan *Metro Ethernet* umumnya didefinisikan sebagai *bridge* dari suatu jaringan atau menghubungkan wilayah yang terpisah juga menghubungkan LAN dan WAN atau *backbone network* yang umumnya dimiliki oleh *service provider*.[7]

Monitoring Jaringan adalah suatu proses pengumpulan dan analisis terhadap data - data pada lalu lintas jaringan dengan tujuan memaksimalkan seluruh sumber daya (*resource*) yang dimiliki suatu arsitektur jaringan. *Monitoring* jaringan ini merupakan bagian dari manajemen jaringan. Manajemen jaringan adalah tindakan melakukan pemantauan, pengujian, konfigurasi, dan penyelesaian masalah pada jaringan untuk memenuhi persyaratan yang dibutuhkan dari suatu kelompok atau organisasi (McGraw Hill, 2007).

Raspberry Pi adalah komputer berukuran kartu kredit berbiaya rendah yang dihubungkan ke monitor komputer

atau TV, dan menggunakan *keyboard* dan *mouse* standar. Ini adalah perangkat kecil yang mampu yang memungkinkan orang - orang dari segala usia untuk menjelajahi komputasi, dan belajar bagaimana memprogram dalam bahasa seperti *Scratch* dan *Python*.

[8] Ini mampu melakukan semua yang diharapkan dari komputer desktop, mulai dari menjelajah internet dan memutar video definisi tinggi, hingga membuat *spreadsheet*, pemrosesan kata, dan bermain *game*. Raspberry Pi telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SoC (*System-on a-chip*) ARM yang dikemas dan diintegrasikan diatas PCB. Perangkat ini menggunakan kartu SD untuk *booting* dan penyimpanan jangka panjang (Bambang Yuwono, dkk, 2015).

SNMP atau (*Simple Network Monitoring Protocol*) adalah sebuah protokol jaringan yang didesain oleh *Internet Engineering Task Force* (IETF) bagi pengguna khususnya administrator jaringan untuk memonitor aktivitas jaringan dan mengontrol sebuah komputer atau *server* atau bahkan perangkat jaringan secara sistematis dari jarak jauh (*remotely*). Pengelolaan ini dilakukan dengan cara melakukan polling dan *setting* variabel variabel elemen jaringan yang dikelolanya.[9]

Zabbix adalah perangkat lunak yang memantau berbagai parameter jaringan dan kesehatan serta integritas *server*. Zabbix menggunakan mekanisme pemberitahuan fleksibel yang memungkinkan pengguna untuk mengkonfigurasi peringatan berbasis surat elektronik (surel) untuk hampir semua media. Ini memungkinkan reaksi cepat terhadap masalah *server*. Zabbix menawarkan fitur pelaporan dan visualisasi data yang sangat baik berdasarkan data yang tersimpan. Ini membuat Zabbix ideal untuk perencanaan kapasitas.[10]

Manajemen jaringan merupakan kemampuan untuk memantau, mengontrol, dan memonitor serta merencanakan sumber daya sistem jaringan dalam sebuah jaringan komputer dari sebuah lokasi. The International Organization for Standardization (ISO) dan International Telecommunication Union (ITU) mendefinisikan arsitektur TMN (*Telecommunication Management Network*) pada tahun 1988 dalam draf kerja pertama (N1719) ISO 10040 mengenalkan sebuah model konseptual untuk menjelaskan fungsi manajemen jaringan bernama FCAPS. FCAPS digunakan untuk berbagai sistem manajemen teknologi.[11]

C. *Tujuan Penelitian*

Tujuan dari dilakukannya penelitian ini adalah mengimplementasi Aplikasi Zabbix pada perangkat Raspberry Pi dalam proses pemantauan *router* CPE pada jaringan *Metro Ethernet* studi kasus DISKOMINFO Kabupaten Garut dengan notifikasi gangguan melalui media surat elektronik (surel) dan aplikasi Telegram serta mengetahui performa sistem dengan model sistem manajemen dan pemantauan yang terdiri lima macam standar yaitu *fault, configuration, accounting, performance, dan security* (FCAPS) pada PT Indonesia Comnets Plus (ICON+).

II METODE PENELITIAN

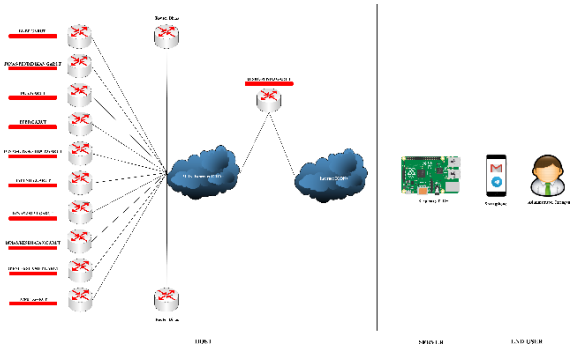
A. *Alat dan Bahan*

Untuk menunjang penelitian ini maka diperlukan alat dan bahan yang dibutuhkan adalah 1 buah *Router* Mikrotik RB1100 (*Router* HO) dan 24 buah *Router* Mikrotik RB2011UiAS-2HnD (*Router* CPE) dengan 1 buah *server* Raspberry Pi 3 model B+ dan 1 unit PC Laptop. Selain itu, dibutuhkan juga suatu *software* penunjang, antara lain: Raspbian GNU/Linux 9 (stretch), Zabbix 4.2, Putty, WinBox, VNC Viewer, Telegram dan *Google Chrome*.

B. *Perancangan Topologi*

Perancangan sistem untuk eksperimen penelitian ini tersaji pada topologi gambar 1. Pada gambar tersebut *router Host* berada dalam jaringan *core* ICON+. Terdapat jaringan *Metro Ethernet* yang menghubungkan *Customer Premise Equipment* (CPE) dengan *Head Office* (HO). HO terhubung ke *server* Raspberry Pi melalui jaringan internet ICON+. Pengambilan data *monitoring* dilakukan pada sisi *Customer Premise Equipment* (CPE). Dalam penelitian ini dibuat rangkaian menggunakan 3 bagian yaitu *Host*, *Server*, dan *End-User*.

Host adalah sebuah perangkat yang digunakan sebagai objek yang akan dipantau oleh *Server* pada jaringan internet ICON+. Sumber data masukan untuk *Host* berasal dari perangkat jaringan yaitu *router*. Perangkat *Host* ini berupa *router* Mikrotik RB2011UiAS-2HnD dan RB1100. *Server* berfungsi sebagai koordinator pusat penerimaan dan pemrosesan data dari *Host*. Data – data yang masuk ke *Server* akan diolah melalui aplikasi pemantauan dengan *cloud storage* berada dalam aplikasi pemantauan didalam *server*. Perangkat *server* yang digunakan adalah Raspberry Pi 3 model B+. Dan *End-user* adalah sebuah perangkat *smartphone* yang digunakan untuk menerima data dari *server* berupa pesan notifikasi yang berisi deskripsi masalah yang terjadi pada *host* maupun *server*.



Gambar 1. Topologi Sistem

C. *Skenario dan Parameter Pengujian*

Dalam pengujian sistem ini, yang akan diuji pertama adalah proses *monitoring* pada jaringan dan *server* menggunakan Zabbix. Pada pengujian proses *monitoring* yaitu pengujian terhadap tampilan antarmuka yang akan menjembatani komunikasi antara *user* dalam hal ini administrator jaringan dengan *host* atau perangkat yang ingin di*monitoring*. Selanjutnya pengujian dilakukan pada proses penyampaian notifikasi melalui surat elektronik (surel) dan aplikasi Telegram saat terjadi masalah pada *server* atau *host*. Dan pengujian performa perangkat *server* Raspberry Pi dalam melakukan *monitoring host* berupa pengukuran CPU *Utilization*, CPU *Load*, *Network Traffic* serta *Memory Usage*.

III. HASIL DAN PEMBAHASAN

A. Hasil Purwarupa Alat

Penelitian ini menghasilkan purwarupa sistem pemantauan perangkat jaringan yang terdiri dari *node server* dan *note host*. Pada gambar 2 adalah alat yang digunakan untuk melakukan *monitoring* perangkat jaringan. *Node Server* menerima data berupa *Data Collection* dari *node host* tentang ketersediaan informasi perangkat kemudian setelah diterima data kemudian di oleh oleh aplikasi Zabbix sebagai aplikasi NMS (*Network Monitoring System*) yang jika terdapat permasalah pada perangkat akan diinformasikan melalui pengiriman pesan masalah ke administrator jaringan melalui media surat elektronik (surel) dan aplikasi Telegram. Alat yang digunakan untuk membuat purwarupa *server* adalah Raspberry Pi 3 model B+. Purwarupa ini menggunakan catu daya listrik sebagai sumber dayanya.

Gambar 2. Tampilan Perangkat Keras *Node Server*

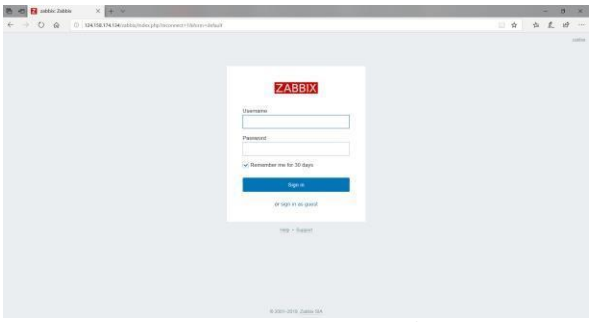
Selain *node server*, pada tugas akhir ini menggunakan perangkat jaringan atau *node host*. *Note host* berfungsi sebagai perangkat yang dipantau yaitu perangkat jaringan milik PT Indonesia Comnets Plus yang ditempatkan di DISKOMINFO Kabupaten Garut. *Node host* secara *real time* dipantau oleh Zabbix yang terdapat pada *node server*. Pada aplikasi Zabbix kondisi *node host* ditampilkan dalam bentuk grafik maupun diagram lingkaran. *Node host* berupa perangkat mikrotik RB2011UiAS-2HnD. Bentuk fisik dari *node host* dapat dilihat pada gambar 3.

Gambar 3. Tampilan Perangkat Kers *Node Host*

B. Tampilan Antarmuka

1) Halaman Login

Halaman *login* zabbix *frontend* diakses dari aplikasi Zabbix seperti pada gambar 4 dengan memasukkan *username* dan *password* yang telah diatur pada Zabbix.



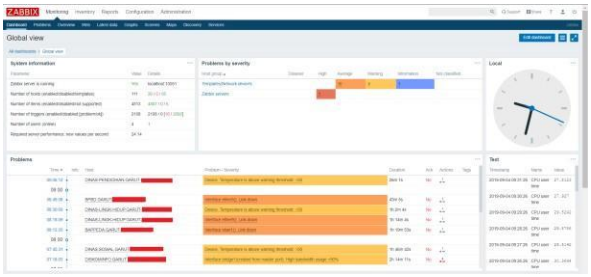
Gambar 4. Halaman *Login* Zabbix *frontend*

Pada halaman ini, terdapat 3 level *user* yang dapat masuk ke sistem, yakni level Zabbix *user*, pengguna memiliki akses ke menu *monitoring*. Pengguna tidak memiliki akses ke sumber daya apapun secara *default*. Izin apa pun untuk meng*host*kan grup harus ditetapkan secara eksplisit. Level Zabbix *Admin*, pengguna memiliki akses ke menu *monitoring* dan *configuration*. Pengguna tidak memiliki akses ke grup *host* apapun secara *default*. Izin apa pun untuk meng*host*kan grup harus diberikan secara eksplisit. Dan level Zabbix Super *Admin*, pengguna memiliki akses ke semuanya: menu *monitoring*, *configuration*, dan *administration*. Pengguna memiliki akses *read* maupun *write* ke semua grup *host*. Izin tidak dapat dicabut dengan menolak akses ke grup *host* tertentu.

Ketika yang akan masuk ke Zabbix adalah seorang administrator maka pada halaman *login* ini administrator harus merupakan seorang *user* yang terdaftar dalam pengguna sistem. Setelahnya, saat *user* sudah terdaftar maka *user* tersebut memiliki *username* dan *password*. Kedua hal tersebut dapat digunakan *user* apabila akan masuk pada Zabbix. Setelah berhasil masuk pada Zabbix, seorang administrator akan memiliki sesi tersendiri sesuai dengan level *user* yang dibuat dan akan diarahkan ke halaman *dashboard* Zabbix.

2) Tampilan Sistem

Setelah *user* dalam hal ini administrator jaringan berhasil masuk maka akan muncul halaman *dashboard* yang terdapat pada menu *monitoring* seperti pada gambar 5.

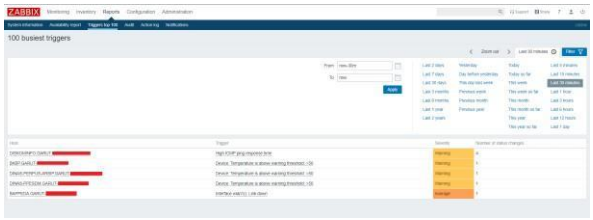


Gambar 5. Halaman *Dashboard* Zabbix *frontend*

Menu *monitoring* merupakan menu pemantauan adalah tentang menampilkan data. Informasi apa pun yang dikonfigurasikan oleh Zabbix untuk dikumpulkan, divisualisasikan, dan ditindaklanjuti, Zabbix akan ditampilkan di berbagai bagian menu pemantauan. Pada menu pemantauan ini data yang ditampilkan seperti *dashboard* yang membuat berbagai informasi, *graph* yang memuat gambaran terkait dengan perangkat, *problem* yang berisikan masalah apa saja yang terjadi serta lainnya. Informasi yang terlihat pada *dashboard* yaitu *system information* yang memuat informasi status *server*, jumlah

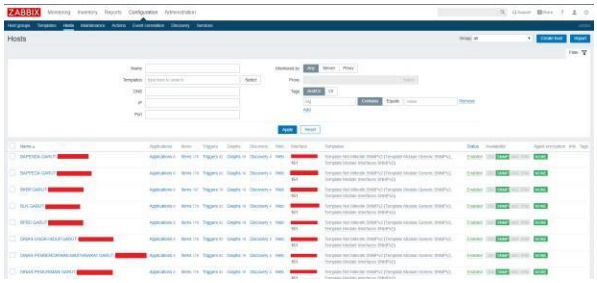
host, jumlah item, jumlah *trigger*, jumlah *user* serta kinerja *server*. Kemudian *problems by severity* yaitu tingkat keparahan masalah pada perangkat maupun *server*. *Problems* memuat histori masalah yang terjadi pada perangkat serta *server*. *Clock* merupakan jam. *Text* berisi keadaan CPU *user time* pada *server*. *Data overview* berisi kondisi *temperature* perangkat dan *Trigger* merupakan pemicu setiap perangkat.

Selanjutnya adalah *inventory*, *inventory* merupakan bagian inventaris menampilkan bagian-bagian yang menyediakan ikhtisar data inventaris *host* dengan parameter yang dipilih serta kemampuan untuk melihat detail inventaris *host*. *Report* merupakan menu laporan untuk menampilkan beberapa bagian yang berisi berbagai laporan yang telah ditentukan sebelumnya dan dapat disesuaikan pengguna yang berfokus pada tampilan ikhtisar parameter seperti status Zabbix, pemicu, dan data yang dikumpulkan seperti pada gambar 6 merupakan laporan dari *triggers* dengan statusnya.



Gambar 6. Halaman *Report* pada sub menu *Triggers Top* 100

Kemudian menu *configuration* merupakan menu konfigurasi berisi bagian untuk mengatur fungsi Zabbix utama, seperti *host* dan grup *host*, pengumpulan data, ambang batas data, mengirimkan pemberitahuan masalah, membuat visualisasi data, dan lainnya.



Gambar 7. Halaman Configuration pada sub menu *Host* 1-8

Pada gambar 7 merupakan halaman *configuration* pada sub menu *host*. Pada gambar 7 *host* yang berhasil ditambahkan berjumlah 8 yaitu BAPENDA Garut, BAPPENDA Garut, BKPB Garut, BPBD Garut, Dinas Lingkungan Hidup Garut, BLK Garut, Dinas Pemberdayaan Masyarakat Garut, Dinas Pemukiman Garut. Selanjutnya pada gambar 8 menampilkan *host* yang berhasil ditambahkan berjumlah 17 *host*.

Gambar 8. Halaman *Configuration* pada sub menu *Host* 9-25

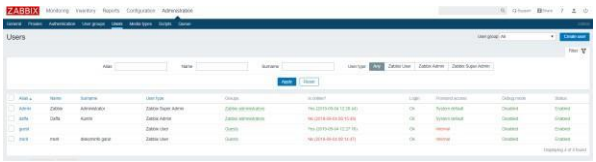
Host yang berhasil ditambahkan Dinas Pendidikan Garut, Dinas Perhubungan Garut, Dinas Perpus Arsip Garut, Dinas Pertanian Garut, Dinas PPESDM Garut, Dinas Ketahanan Pangan Garut, Dinas Koperasi & UKM Garut, Dinas Pemadam Kebakaran, Dinas Perikanan Peternakan Garut, Dinas Sosial Garut, DISDUKCAPIL Garut, DISKOMINFO Garut, DPMPT Garut, DPRSu dr. Slamet Garut, dan Inspektorat Garut. Total *host* yang berhasil ditambahkan berjumlah 25 *host* dengan 1 Zabbix *Server* seperti pada gambar 9 untuk Zabbix *Server*.

Gambar 9. Halaman *Configuration* pada sub menu *Host* Zabbix *Server*

Data yang ditampilkan pada setiap *host* pada sub menu *host* yaitu:

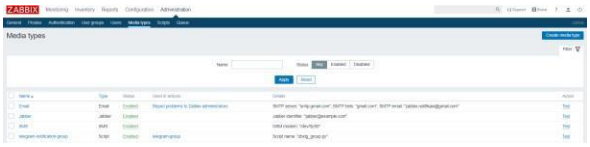
a. *Name* yaitu nama *host* jika mengklik pada nama *host* membuka formulir konfigurasi *host*.
b. *Elements (Applications, Items, Triggers, Graphs, Discovery, Web)* yaitu elemen yang menyertai *host* untuk melihat maka mengklik pada nama elemen akan menampilkan item, memicu dll dari *host*. Jumlah elemen masing-masing ditampilkan dalam warna abu-abu.
c. *Interface* yaitu Alamat dari SNMP perangkat dengan port 161.
d. *Templates* yaitu *Template* yang ditautkan ke *host* ditampilkan untuk melihat maka mengklik pada nama templat akan membuka formulir konfigurasinya.
e. *Status* yaitu status *host* yang ditampilkan dengan 2 kategori diaktifkan atau dinonaktifkan. Dengan mengklik pada status dapat mengubahnya.
f. *Availability* yaitu ketersediaan *host* yang ditampilkan. Masing-masing empat ikon mewakili antarmuka yang didukung (Zabbix agent, SNMP, IPMI, JMX). Status antarmuka saat ini ditampilkan oleh masing-masing warna: hijau yaitu tersedia, merah yaitu tidak, abu-abu - tidak diketahui atau tidak dikonfigurasi.
g. *Agent encryption* yaitu status enkripsi untuk koneksi ke *host* ditampilkan.
h. *Info* yaitu informasi kesalahan (jika ada) mengenai *host* ditampilkan.

Dan terakhir yaitu *administration* merupakan menu administrasi untuk fungsi administratif Zabbix.



Gambar 10. Halaman *Administrator* pada sub menu *Users*

Pada gambar 10 merupakan *user* yang ada di dalam Zabbix. *User Admin* merupakan *user* yang bertipe Zabbix Super *User*, untuk daffa merupakan *user* yang bertipe Zabbix Admin, dan msrit untuk *user* yang bersifat Zabbix *user*. Pada informasi *user* ditampilkan *user* yang sedang *online* maupun yang terakhir *online*. Selanjutnya pada sub menu *media types* ditampilkan media yang aktif digunakan untuk mengirimkan notifikasi ke surat elektronik (surel) maupun aplikasi Telegram seperti pada gambar 11.



Gambar 11. Halaman *Administrator* pada sub menu *Users*

C. *Pengujian Notifikasi*

Permasalahan yang ada pada suatu *host* dapat diketahui oleh administrator jaringan yang bertanggung jawab tanpa harus terus menerus memantau melalui Zabbix *Frontend*, maka diperlukan suatu proses pengiriman notifikasi melalui surat elektronik (surel) maupun aplikasi Telegram yang berisi detail *host* serta permasalahan yang terjadi serta tampilan grafik permasalah.

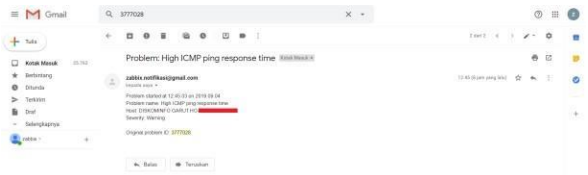
1) *Notifikasi Melalui Surat Elektronik (surel)*

Informasi permasalahan pada *host* dikirimkan melalui surat elektronik (surel) yang sudah didaftarkan di *media type*. Tipe media dengan nama *Email* dengan memasukan alamat surat elektronik (surel) zabbix.notifikasi@gmail.com pada kolom SMTP *email*. Pada bagian *user profile* untuk super *user* pada bagian media dengan pengiriman 24 jam selama 7 hari untuk semua kategori permasalahan.

Kasus permasalahan (*Problem*) *High ICMP Ping Response Time* yang terjadi pada *host* DISKOMINFO Garut pada Zabbix terbaca mulai terjadi masalah pada pukul 12:45:03 pada hari Selasa, 04 September 2019. Pengiriman notifikasi permasalahan pada email diterima pada pukul 12:45:08 pada hari Selasa, 04 September 2019. Selisih waktu pesan permasalahan yang terjadi dengan pengiriman yaitu 5 detik dengan status pengiriman *sent* seperti pada gambar 12.

Gambar 12. *Action log Problem Host* untuk Pengiriman *Email*

Selanjutnya pada kotak masuk surat elektronik, pesan permasalahan (*Problem*) *telah* masuk pada pukul 12:45 dengan pesan permasalah sama dengan *action log* pada Zabbix *frontend* seperti pada gambar 13.

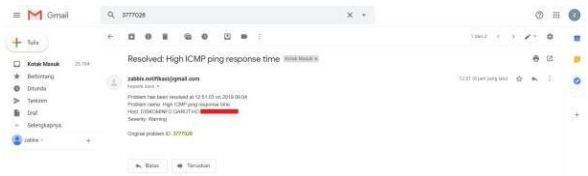


Gambar 13. Pesan *Problem* pada *Host* di Email

Selanjutnya pengiriman notifikasi terselesaikan masalah (*Resolved*) diterima pada pukul 12:51:03 pada hari Selasa, 04 September 2019. Kemudian dilakukan pengiriman pesan pada pukul 12:51:10. Selisih waktu pesan terselesaikannya masalah (*Resolved*) dengan pengiriman yaitu 7 detik dengan status pengiriman *sent* seperti pada gambar 14.

Gambar 14. *Action log Resolved Host* untuk Pengiriman Email

Selanjutnya pada kotak masuk surat elektronik, pesan terselesaikannya masalah (*Resolved*) telah masuk pada pukul 12:51 dengan pesan terselesaikannya masalah sama dengan *action log* pada Zabbix *frontend* seperti pada gambar 15.



Gambar 15. Pesan *Resolved* pada *Host* di Email.

2) *Notifikasi Melalui Aplikasi Telegram*

Informasi permasalahan pada *host* dikirimkan melalui *Bot* Telegram dengan *Token Bot* yang sudah didaftarkan di *media type*. Tipe media dengan nama *telegram-notification-group* dengan memanggil *script* python yang terdapat di kolom *script name* dengan pengiriman data dimasukan pada grup telegram dengan nama Zabbix MSRTI. Pada bagian *user profile* untuk super *user* pada bagian media dengan pengiriman 24 jam selama 7 hari untuk semua kategori permasalahan.

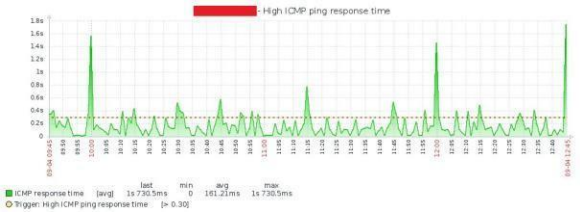
Kasus permasalahan (*Problem*) *High ICMP Ping Response Time* yang terjadi pada *host* DISKOMINFO Garut pada Zabbix terbaca mulai terjadi masalah pada pukul 12:45:03 pada hari Selasa, 04 September 2019. Pengiriman notifikasi permasalahan pada Telegram diterima pada pukul 12:45:08 pada hari Selasa, 04 September 2019. Selisih waktu pesan permasalahan yang terjadi dengan pengiriman yaitu 5 detik dengan status pengiriman *sent* seperti pada gambar 16.

Gambar 16. *Action log Problem Host* untuk Pengiriman Telegram

Selanjutnya pada kotak masuk grup Telegram, pesan permasalahan (*Problem*) *telah* masuk pada pukul 12:45 dengan pesan permasalah sama dengan *action log* pada Zabbix *frontend* seperti pada gambar 17.

Gambar 17. Pesan *Problem* pada *Host* di Grup Telegram

Pengiriman notifikasi pada grup Telegram disertai dengan gambar grafik terjadinya permasalahan (*Problem*). Pada gambar grafik terdapat informasi nilai *response time* dalam periode 3 jam. Pada permasalahan (*Problem*) Diskominfo Garut, nilai *problem* sebesar 1s 730,5ms dengan batas *trigger* yaitu >0.30 seperti pada gambar 18.

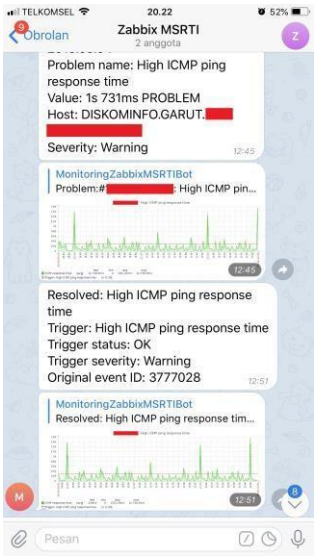


Gambar 18. Gambar Grafik *Problem* pada *Host* di Grup Telegram

Selanjutnya pengiriman notifikasi terselesaikan masalah (*Resolved*) diterima pada pukul 12:51:03 pada hari Selasa, 04 September 2019. Kemudian dilakukan pengiriman pesan pada pukul 12:51:10. Selisih waktu pesan terselesaikannya masalah (*Resolved*) dengan pengiriman yaitu 7 detik dengan status pengiriman *sent* seperti pada gambar 19.

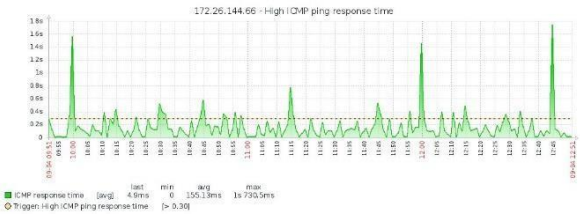
Gambar 19 *Action log Resolved Host* untuk Pengiriman Grup Telegram

Selanjutnya pada kotak masuk grup Telegram, pesan terselesaikannya masalah (*Resolved*) telah masuk pada pukul 12:51 dengan pesan terselesaikannya masalah sama dengan *action log* pada Zabbix *frontend* seperti pada gambar 20.



Gambar 20 Pesan *Resolved* pada *Host* di Grup Telegram

Pengiriman notifikasi pada grup Telegram disertai dengan gambar grafik terselesaikannya masalah (*Resolved*). Pada gambar grafik garis yang semula 1s 730,5ms menjadi turun maka *problem* menjadi terselesaikan seperti pada gambar 21.



Gambar 21 Gambar Grafik *Resolved* pada *Host* di Grup Telegram

D. Analisa Data

Pada analisis data, akan dibahas mengenai data-data yang telah berhasil direkam ke dalam *database*. Data yang dianalisis merupakan data yang masuk pada rentang waktu 21 Juli 2019 sampai dengan 10 September 2019. Berikut merupakan analisis data performa dari pemakaian CPU, *Memory*, dan trafik pada protokol ethernet maupun *wireless* yang berhasil di*monitoring* oleh sistem yang terbagi menjadi 5 skenario yaitu:

a. Skenario Pertama

Pada skenario pertama, data diambil pada kurun waktu 7 hari dimulai dari Kamis, 21 Juni 2019 sampai dengan Rabu, 27 Juni 2019. Perangkat berjumlah 8 yang di*monitoring* yaitu 7 *router* mikrotik diantaranya sebagai *host* yaitu BLK Garut, DISKOMINFO Garut, BKBP Garut, BAPENDA Garut, BPBD Garut, DINAS PENDIDIKAN Garut, dan DINAS LINGKUNGAN HIDUP Garut serta satu *server* ZABBIX terhubung ke jaringan *metro ethernet* dan internet ICON+. Dari jaringan internet ICON+, untuk melakukan proses *monitoring* pada *server* Zabbix di Raspberry Pi dihubungkan dengan jaringan nirkabel melalui protokol jaringan *wireless* (wifi) 2.4GHz 802.11n.

b. Skenario Kedua

Pada skenario kedua, data diambil pada kurun waktu 7 hari dimulai dari Kamis, 04 Juli 2019 sampai dengan Rabu, 10 Juli 2019. Perangkat berjumlah 8 yang di*monitoring* yaitu 7 *router* mikrotik diantaranya sebagai *host* yaitu BLK Garut, DISKOMINFO Garut, BKBP Garut, BAPENDA Garut, BPBD Garut, DINAS PENDIDIKAN Garut, dan DINAS LINGKUNGAN HIDUP, DINAS KESEHATAN Garut, DPMPT Garut, DINAS PUPR Garut, DPRSu dr Slamet Garut serta satu *server* ZABBIX terhubung ke jaringan *metro ethernet* dan internet ICON+. Dari jaringan internet ICON+, untuk melakukan proses *monitoring* pada *server* Zabbix di Raspberry Pi dihubungkan dengan jaringan nirkabel melalui protokol jaringan *wireless* (wifi) 2.4GHz 802.11n.

c. Skenario Ketiga

Pada skenario ketiga, data diambil pada kurun waktu 7 hari dimulai dari hari Minggu, 21 Juli 2019 sampai dengan hari Sabtu, 27 Juli 2019. Perangkat berjumlah 12 yang di*monitoring* yaitu 11 *router* mikrotik diantaranya sebagai *host* yaitu BLK Garut, DISKOMINFO Garut, BKBP Garut, BAPENDA Garut, BPBD Garut, DINAS PENDIDIKAN Garut, DINAS LINGKUNGAN HIDUP Garut, DINAS KESEHATAN Garut, DPMPT Garut, DINAS PUPR Garut, dan DPRSu dr Slamet Garut serta satu *server* ZABBIX terhubung ke jaringan *metro ethernet* dan internet ICON+. Dari jaringan internet ICON+, untuk melakukan proses *monitoring* pada *server* Zabbix di Raspberry Pi dihubungkan dengan jaringan nirkabel melalui protokol jaringan 10/100 *Ethernet*.

d. Skenario Keempat

Pada skenario keempat, data diambil pada hari Selasa, 03 September 2019 dengan dilakukan penambahan *host* setiap 15 menit sekali dengan melakukan pengukuran rata – rata parameter yang akan diambil selama 15 menit. Pengambilan data dilakukan pada *server* dengan 0 *host*, 5 *host*, 10 *host*, 15 *host*, 20 *host*, dan 25 *host*.

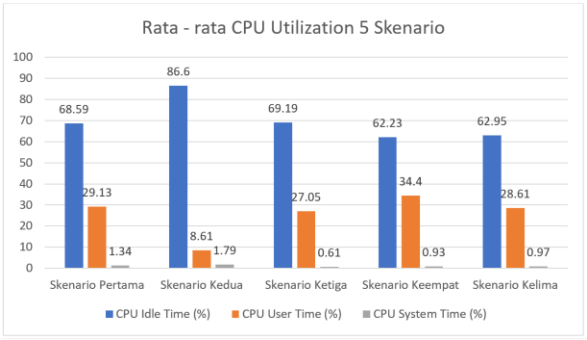
e. Skenario Kelima

Pada skenario kelima, data diambil pada kurun waktu 7 hari dimulai dari hari Rabu, 04 September 2019 sampai dengan hari Selasa, 10 September 2019. Perangkat berjumlah 26 yang di*monitoring* yaitu 25 *router* mikrotik diantaranya

sebagai *host* yaitu BAPENDA Garut, BKBP Garut, BAPENDA Garut, BPBD Garut, BLK Garut, DINAS LINGKUNGAN HIDUP Garut, DISKOMINFO Garut, DINAS PEMBERDAYAAN MASYARAKAT Garut, DINAS PEMUKIMAN Garut, DINAS PENDIDIKAN Garut, DINAS PERHUBUNGAN Garut, DINAS PERPUS ARSIP Garut, DINAS PERTANIAN Garut, DINAS PPESDM Garut, DINAS PUPR Garut, DINAS KESEHATAN Garut, DINAS KETAHANAN PANGAN Garut, DINAS KOPERASI & UKM Garut, DINAS PEMADAM KEBAKARAN, DINAS PERIKANAN PETERNAKAN Garut, DINAS SOSIAL Garut, DISKOMINFO Garut, DPMPT Garut, DPRSu dr Slamet Garut dan INSPEKTORAT Garut serta satu *server* ZABBIX terhubung ke jaringan *metro ethernet* dan internet ICON+. Dari jaringan internet ICON+, untuk melakukan proses *monitoring* pada *server* Zabbix di Raspberry Pi dihubungkan dengan jaringan nirkabel melalui protokol jaringan 10/100 *Ethernet*.

1) CPU Utilization

Pengambilan nilai CPU Utilization bertujuan untuk mengetahui atau persentase waktu prosesor sibuk yang dikategorikan menjadi 3 yaitu CPU Idle Time adalah waktu untuk pengerjaan I/O, CPU User Time adalah waktu eksekusi aplikasi untuk CPU dan CPU System Time adalah waktu kerja sistem operasi atau OS. Adapun hasil dari pengambilan nilai CPU Utilization adalah sebagai berikut:

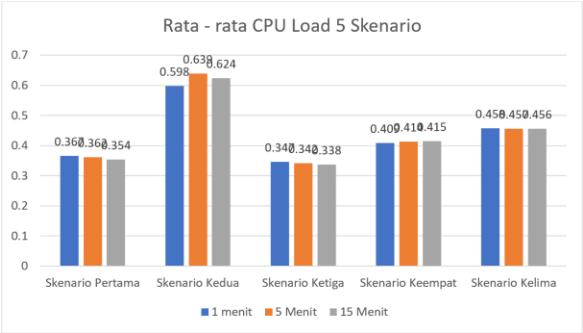


Gambar 22 Grafik Rata – rata CPU *Utilization* untuk Masing – masing Skenario

Pada gambar 22 merupakan grafik rata – rata CPU *Utilization* untuk setiap skenario. CPU *Idle Time*, nilai tertinggi sebesar 86.60 % pada skenario kedua sedangkan nilai terendah sebesar 62.23 % untuk skenario keempat. CPU *User Time*, nilai tertinggi pada skenario keempat sebesar 34.40 % untuk nilai terendah pada skenario kedua sebesar 8.61 %. Dan CPU *System Time*, nilai tertinggi pada skenario kedua sebesar 1.79% untuk nilai terendah pada skenario ketiga sebesar 0.61%.

2) CPU Load

Pengambilan nilai CPU *Load* bertujuan untuk mengetahui beban dari jumlah proses yang sedang dieksekusi oleh CPU. Nilai pada CPU *Load* adalah jumlah rata-rata proses yang sedang atau menunggu dieksekusi selama 1, 5 dan 15 menit terakhir. Nilai pada beban CPU direpresentasikan 1.0 mewakili 100 % untuk setiap *core*nya maka pada *server* ini terdapat 4 *core* nilainya menjadi 4.0. Untuk Zabbix batas CPU *Load* pada *server* < 5 sehingga kalau > 5 maka akan muncul notifikasi beban CPU telah melewati batas normal. Adapun hasil dari pengambilan nilai CPU *Load* adalah sebagai berikut:

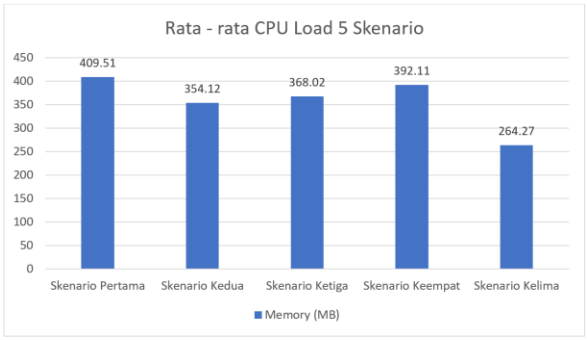


Gambar 23 Grafik Rata – rata CPU *Load* untuk Masing – masing Skenario

Pada gambar 23 merupakan grafik rata – rata CPU *Load* untuk setiap skenario. Rata – rata 1 menit, nilai tertinggi sebesar 0.598 pada skenario kedua sedangkan nilai terendah sebesar 0.347 untuk skenario ketiga. Rata – rata 5 menit, nilai tertinggi pada skenario kedua sebesar 0.639 untuk nilai terendah pada skenario ketiga sebesar 0.342. Dan rata rata 15 menit, nilai tertinggi pada skenario kedua sebesar 0.624, untuk nilai terendah pada skenario ketiga sebesar 0.338.

3) Memory Usage

Pengambilan nilai *Memory Usage* bertujuan untuk mengetahui jumlah memori yang tersedia pada sistem, serta jumlah memori yang digunakan oleh untuk menjalankan aplikasi Zabbix pada *server*. Untuk Zabbix, batas *Memory Usage* pada *server* adalah sisa ruang memori > 20 atau 2.1 % dari jumlah memori total yaitu sebesar 927.2 MB sehingga kalau < 20 maka akan muncul notifikasi ketersediaan memori telah melewati batas normal. Adapun hasil dari pengambilan nilai *Memory Usage* adalah sebagai berikut:



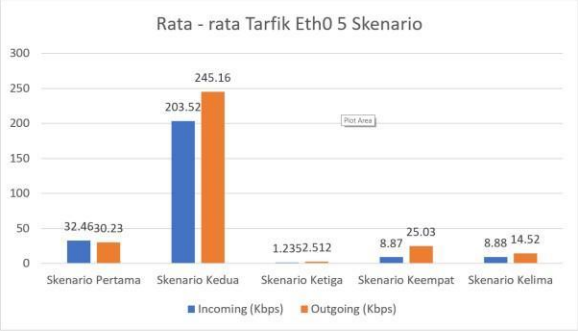
Gambar 24 Grafik Rata – rata CPU *Load* untuk Masing – masing skenario

Pada gambar 24 merupakan grafik rata – rata memori untuk setiap skenario. Rata – rata penggunaan memori terendah adalah pada skenario kelima yaitu sebesar 264.27 MB, sedangkan untuk penggunaan memori tertinggi yaitu pada skenario pertama sebesar 409.51 MB.

4) Network Traffic on eth0

Pengambilan nilai *Traffic* pada *ethernet* 0 pada *server* bertujuan untuk mengetahui jumlah data yang bergerak melintasi jaringan pada titik waktu tertentu. Pada pemantulan Zabbix, trafik dibagi menjadi dua yaitu *incoming* dan *outgoing*. *Incoming* yaitu jumlah data yang ditransfer *server* dari *host* atau dari luar menuju *server*. Sedangkan *outgoing* yaitu jumlah yang dikirim dari *server*.

Total trafik adalah *incoming* + *outgoing*. Adapun hasil dari pengambilan nilai *traffic* adalah sebagai berikut:

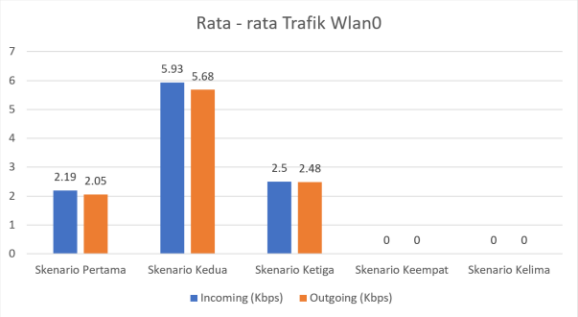


Gambar 25 Grafik Rata – rata Trafik Eth0 untuk Masing – masing Skenario

Pada gambar 25 merupakan grafik rata – rata trafik untuk setiap skenario. Rata – rata penggunaan trafik terendah adalah pada skenario ketiga yaitu sebesar 1.235 Kbps untuk *incoming* dan 2.512 Kbps untuk *outgoing* dengan total rata – rata trafik sebesar 3.747 Kbps sedangkan untuk penggunaan trafik terbesar yaitu pada skenario kedua sebesar 203.52 Kbps untuk *incoming* dan 245.16 Kbps untuk *outgoing* dengan total rata – rata trafik sebesar 448.68 Kbps.

5) Network Traffic on wlan0

Pengambilan nilai *Traffic* pada *wlan0* pada *server* bertujuan untuk mengetahui jumlah data yang bergerak melintasi jaringan pada titik waktu tertentu. Pada pemantulan Zabbix, trafik dibagi menjadi dua yaitu *incoming* dan *outgoing*. *Incoming* yaitu jumlah data yang ditransfer *server* dari *host* atau dari luar menuju *server*. Sedangkan *outgoing* yaitu jumlah yang dikirim dari *server*. Total trafik adalah *incoming* + *outgoing*. Adapun hasil dari pengambilan nilai *traffic* adalah sebagai berikut:



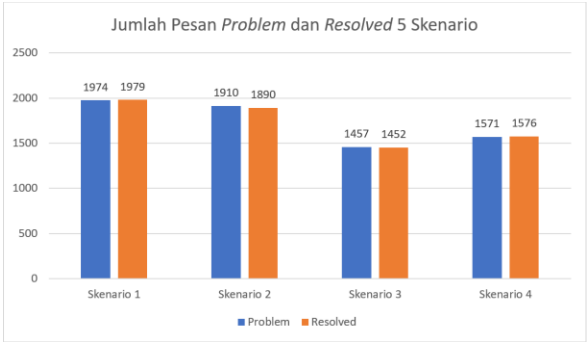
Gambar 26 Grafik Rata – rata Trafik Wlan0 untuk Masing – masing Skenario

Pada gambar 26 merupakan grafik rata – rata trafik untuk setiap skenario. Trafik wlan0 pada skenario 4 dan skenario 5 sebesar 0 Kbps lagi karena semua proses *monitoring* dilakukan melalui *port ethernet*. Rata – rata penggunaan trafik terendah adalah pada skenario pertama yaitu sebesar 2.19 Kbps untuk *incoming* dan 2.05 Kbps untuk *outgoing* dengan total rata – rata trafik sebesar 4.24 Kbps sedangkan untuk penggunaan memori tertinggi yaitu pada skenario kedua sebesar 5.93 Kbps untuk *incoming* dan 5.68 Kbps untuk *outgoing* dengan total rata – rata trafik sebesar 11.61 Kbps.

6) Problems and Resolved

Problems adalah pemicu yang berada dalam status "masalah" atau merupakan suatu kondisi perangkat jaringan dan *server* mengalami permasalah sehingga membuat Zabbix menangkap permasalahan tersebut dengan mengkategorikan ke dalam level – level yang berbeda sesuai standar Zabbix. Sedangkan *resolved* adalah masalah yang terselesaikan. Berbagai permasalahan maupun penyelesaian masalah baik pada perangkat jaringan maupun *server* dikirimkan kepada administrator jaringan melalui surat elektronik (surel) dan Telegram.

Pada saat pesan *problem* maka tandanya perangkat jaringan atau *server* sesuai isi pesan sedang terjadi masalah. Sebaliknya jika pesan *resolved* maka tandanya masalah pada perangkat jaringan atau *server* telah terselesaikan. Pada Bahasa ini akan dilihat jumlah pesan *problem* dan *resolved* sesuai dengan setiap skenario guna melihat persentase penyelesaian permasalahan.



Gambar 27 Grafik Jumlah Pesan *Problem* dan *Resolved* Masing – masing Skenario

Gambar 27 merupakan jumlah *problem* dan *resolved* setiap skenario. Jumlah pesan *problem* yang terjadi dalam 4 skenario yaitu sebanyak 6912 dan pesan *resolved* sebanyak 6888. Maka tingkat penyelesaian masalah 4 skenario sebesar 99.65 % dapat terselesaikan. Rata – rata pesan *problem* sebanyak 246.86 dan rata – rata pesan *resolved* sebanyak 246.

V. KESIMPULAN

Pengimplementasian Aplikasi Zabbix dapat mengetahui kondisi dan status perangkat yang dipantau secara *real time*, mulai dari kinerja setiap perangkat, lalu lintas data, penggunaan CPU, penggunaan memori, suhu perangkat dan permasalahan yang terjadi pada perangkat jaringan maupun *server*.

Notifikasi melalui surat elektronik (surel) dan aplikasi Telegram memudahkan administrator jaringan untuk mengetahui permasalahan pada perangkat jaringan maupun *server* secara cepat tanpa harus mengakses Zabbix *Frontend* terus menerus serta menjadi informasi guna mengambil langkah penyelesaian masalah perangkat jaringan dan *server*.

Koneksi antara Aplikasi Zabbix dan aplikasi Telegram menggunakan BOT Telegram yang sudah di inisialisasikan pada 2 script pada sistem *alert scripts* berbasis python.

DAFTAR PUSTAKA

[1] J. Nofriandi, 2014. *Instalasi Cacti sebagai Network Monitoring System (NMS) pada Linux CentOS 6.3*. [Online] (Updated 25 Feb 2014)
Tersedia di : https://acenk90.wordpress.com [Accessed 06 April 2019]

[2] Zabbix, 2018. *Zabbix on the Raspberry Pi (OS Raspbian)*. [Online] (Updated 01 Des 2018)
Tersedia di : https://zabbix.org [Accessed 06 April 2019]

[3] Indriana S M B, Ayu. 2018. *Implementasi Dan Analisis Sistem Monitoring Jaringan Supervisory Control And Data Acquisition (SCADA) PLN Disjaya-shelter Gambir Dengan Menggunakan Aplikasi Zabbix Pada PT Indonesia Comnets Plus*. Yogyakarta : Universitas Gadjah Mada.

[4] Mediyanto, Beni, dan Irfan Mahendra. 2017. *Penerapan Metode Utaut Untuk Memprediksi Behavioral Intentions User Dalam Menggunakan Aplikasi Zabbix*. Depok: STMIK Nusa Mandiri.

[5] Aji Suparman, Iskandar 2016. *Eksplorasi Zabbix Untuk Monitoring Perangkat Jaringan (Studi Kasus Teknik Informatika Universitas Pasundan)*. Bandung : Universitas Pasundan.

[6] Prayuda, Eka. 2015. *Network dan Service Monitoring Menggunakan Nagios Dan Zabbix Pada Laboratorium Informatika UMM*. Malang : Universitas Muhammadiyah Malang.

[7] Tri Wibisono, Septiaji dan Wahyul Amien Syafei. 2013. *Layanan Jaringan Metronet di PT ICON +*. [Online](Updated 21 Mei 2012) Tersedia di : www.elektro.undip.ac.id [Accessed 07 April 2019]

[8] Raspberrypi, 2018. *What is a Raspberry Pi?*. [Online] Tersedia di : www.raspberrypi.org [Accessed 07 April 2019]

[9] Cisco Networking Academy, 2008. *Chapter 8 : Monitoring The Network*. USA : Cisco.

[10] Zabbix, 2018. *Zabbix Documentation 4.4*. [Online] Tersedia di : www.zabbix.com [Accessed 06 April 2019].

[11] International Organization for Standardization and International Electrotechnical Commission, 1998. *ISO/IEC 10040 Information technology - Open Systems Interconnection*