

Volume 1, No. 1 2020

JISE

Journal of Internet and Software Engineering

<https://ugm.id/jise>

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

- 1. THE INFLUENCE OF FAILOVER ON SOFTWARE-DEFINED NETWORK AND CONVENTIONAL NETWORK** 1-9
Riffat Hasan Saputra, Alif Subardono
- 2. RECEIVED SIGNAL LEVEL (RSL) STRENGTH ASSESSMENT OF 2.4 GHZ IN MEETING ROOM** 10-16
Amar Fauzi, Muhammad Arrofiq
- 3. DATA TRAFFIC ANALYSIS ON AN ELECTRIC CURRENT MONITORING SYSTEM** 17-22
Mohammad Rizki Romadhon, Muhammad Arrofiq
- 4. A PERFORMANCE COMPARISON OF XBEE S2 MODULE AND ESP8266 (12E) MODULE ON AIR QUALITY MONITORING SYSTEM** 23-27
Ricky Hastomo E, Muhammad Arrofiq
- 5. WEB-MONITORING BASED VISUALIZATION OF VIRTUAL LABORATORY USER DATA ON OPENSTACK TELEMETERING COMPONENTS** 28-36
Prajaka Brian Lizadi, Hidayat Nur Isnianto

PENGARUH *FAILOVER* PADA JARINGAN *SOFTWARE-DEFINED NETWORK* DAN KONVENSIONAL

Riffat Hasan Saputra, Alif Subardono
Departemen Teknik Elektro dan Informatika
Universitas Gadjah Mada
riffat.hasan.s@mail.ugm.ac.id, alif@ugm.ac.id

Abstract – *The growing use of the Internet resulted in the number of agencies that use two or more internet gateway paths or so-called Internet Service Providers (ISPs) to maintain network availability. Using two or more gateway paths allows break-up of connections on one of the gateway paths. This study aims to determine the effect of failover on SDN and conventional networks that are integrated with load balancing method of ECMP on MikroTik RB951-2n router device. The controller used on the SDN network is OpenDaylight. Testing is done by doing failover test which then data of bandwidth traffic from both network architecture recorded using Graphing tool from MikroTik. For average traffic efficiency without failover influence on conventional network better with difference of 9.8%. And with the difference in efficiency of 0.5% menunjukkan that the effect of failover on the SDN network is still better than the conventional network.*

Keywords-- *Software-Defined Network, Gateway, Load Balancing, Failover.*

Intisari – Berkembangnya penggunaan internet mengakibatkan banyaknya instansi yang menggunakan dua atau lebih jalur *gateway* internet atau yang biasa disebut *Internet Service Provider (ISP)* untuk menjaga ketersediaan jaringan. Dengan menggunakan dua atau lebih jalur *gateway* memungkinkan terjadinya putus koneksi pada salah satu jalur *gateway*. Penelitian ini bertujuan mengetahui pengaruh *failover* pada jaringan SDN dan konvensional yang diintegrasikan dengan *load balancing* metode ECMP pada perangkat *router* MikroTik RB951-2n. Kontroler yang digunakan pada jaringan SDN adalah *OpenDaylight*. Pengujian dilakukan dengan melakukan uji *failover* yang kemudian data trafik *bandwidth* dari kedua arsitektur jaringan dicatat menggunakan *tools Graphing* dari MikroTik. Untuk rata-rata efisiensi trafik tanpa pengaruh *failover* pada jaringan konvensional lebih baik dengan selisih 9,8%. Dan dengan selisih efisiensi sebesar 0,5% menunjukkan bahwa pengaruh *failover* pada jaringan SDN masih lebih baik dibanding dengan jaringan konvensional.

Kata Kunci—*Software-Defined Network; Load Balancing; ECMP; Failover.*

I. PENDAHULUAN

Perkembangan teknologi layanan internet beberapa tahun terakhir ini mengalami perkembangan yang pesat, khususnya pada jaringan yang memiliki tingkat kompleksitas rendah. Hal ini mengakibatkan meningkatnya layanan akses internet pada jaringan tersebut, yang membuat penggunaan dua atau lebih jalur *gateway* internet sangat diperlukan untuk menjaga ketersediaan jaringan [1]. Sejalan dengan itu, kendala yang harus dihadapi guna menjaga kinerja jaringan agar tetap optimal tentunya juga mengalami peningkatan. Banyaknya permintaan dari pengguna yang menyebabkan *overload* pada salah satu jalur koneksi yang diakibatkan pembagian beban trafik yang tidak berimbang dan putusnya jalur koneksi pada salah satu jalur *gateway* merupakan beberapa masalah yang muncul dari penerapan dua jalur *gateway* atau lebih.

Load balancing yang mempunyai fungsi sebagai pembagi dan penyeimbang beban trafik pada dua atau lebih jalur koneksi, merupakan salah satu solusi yang tepat guna mengatasi kendala tersebut. Terdapat beberapa metode yang tepat untuk menangani masalah pada skala jaringan tersebut, di antaranya adalah ECMP, Nth, dan PCC. Metode tersebut merupakan metode yang paling umum digunakan pada penerapan *load balancing*. Metode ECMP merupakan metode *load balancing* terbaik di antara dua metode lainnya untuk menangani pembagian beban secara

merata pada dua jalur *gateway* dan juga handal terhadap efek *failover*.

Software-Defined Network (SDN) yang merupakan sebuah teknologi yang mempunyai konsep pemisahan antara sistem kontrol (*control plane*) dan sistem *forwarding (forwarding plane)*, memungkinkan jaringan untuk dapat menyederhanakan manajemen dan skalabilitas jaringan yang diwujudkan melalui implementasi sederhana dari penambahan komponen dan layanan jaringan [2]. Dalam penelitian ini kontroler SDN yang digunakan adalah *OpenDaylight Controller*. *Controller* tersebut memiliki keunggulan dibanding *controller* lain yaitu *up-to-date*, mudah digunakan, menggunakan antarmuka Java, menggunakan Maven dan kerangka OSGi [3].

Pengaruh *failover* pada jaringan SDN dan jaringan konvensional yang dihasilkan dari penerapan *load balancing*, memungkinkan memiliki hasil kinerja yang berbeda. Penelitian ini diharapkan menghasilkan keluaran yang dapat mengetahui sejauh mana pengaruh *failover* pada arsitektur konvensional dan arsitektur SDN.

II. TEORI PENDUKUNG

A. Jaringan Komputer

Jaringan Komputer merupakan kumpulan dari beberapa PC (*Personal Computer*) atau peripheral yang saling terhubung melalui media transmisi (melalui kabel atau nirkabel) dan melakukan akses bersama terhadap suatu

resource. Secara lebih sederhana, jaringan komputer dapat diartikan sebagai sekumpulan komputer berserta mekanisme dan prosedurnya yang saling terhubung dan berkomunikasi. Komunikasi yang dilakukan oleh komputer tersebut dapat berupa transfer berbagai data, instruksi, dan informasi dari satu komputer ke komputer yang lain [4].

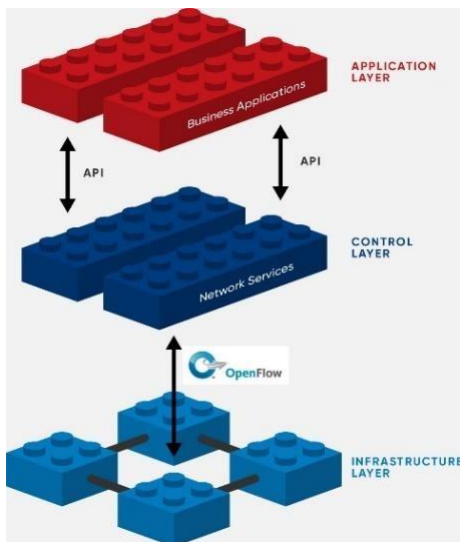
Dibangunnya sebuah sistem jaringan komputer memiliki tujuan agar pengiriman data dan informasi dari pengirim ke penerima dapat dilakukan dengan cepat dan tepat serta meminimalisir kesalahan pada media transmisi atau media komunikasi tertentu [5].

Manfaat dengan dibangunnya sebuah jaringan komputer dapat diperoleh beberapa keuntungan, diantaranya.

1. Berbagi Sumberdaya
2. Media Komunikasi
3. Integrasi Data
4. Keamanan Data
5. Manajemen Sumberdaya
6. Pengembangan dan Pemeliharaan

B. Software-Defined Network

Software-Defined Network adalah sebuah arsitektur yang mana dinamis, mudah diatur, hemat, dan mudah diadaptasi, membuatnya ideal untuk kebutuhan *bandwidth* tinggi yang merupakan sifat dinamis dari aplikasi saat ini. Arsitektur ini memisahkan *control plane* dan *forwarding plane* yang memungkinkan kontrol jaringan menjadi terprogram langsung dan infrastruktur yang mendasarinya untuk diabstraksikan untuk aplikasi dan layanan jaringan. Protokol *OpenFlow* adalah elemen dasar untuk membangun solusi SDN [6].



Gambar 1 Arsitektur *Software-Defined Network*

(Sumber Gambar : <https://www.opennetworking.org/>)

Terlihat pada Gambar 2.1, menunjukkan bahwa SDN memiliki tiga lapis/bidang:

a. Lapisan Aplikasi

Lapisan yang berada paling atas, memiliki fungsi untuk menyediakan *interface* untuk pembuatan program

aplikasi yang mengatur *network requirement* dan *network behavior* sesuai yang diinginkan. Sehingga dapat mengoptimalkan jaringan secara baik dan fleksibel.

b. Lapisan Kontrol

Entitas kontrol (*SDN Controller*) yang mentranslasikan kebutuhan aplikasi dengan infrastruktur dengan memberikan instruksi yang sesuai untuk *SDN Datapath* serta memberikan informasi yang relevan dan dibutuhkan oleh *SDN Application*.

c. Lapisan Arsitektur

Terdiri dari elemen jaringan yang dapat menerima instruksi dari *Control Plane*. *Interface* antara *Control Plane* dan *Data Plane* disebut *South Bound Interface* (SBI), atau *Control-To-Data-Plane Interface* (CDPI).

C. OpenFlow

OpenFlow (OF) adalah salah satu standar SDN pertama. Semula OF menetapkan protokol komunikasi pada lingkungan SDN yang memungkinkan kontroler SDN untuk dapat langsung berinteraksi dengan *forwarding plane* dari perangkat jaringan seperti *switch* dan *router*, baik secara fisik maupun *virtual* (*hypervisor-based*), menjadikan jaringan dapat beradaptasi pada perubahan kebutuhan bisnis [7]. *Openflow* adalah protokol SDN yang paling banyak digunakan. Hal ini dikarenakan *Openflow* memungkinkan peneliti untuk mengembangkan bahkan merancang protokol yang baru [8].

D. OpenDaylight Controller

OpenDaylight dapat dijalankan diberbagai macam produksi *network environments*. *OpenDaylight* dapat mendukung *framework* kontroler modular, tetapi dapat menyediakan dukungan untuk SDN standar lain dan protokol yang akan datang. *OpenDaylight* mengekspos membuka *northbound APIs*, yang digunakan oleh aplikasi. Aplikasi ini menggunakan kontroler untuk mengumpulkan informasi mengenai jaringan, algoritma yang berjalan untuk melakukan analisis, dan kemudian menggunakan *OpenDaylight* untuk membuat *rules* baru di seluruh jaringan. *OpenDaylight* diimplementasikan hanya di *software*, dan disimpan di dalam *Java Virtual Machine* (JVM). Ini berarti *OpenDaylight* dapat dijalankan pada perangkat keras dan OS yang mendukung Java. Untuk hasil terbaik, *OpenDaylight* disarankan menggunakan *Linux distribution* terbaru dan setidaknya JVM 1.7 [9].

E. Load Balance

Load balance adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi [10].

Dalam *load balance* di MikroTik, hal-hal yang perlu diperhatikan dalam mengatur *load balance* adalah *static route*, *policy route*, *firewall mangle*, dan *firewall src-nat*. *Static route* dan *policy router* mengatur *uplink flow*, yaitu kebijakan *routing* yang akan dilalui paket yang telah

ditandai, sedangkan *firewall mangle* adalah penandaan paket sebelum masuk *routing* [11].

1. Equal-Cost Multi-Path (ECMP)

Equal-Cost Multi-Path dikenal sebagai metode *load balancing* yang paling sederhana dan lebih cocok digunakan pada jaringan dengan tingkat kompleksitas rendah. Pembagian beban dalam metode ini menggunakan pengaturan perbandingan langsung antara beban yang diterima oleh setiap *gateway*. Tujuan penerapan algoritma ini yaitu untuk dimungkinkannya penggunaan lebih dari satu jalur *gateway* digunakan sebagai *exit interface* dan mendapatkan pembagian beban yang merata antar setiap *gateway* dalam membagi *traffic* yang ada. Dalam penggunaannya metode ini juga mendukung penggunaan *routing* protokol baik statis maupun dinamis [12].

ECMP yang merupakan “*persisten per-connection load balancing*” atau “*per-src-dst-address combination load balancing*” hal ini membuat begitu salah satu *gateway unreachable* atau terputus, *check-gateway* akan langsung menggunakan *gateway* yang masih aktif saja, sehingga bisa mendapatkan efek *failover*. Ketika sesuatu memicu *routing table flushing* dan *cache* ECMP terhapus. Koneksi akan dikoneksikan sekali lagi ke *gateway* dan dalam kasus dua *gateway* ada 50% kemungkinan bahwa trafik akan melalui *gateway* lain. *Routing table flushing* dapat dipicu dua hal yaitu perubahan *routing table* untuk protokol *routing* dinamis dan setiap 10 menit *routing table* akan terhapus untuk alasan keamanan. Jadi meskipun tidak ada perubahan *routing table*, koneksi dapat berpindah ke *gateway* lain setiap 10 menit [13].

F. Bandwidth

Bandwidth adalah nilai hitung atau perhitungan konsumsi transfer data telekomunikasi yang dihitung dalam satuan *bit* per detik atau yang biasa disingkat bps yang terjadi antara komputer *server* dan komputer klien dalam waktu tertentu dalam sebuah jaringan komputer. *Bandwidth* sendiri akan dialokasikan ke komputer dalam jaringan dan akan mempengaruhi kecepatan transfer data pada jaringan komputer tersebut sehingga semakin besar *bandwidth* pada jaringan komputer maka semakin cepat pula kecepatan transfer data yang dapat dilakukan oleh klien maupun *server* [1] [1].

G. Failover

Failover adalah teknik untuk membagi beban trafik di jaringan yang terlalu besar dan memindahkannya secara otomatis apabila pada jaringan tersebut mengalami kendala dalam koneksi, dalam hal ini jaringan internet bisa terputus koneksi [14]. Aplikasi teknik *failover* biasanya digabungkan dengan teknik *load balancing*. Penggunaan *failover* ditujukan untuk membangun jaringan komputer yang handal [15].

III. METODE PENELITIAN

A. Alat dan Bahan

Adapun perangkat keras dan perangkat lunak yang dibutuhkan sebagai berikut.

1. Perangkat Keras:
 - 3 PC / Laptop
 - 4 Mikrotik RB951-2n
 - 7 Kabel LAN
2. Perangkat Lunak:
 - Ubuntu Desktop 14.04
 - OpenDaylight Controller
 - Winbox

B. Metode Penelitian

Metode penelitian yang dilakukan penulis dalam penelitian disajikan dalam bentuk *flow chart*. Gambar 2 menunjukkan *flow chart* penelitian.

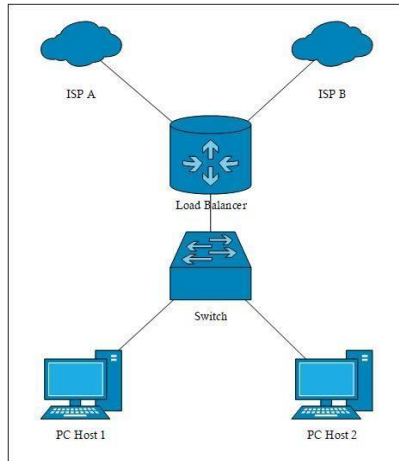


Gambar 2. Flow Cart Penelitian

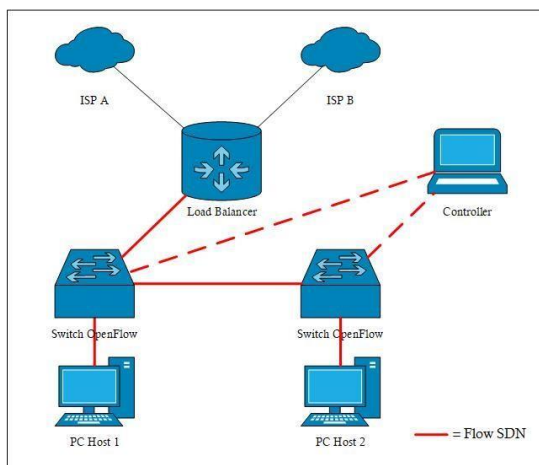
C. Implementasi Sistem Pengujian

1. Perancangan Topologi

Topologi yang digunakan pada Perbandingan dan Analisis Kinerja *Load Balancing* Metode ECMP Pada Jaringan *Software-Defined Network* sebanyak 2 skenario topologi yang semuanya menggunakan topologi *tree*.



Gambar 3. Skenario Topologi Jaringan Konvensional



Gambar 4. Skenario Topologi Jaringan SDN

2. Pengujian Sistem

Pengujian yang dilakukan untuk menganalisis perbandingan pada penelitian ini menggunakan *graphing* yang merupakan *proprietary tools* dari *router* MikroTik. *Graphing* dapat melakukan *monitoring* terhadap beberapa parameter pada *router* seperti trafik, *memory* dan CPU yang disajikan dalam bentuk grafik dan angka. Pada penelitian ini parameter yang digunakan hanya trafik saja. Grafik dari *tools graphing* dapat diakses dengan cara membuka *web browser*, dengan format alamat <http://alamat-IP-router/graphs> atau <http://192.168.1.1/graphs> pada penelitian ini. Dan untuk parameter pengujian yang digunakan yaitu berupa trafik *bandwidth* yang masuk dari *gateway* ISP-A (Internet UGM) dan ISP-B (Internet Telkomsel) ke dalam *router* untuk menguji pembagian beban serta efek *failover*. Dan efisiensi

trafik yang didistribusikan menuju jaringan Lokal yang terdapat perbedaan topologi.

Pengujian dilakukan dengan lima buah kasus yang berbeda pada masing-masing arsitektur jaringan. Host-1 dan Host-2 akan melakukan tugas yang sama pada setiap kasus dan untuk variabel pembedanya adalah *gateway* yang digunakan. Pada setiap kasus Host-1 akan melakukan pengunduhan berkas yang berukuran 1,5 GB dari situs *ubuntu.com* dan untuk Host-2 melakukan *streaming* video dari situs *youtube.com* dengan kualitas 360p.

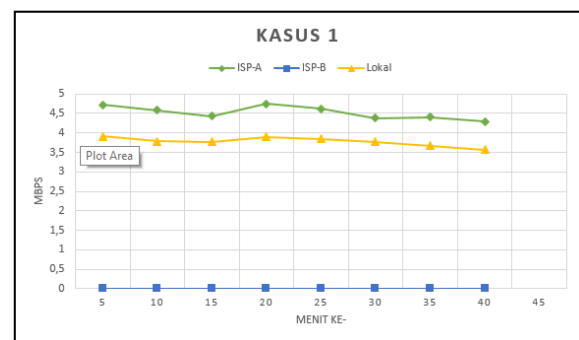
Untuk kasus pertama *gateway* yang digunakan hanya ISP-A dengan alokasi *bandwidth* 4 Mbps. Pada kasus kedua *gateway* yang digunakan hanya ISP-B dengan alokasi *bandwidth* 2 Mbps. Sedangkan pada kasus ketiga menggunakan dua *gateway* secara bersamaan yaitu ISP-A dengan alokasi *bandwidth* 4 Mbps dan ISP-B dengan alokasi *bandwidth* 2 Mbps. Pada kasus keempat alokasi *bandwidth* yang digunakan sama dengan kasus ketiga, hanya saja ditambahkan dilakukan uji *failover* dengan memutus koneksi *gateway* ISP-A dan ISP-B secara bergantian di tengah pengaksesan. Untuk kasus terakhir akan menggunakan dua jalur *gateway* dengan alokasi *bandwidth* yang sama yaitu 2 Mbps serta ditambahkan uji *failover*. Data rata-rata dari trafik *bandwidth* yang diambil dicatat menggunakan *graphing* setiap lima menit terhitung sejak setiap kasus mulai dijalankan selama 40 menit untuk masing-masing kasus.

IV. ANALISIS DAN PEMBAHASAN

Analisis data dilakukan berdasarkan kumpulan data yang telah diambil berdasar parameter pada tiap skenario pengujian yang telah ditentukan. Nilai parameter disajikan dalam bentuk grafik kemudian di analisis dan dilakukan perbandingan pada setiap skenario. Adapun analisis yang dihasilkan adalah sebagai berikut.

A. Pengujian Jaringan Konvensional

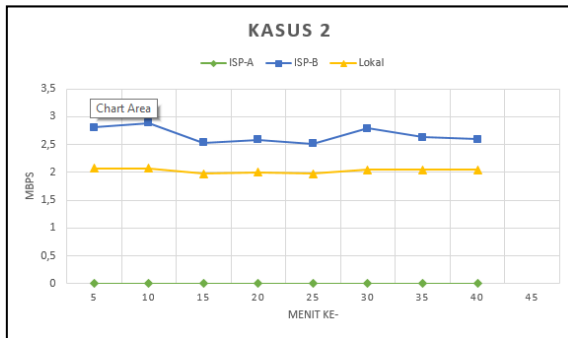
1. Pengujian Kasus 1



Gambar 5. Hasil Pengujian Trafik *Bandwidth* Pada Kasus 1 Topologi Konvensional

Dari Gambar 5 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A mengalami fluktuasi pada kisaran 4,5 Mbps. Dan trafik yang keluar menuju jaringan Lokal dibawah 4 Mbps, hal ini berarti trafik yang benar-benar dapat didistribusikan ke jaringan Lokal tidak dapat penuh 4 Mbps seperti *bandwidth* yang disediakan oleh ISP-A.

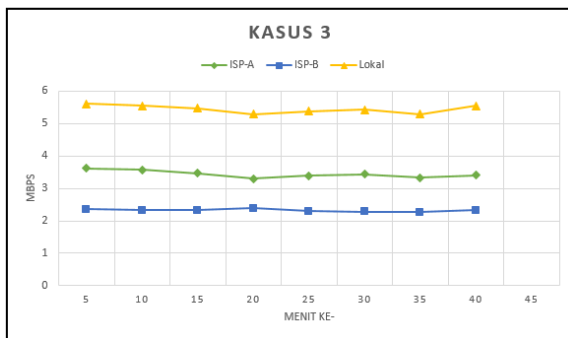
2. Pengujian Kasus 2



Gambar 6. Hasil Pengujian Trafik Bandwidth Pada Kasus 2 Topologi Konvensional

Dari Gambar 6 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-B mengalami fluktuasi pada kisaran 2,5 hingga 3 Mbps. Dan trafik yang keluar menuju jaringan Lokal stabil pada angka 2 Mbps, hal ini berarti trafik yang benar-benar dapat didistribusikan ke jaringan Lokal dapat 2 Mbps seperti *bandwidth* yang disediakan oleh ISP-B.

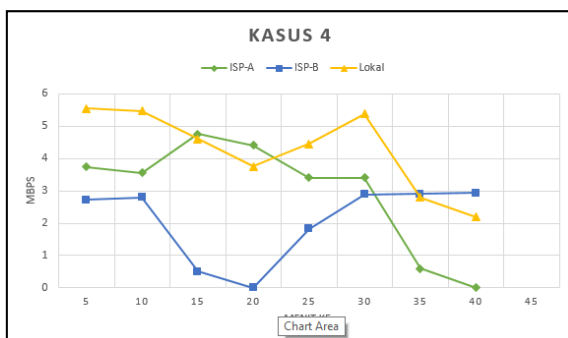
3. Pengujian Kasus 3



Gambar 7. Hasil Pengujian Trafik Bandwidth Pada Kasus 3 Topologi Konvensional

Dari Gambar 7 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A cukup stabil pada 3,5 Mbps dan ISP-B stabil pada 2,3 Mbps. Dan trafik yang keluar menuju jaringan Lokal stabil pada terjadi sedikit fluktuasi pada kisaran 5,5 Mbps, hal ini berarti trafik yang benar-benar dapat didistribusikan ke jaringan Lokal sebesar 5,5 Mbps mendekati *bandwidth* yang digabungkan oleh ISP-A dan ISP-B yaitu 6 Mbps, hal ini dikarenakan penyeimbangan beban trafik yang dilakukan *load balancer*.

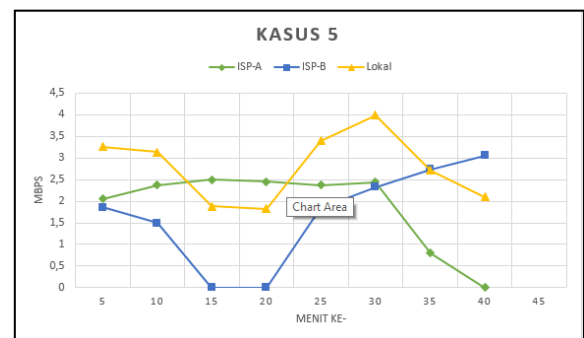
4. Pengujian Kasus 4



Gambar 8. Hasil Pengujian Trafik Bandwidth Pada Kasus 4 Topologi Konvensional

Dari Gambar 8 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A pada 10 menit pertama berkisar pada 3,5 Mbps dan ISP-B stabil pada 2,8 Mbps. Dan trafik yang keluar menuju jaringan Lokal pada 10 menit pertama juga stabil pada kisaran 5,5 Mbps, hal ini berarti seperti yang sudah dijelaskan pada kasus 3. Kemudian pada menit 10 hingga 20 dilakukan pemutusan koneksi pada *gateway* ISP-B sehingga dapat dilihat pada Gambar 4.4 terjadi penurunan drastis pada *bandwidth* ISP-B yang pada menit ke 15 masih terdapat sisa trafik *bandwidth* yang pada menit ke 20 tercatat sudah tidak ada lagi. Kebalikan dari ISP-B, pada koneksi ISP-A terjadi peningkatan secara signifikan dari kisaran 3,5 Mbps menjadi kisaran 4,5 Mbps. Pada *interface* Lokal trafik keluar yang awalnya kisaran 5,5 Mbps menurun pada kisaran 4,5 hingga 3,7 Mbps dikarenakan yang bekerja hanya pada *gateway* ISP-A yang mempunyai alokasi *bandwidth* 4 Mbps seperti pada kasus 1. Selanjutnya pada menit 25 hingga 30 koneksi ISP-B dinyalakan kembali sehingga trafik pada ISP-A kembali seperti 10 menit pertama pada kisaran 3,5 Mbps, pada ISP-B terjadi kenaikan bertahap hingga normal seperti pada 10 menit pertama pada menit ke 30 di kisaran 2,8 Mbps. Berikutnya pada menit 30 dan 35 dilakukan pemutusan koneksi pada ISP-A, sehingga dapat dilihat trafik ISP-A terjadi penurunan secara bertahap dari menit 35 yang masih tercatat sisa koneksi pada kisaran 0,5 Mbps hingga habis pada 5 menit terakhir. Sedangkan trafik pada ISP-B terlihat stabil pada kisaran 2,8 Mbps. Dan untuk trafik yang keluar ke jaringan Lokal terjadi penurunan menyesuaikan besaran trafik yang masuk dari ISP-A dan ISP-B.

5. Pengujian Kasus 5



Gambar 9. Hasil Pengujian Trafik Bandwidth Pada Kasus 5 Topologi Konvensional

Dari Gambar 9 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A pada 10 menit pertama berkisar pada 2 sampai 2,3 Mbps dan ISP-B berkisar pada 1,5 Mbps sampai 2 Mbps. Dan trafik yang keluar menuju jaringan Lokal pada 10 menit pertama stabil pada kisaran 3,2 Mbps. Kemudian mulai menit 10 hingga 20 dilakukan pemutusan koneksi pada *gateway* ISP-B sehingga dapat dilihat pada Gambar 4.5 terjadi penurunan drastis pada *bandwidth* ISP-B yang pada menit ke 15 tidak terdapat trafik. Kebalikan dari ISP-B, pada koneksi ISP-A terjadi peningkatan dari kisaran 2 Mbps menjadi kisaran 2,5 Mbps. Pada *interface* Lokal trafik keluar yang awalnya kisaran 3,2 Mbps menurun pada kisaran 1,8 Mbps dikarenakan

yang bekerja hanya pada *gateway* ISP-A yang mempunyai alokasi *bandwidth* 2 Mbps. Selanjutnya pada menit 20 koneksi ISP-B dinyalakan kembali sehingga trafik pada ISP-A kembali seperti 10 menit pertama pada kisaran 2,3 Mbps, pada ISP-B terjadi kenaikan bertahap di kisaran 1,8 hingga 2,3 Mbps. Berikutnya mulai menit 30 dilakukan pemutusan koneksi pada ISP-A, sehingga dapat dilihat trafik ISP-A terjadi penurunan secara bertahap dari menit 35 yang masih tercatat sisa koneksi pada kisaran 0,8 Mbps hingga habis pada 5 menit terakhir. Sedangkan trafik pada ISP-B terlihat terjadi peningkatan dari kisaran 2,3 Mbps hingga 3 Mbps. Dan untuk trafik yang keluar ke jaringan Lokal terjadi penurunan bertahap menyesuaikan besaran trafik yang masuk dari ISP-A dan ISP-B yaitu dari 3,9 Mbps menjadi 2,1 Mbps.

6. Rangkuman Data Pengujian

Tabel 1. Rangkuman Data Pengujian Jaringan Konvensional (Dalam Mbps)

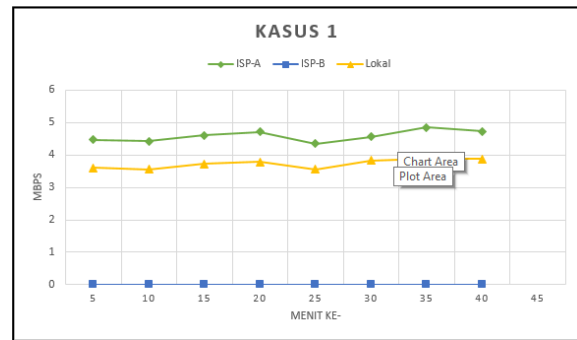
Kasus	ISP-A	ISP-B	Lokal
1	4,52	-	3,77
2	-	2,67	2,03
3	3,44	3,32	5,44
4	2,86	2	4,27
5	1,87	1,66	2,78

Tabel 1 menjelaskan pada kasus pertama terlihat bahwa trafik *bandwidth* yang masuk dari ISP-A yaitu sebesar 4,52 Mbps dan trafik yang keluar menuju jaringan Lokal sebesar 3,77 Mbps. Selanjutnya pada kasus kedua trafik yang masuk dari ISP-B sebesar 2,67 Mbps dan trafik yang keluar sebesar 2,03 Mbps. Kemudian pada kasus ketiga ketika menggunakan dua *gateway* secara bersamaan terlihat penurunan trafik pada masing-masing *gateway* di mana ISP-A menjadi 3,44 Mbps dan ISP-B menjadi 2,3 Mbps, namun trafik yang menuju jaringan Lokal menjadi sebesar 5,42 Mbps. Hal ini disebabkan pembagian beban trafik yang dilakukan oleh *load balancing* ECMP untuk menyeimbangkan beban trafik secara *equal-cost* pada kedua *gateway* dan membuat trafik yang keluar lebih besar sebanding dengan jumlah trafik yang masuk.

Pada kasus keempat, meskipun alokasi *bandwidth* yang digunakan sama dengan kasus ketiga, dapat dilihat terjadi penurunan trafik pada seluruh *interface*, hal ini disebabkan dilakukannya uji *failover* pada pertengahan pengaksesan, yang menunjukkan trafik ISP-A menjadi sebesar 2,86 Mbps, ISP-B sebesar 2 Mbps dan pada *interface* Lokal sebesar 4,27 Mbps. Untuk kasus ke lima menggunakan dua *gateway* secara bersamaan dengan alokasi *bandwidth* sama besar 2 Mbps, rata-rata trafik untuk ISP-A tercatat sebesar 1,87 Mbps dan ISP-B 1,66 Mbps sehingga menghasilkan rata-rata trafik keluar sebesar 2,78 Mbps. Meskipun pada masing-masing *gateway* trafik yang dialokasikan sebesar 2 Mbps, rata-rata trafik yang didapat dibawah 2 Mbps karena dilakukan uji *failover* seperti kasus ke 4.

B. Pengujian Jaringan SDN

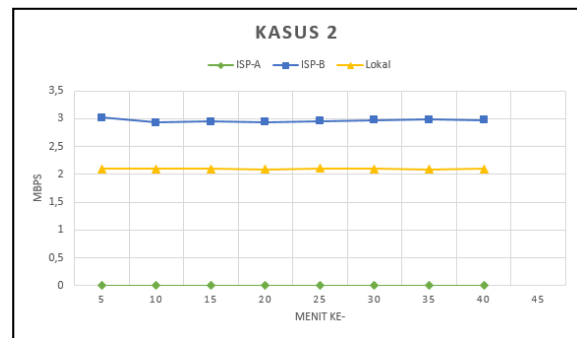
1. Pengujian Kasus 1



Gambar 10. Hasil Pengujian Trafik *Bandwidth* Pada Kasus 1 Topologi SDN

Dari Gambar 10 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A mengalami fluktuasi pada kisaran 4,5 Mbps. Dan trafik yang keluar menuju jaringan Lokal dibawah 4 Mbps, hal ini berarti trafik yang benar-benar dapat didistribusikan ke jaringan Lokal tidak dapat penuh 4 Mbps seperti *bandwidth* yang disediakan oleh ISP-A.

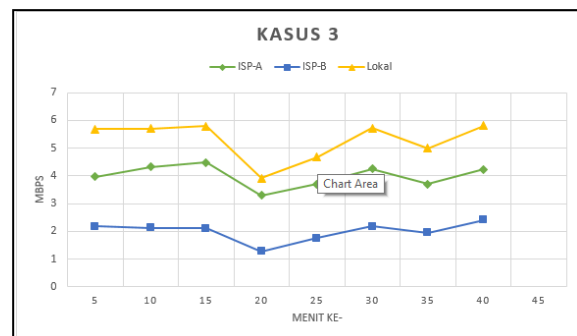
2. Pengujian Kasus 2



Gambar 11. Hasil Pengujian Trafik *Bandwidth* Pada Kasus 2 Topologi SDN

Dari Gambar 11 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-B terlihat stabil pada kisaran 2,9 Mbps. Dan trafik yang keluar menuju jaringan Lokal stabil pada angka 2,1 Mbps, hal ini berarti trafik yang benar-benar dapat didistribusikan ke jaringan Lokal dapat 2 Mbps seperti *bandwidth* yang disediakan oleh ISP-B

3. Pengujian Kasus 3

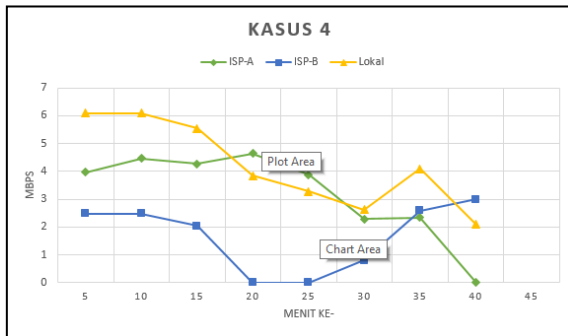


Gambar 12. Hasil Pengujian Trafik *Bandwidth* Pada Kasus 3 Topologi SDN

Dari Gambar 12 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A cukup stabil pada kisaran 4,2

Mbps hanya saja mengalami penurunan pada menit 20 dan menit 35 dikarenakan terjadi penurunan *request* trafik pada *host* yang berpengaruh pada kedua *gateway*. Pada ISP-B juga mengalami juga stabil pada 2,1 Mbps dan sempat mengalami penurunan trafik yang sama terjadi pada ISP-A antara menit 15 sampai 25 dan pada menit 35. Dan trafik yang keluar menuju jaringan Lokal stabil pada kisaran 5,7 Mbps, karena trafik keluar menuju jaringan Lokal menyesuaikan trafik yang masuk maka terjadi penurunan juga antara menit 15 sampai 25 dan pada menit 35

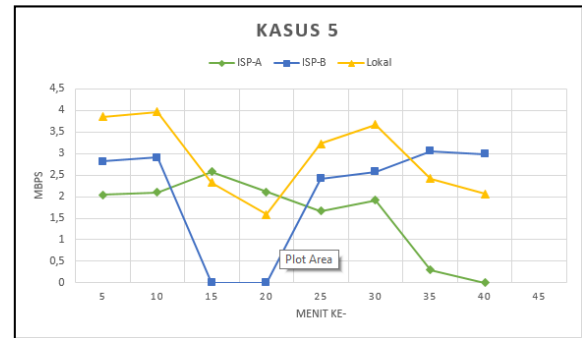
4. Pengujian Kasus 4



Gambar 13. Hasil Pengujian Trafik *Bandwidth* Pada Kasus 4 Topologi SDN

Dari Gambar 13 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A pada 10 menit pertama berkisar antara 4 Mbps sampai 4,3Mbps kemudian mengalami pada ISP-B stabil dikisaran 2,5 Mbps. Dan trafik yang keluar menuju jaringan Lokal pada 10 menit pertama juga stabil pada kisaran 6 Mbps sebanding dengan trafik yang masuk. Kemudian pada menit 15 hingga 25 dilakukan pemutusan koneksi pada *gateway* ISP-B sehingga dapat dilihat pada Gambar 4.8 terjadi penurunan drastis pada *bandwidth* ISP-B pada menit ke 20 hingga trafik menjadi 0 Mbps. Dan pada koneksi ISP-A terjadi peningkatan sedikit dari kisaran 4 Mbps menjadi 4,5 Mbps. Pada *interface* Lokal trafik keluar yang awalnya kisaran 6 Mbps menurun pada kisaran 3,9 Mbps dikarenakan yang bekerja hanya pada *gateway* ISP-A yang mempunyai alokasi *bandwidth* 4 Mbps seperti pada kasus 1. Selanjutnya pada menit 25 hingga 35 koneksi ISP-B dinyalakan kembali sehingga pada ISP-B terjadi kenaikan bertahap hingga normal seperti pada 10 menit pertama pada menit ke 35 di kisaran 2,7 Mbps. Berikutnya pada menit 35 dilakukan pemutusan koneksi pada ISP-A, sehingga dapat dilihat trafik ISP-A terjadi penurunan secara bertahap dari menit 35 hingga habis pada menit 40. Sedangkan trafik pada ISP-B terlihat stabil pada kisaran 2,8 Mbps. Dan untuk trafik yang keluar ke jaringan Lokal terjadi penurunan menyesuaikan besaran trafik yang masuk dari ISP-A dan ISP-B.

5. Pengujian Kasus 5



Gambar 14. Hasil Pengujian Trafik *Bandwidth* Pada Kasus5 Topologi SDN

Dari Gambar 14 dapat dilihat bahwa *bandwidth* yang masuk dari ISP-A pada 10 menit pertama berkisar pada 2 Mbps dan ISP-B berkisar pada 2,8 Mbps. Dan trafik yang keluar menuju jaringan Lokal pada 10 menit pertama stabil pada kisaran 3,8 Mbps. Kemudian mulai menit 10 hingga 20 dilakukan pemutusan koneksi pada *gateway* ISP-B sehingga dapat dilihat pada Gambar 4.10 terjadi penurunan drastis pada *bandwidth* ISP-B yang pada menit ke 15 tidak terdapat trafik. Kebalikan dari ISP-B, pada koneksi ISP-A terjadi peningkatan dari kisaran 2 Mbps menjadi kisaran 2,5 Mbps. Pada *interface* Lokal trafik keluar yang awalnya kisaran 3,8 Mbps menurun pada kisaran 2,3 Mbps hingga 1,5 Mbps dikarenakan yang bekerja hanya pada *gateway* ISP-A yang mempunyai alokasi *bandwidth* 2 Mbps. Selanjutnya pada menit 20 koneksi ISP-B dinyalakan kembali sehingga trafik pada ISP-A turun lagi pada kisaran 2 Mbps, pada ISP-B terjadi kenaikan menjadi 2,5 Mbps. Berikutnya mulai menit 30 dilakukan pemutusan koneksi pada ISP-A, sehingga dapat dilihat trafik ISP-A terjadi penurunan secara bertahap dari menit 35 yang masih tercatat sisa koneksi pada kisaran 0,3 Mbps hingga habis pada 5 menit terakhir. Sedangkan trafik pada ISP-B terlihat terjadi peningkatan hingga 3 Mbps. Dan untuk trafik yang keluar ke jaringan Lokal terjadi penurunan bertahap menyesuaikan besaran trafik yang masuk dari ISP-A dan ISP-B yaitu dari 3,6 Mbps menjadi 2 Mbps

6. Rangkuman Data Pengujian

Tabel 2. Rangkuman Data Pengujian Jaringan SDN (Dalam Mbps)

Kasus	ISP-A	ISP-B	Lokal
1	4,58	-	3,72
2	-	2,96	2,09
3	4,37	2,17	5,74
4	3,14	1,62	4,2
5	1,58	1,72	2,62

Tabel 2 Menjelaskan bahwa pada kasus pertama terlihat bahwa trafik *bandwidth* yang masuk dari ISP-A yaitu sebesar 4,58 Mbps dan trafik yang keluar menuju jaringan Lokal sebesar 3,72 Mbps. Selanjutnya pada kasus kedua trafik yang masuk dari ISP-B sebesar 2,96 Mbps dan trafik yang keluar sebesar 2,09 Mbps. Kemudian pada kasus ketiga ketika menggunakan dua *gateway* secara bersamaan

terlihat penurunan trafik pada masing-masing *gateway* di mana ISP-A menjadi 4,37 Mbps dan ISP-B menjadi 2,17 Mbps, namun trafik yang menuju jaringan Lokal menjadi sebesar 5,74 Mbps.

Pada kasus keempat, meskipun alokasi *bandwidth* yang digunakan sama dengan kasus ketiga, dapat dilihat terjadi penurunan trafik pada seluruh *interface*, hal ini disebabkan dilakukannya uji *failover* pada pertengahan pengaksesan, yang menunjukkan trafik ISP-A menjadi sebesar 3,14 Mbps, ISP-B sebesar 1,62 Mbps dan pada *interface* Lokal sebesar 4,2 Mbps. Ini menunjukkan bahwa beban trafik yang sebelumnya diseimbangkan oleh *load balancing* pada dua *gateway* menjadi hilang sehingga beban trafik kembali membebani salah satu *gateway* yang masih aktif saja. Untuk kasus ke lima menggunakan dua *gateway* secara bersamaan dengan alokasi *bandwidth* sama besar 2 Mbps, rata-rata trafik untuk ISP-A tercatat sebesar 1,58 Mbps dan ISP-B 1,72 Mbps² sehingga menghasilkan rata-rata trafik keluar sebesar 2,62 Mbps. Meskipun pada masing-masing *gateway* trafik yang dialokasikan sebesar 2 Mbps, rata-rata trafik yang didapat dibawah 2 Mbps karena dilakukan uji *failover* seperti kasus ke 4.

C. Perbandingan Data Hasil Pengujian

Tabel 3. Perbandingan Data Hasil Pengujian Paringan Konvensional (K) dan SDN (S) (Dalam Mbps)

Kasus	ISP-A	ISP-B	Lokal	Efisiensi
1 – K	4,52	-	3,77	83,4%
1 – S	4,58	-	3,72	81,2%
2 – K	-	2,67	2,03	76%
2 – S	-	2,96	2,09	70,6%
3 – K	3,44	3,32	5,44	94,4%
3 – S	4,37	2,17	5,74	87,7%
4 – K	2,86	2	4,27	87,8%
4 – S	3,14	1,62	4,2	88,2%
5 – K	1,87	1,66	2,78	78,7%
5 - S	1,58	1,72	2,62	79,3%

Untuk pembandingan pengaruh *failover* dari kedua topologi dapat dilihat efisiensi trafik pada *interface* Lokal yang didistribusikan menuju jaringan lokal, sedangkan untuk *interface* ISP-A dan ISP-B di sini untuk melihat efisiensi data yang masuk dari kedua *gateway* yang kemudian didistribusikan keluar ke *interface* Lokal. Dari Tabel 4.7 dapat dilihat bahwasannya secara keseluruhan efisiensi trafik dari jaringan konvensional lebih baik dari jaringan SDN dan untuk data yang masuk pada SDN secara keseluruhan lebih besar dibanding dengan jaringan konvensional. Merujuk dari Tabel 4.7 pada kasus 4 dan 5, topologi jaringan SDN lebih baik tingkat efisiensi trafiknya, dapat disimpulkan bahwa untuk pengaruh *failover* pada jaringan SDN memiliki kemampuan yang lebih baik.

V. KESIMPULAN

Berdasarkan hasil analisis pengujian, maka dapat diambil kesimpulan bahwa Efisiensi trafik pada jaringan konvensional bila tanpa pengaruh *failover* pada jaringan konvensional menunjukkan angka yang lebih baik dari

jaringan SDN dengan selisih 2,2% untuk kasus 1, 5,4% untuk kasus 2 dan 6,7% untuk kasus 3. Kemudian untuk Efisiensi trafik dengan pengaruh *failover* menunjukkan bahwa jaringan SDN memiliki angka yang lebih baik dari jaringan konvensional dengan selisih 0,4% untuk kasus 4 dan 0,6 % untuk kasus 5. Dan terakhir bahwa Intergrasi *load balancing* metode ECMP baik pada arsitektur topologi konvensional dan arsitektur topologi SDN, kurang cocok untuk jaringan yang membutuhkan *session login*, dikarenakan pada metode ECMP setiap 10 menit melakukan *routing table flushing* seperti yang sudah dijelaskan pada Bab II, sehingga memerlukan *login* ulang atau setidaknya menunggu untuk *refresh login* yang mana akan membuat jaringan kurang stabil.

DAFTAR PUSTAKA

- [1] G. Prakosa, Analisis Perbandingan Metode Load Balancing Ecmp, Nth, Dan Pcc Menggunakan Mikrotik Cloud Hosted Router Pada Gns3, Yogyakarta: Universitas Gadjah Mada, 2016.
- [2] I. Grgurević, Z. Kavran dan A. Pušeljić, Simulation Analysis of Characteristics and Application of Software-Defined Networks, Zagreb: University of Zagreb, 2015.
- [3] R. Kartadi, Uji Performa Kontroler Floodlight Dan OpenDaylight Sebagai Komponen Utama Arsitektur Software-Defined Network, Yogyakarta: STIMIK AMIKOM Yogyakarta, 2015.
- [4] M. Didik, Modul Praktikum Jaringan Komputer, Yogyakarta: UIN Sunan Kalijaga, 2011.
- [5] B. S. D. Oetomo, Konsep dan Perancangan Jaringan Komputer, Yogyakarta: Andi, 2003.
- [6] Open Network Foundation, "Software-Defined Networking (SDN) Definition," 2018. [Online]. Available: <https://www.opennetworking.org/sdn-definition/>. [Diakses 11 Mei 2018].
- [7] SDX-Central, "What is OpenFlow? Definition and How it Relates to SDN," 2012. [Online]. [Diakses 12 Mei 2018].
- [8] W. Rahmaning, Implementasi Dan Analisis Kinerja Protokol Routing BGP (Border Gateway Protocol) Pada Jaringan SDN (Software Defined Network), Yogyakarta: UGM, 2017.
- [9] SDX-Central, "What is an OpenDaylight Controller? AKA: OpenDaylight Platform," 2012. [Online]. Available: <https://www.sdxcentral.com/sdn/definitions/sdn-controllers/openflow-controller/>. [Diakses 13 Mei 2018].
- [10] H. Nugroho dan M. R. Utama, "Simulasi Management Bandwidth dan Load Balancing Server Menggunakan Clear OS pada Virtual Box," *Jurnal ICT Penelitian dan Penerapan Teknologi*, Vol. %1 dari %2-, no. -, p. 82, 2014.
- [11] M. F. Adani, Analisis Perbandingan Metode Load Balance PCC Dengan Nth Menggunakan MikroTIK, 1st penyunt., Surabaya: Institu Bisnis & Informatika STIKOM, 2016.

-
- [12] R. Ramandito, S. Sumaryono dan S. Suning, "Analisis Performance Jaringan Komputer dengan Mekanisme Load Balancing-Failover," *Jurnal Penelitian Teknik Elektro*, vol. 3, p. 178, 2010.
- [13] MikroTik, "ECMP load balancing with masquerade," 2016. [Online]. Available: https://wiki.mikrotik.com/wiki/ECMP_load_balancing_with_masquerade. [Diakses 28 Juli 2018].
- [14] R. Towidjojo, Mikrotik Kung Fu Kitab 2, Jakarta: Jasakom, 2013.
- [15] K. D. T. Saputro, Membangun Server Internet dengan Mikrotik OS, Yogyakarta: Gava Media, 2008.

ASESSMENT KEKUATAN DAYA *RECEIVED SIGNAL LEVEL* (RSL) WIRELESS 2,4 GHZ DI RUANG MEETING

Amar Fauzi, Muhammad Arrofiq

amarfauzi17@gmail.com

Departemen Teknik Elektro dan Informatika
Sekolah Vokasi Universitas Gadjah Mada

Abstarct - *Wireless Local Area Network (WLAN)* is a computer networking technology that uses radio transmission media by utilizing free space as a transmission line. By utilizing free space as transmission line of signal transmitted by distance, frequency, tx power, receiver sensitivity, and antenna gain.

This study aims to analyze the signal quality of the building profile factor at the meeting room, organized by the value of RSL (*Received Signal Level*) as an indicator of signal strength level of each access point of the meeting room, and calculate other parameters such as EIRP (*Effective Isotropic Radiated Power*), free space loss with theoretical and measurement levels. Receive level signals in the field using the Netsurveyor and Vistumbler applications installed on the laptop and the Wifi Overview and Wifi Analyzer app installed on the smartphone.

From the results of both RSL calculations in theory and its changes. Damping factors in the field such as diffraction effect, refraction, reflection, absorption or blocking. The average field summary shows the RSL values in turn "very good" and "good" there is no "low" indicator, the connection in every meeting room is good and normal. Will, found some problems in the field that can cause a decrease in signal quality so that efforts need to optimization.

Keywords: *Wireless Local Network, Received Signal Level, Effective Isotropic Activated Power, free space loss*

Intisari - *Wireless Local Area Network (WLAN)* merupakan teknologi jaringan komputer yang menggunakan media transmisi radio dengan memanfaatkan ruang bebas sebagai jalur transmisi. Dengan pemanfaatan ruang bebas sebagai jalur transmisi maka sinyal yang ditransmisikan dipengaruhi oleh jarak, frekuensi, Tx power, sensitivitas penerima, dan gain antena.

Penelitian ini bertujuan untuk menganalisis kualitas sinyal terhadap faktor profil gedung pada suatu Ruang Meeting, yang ditunjukkan oleh nilai RSL (*Received Signal Level*) sebagai indikator level signal strength setiap access point ruangan meeting Ballroom Hotel, dan menghitung nilai parameter lainnya seperti EIRP (*Effective Isotropic Radiated Power*), free space loss dengan perhitungan teoritis dan pengukuran nilai receive level signal di lapangan menggunakan aplikasi Netsurveyor dan Vistumbler yang terinstal pada laptop serta aplikasi Wifi Overview dan Wifi Analyzer yang terinstal pada smartphone .

Dari hasil kedua perhitungan RSL secara teori dengan pengukuran di lapangan terjadi perbedaan cukup besar, dikarenakan perhitungan teori mengabaikan semua faktor redaman di lapangan seperti pengaruh difraksi, refraksi, refleksi, absorsi maupun blocking. Sedangkan pengukuran di lapangan rata-rata menunjukkan nilai RSL secara indikator "sangat bagus" dan "bagus" tidak terdapat indikator "rendah" , artinya koneksi di setiap ruangan meeting sudah baik dan normal. Akan tetapi, ditemukan beberapa masalah di lapangan yang dapat menyebabkan penurunan kualitas sinyal sehingga perlu upaya untuk mengoptimalkannya.

Kata kunci: *Wireless Local Area Network , Received Signal Level, Effective Isotropic Radiated Power, free space loss*

I. PENDAHULUAN

Jaringan *wireless* merupakan sekumpulan komputer yang saling terhubung antara satu dengan lainnya sehingga terbentuk sebuah jaringan komputer dengan menggunakan media udara/gelombang sebagai jalur lintas datanya. Teknologi ini merupakan perkembangan dari teknologi jaringan lokal (*local area network*) yang memungkinkan efisiensi dalam implementasi dan pengembangan jaringan komputer karena dapat meningkatkan mobilitas *user* dan mengingat keterbatasan dari teknologi jaringan komputer menggunakan media kabel [1]. Jaringan *wireless* sering digunakan oleh perusahaan-perusahaan besar karena proses instalasi jaringan ini relatif lebih cepat dan mudah karena tidak membutuhkan kabel yang harus dipasang sebagai penghubung dan sangat fleksibel terhadap tempat. Salah satunya adalah ruang meeting yang menggunakan jaringan *wireless* sebagai fasilitas layanan internet. Untuk memberikan kemudahan dan kenyamanan bagi para tamu hotel dalam memaksimalkan penggunaan internet yang kadang terjadi sebuah keluhan oleh pengunjung terhadap koneksi internet akibat konektivitas yang lama dan terjadinya kegagalan konektivitas. Oleh karena itu, dalam mengatasi masalah tersebut perlunya sebuah pengamatan terhadap kualitas sinyal *wireless* yang dipancarkan oleh *access point*. Penulis akan mengadakan sebuah penelitian

yang berjudul "Assessment Kekuatan Daya *ReceivedSignal Level* (RSL) *Wireless* 2,4 GHz di Ruang Meeting".

Penelitian ini akan menghitung kekuatan daya *received signal level* tiap-tiap *access point* pada ruang meeting yang dilakukan dengan dua metode, yaitu dengan cara perhitungan dengan rumus dan pengukuran di lapangan. Melalui penelitian ini diharapkan data informasi yang didapat selama penelitian dapat digunakan sebagai referensi dalam mengoptimalkan jaringan *wireless* di ruang meeting

II. TEORI PENDUKUNG

A. Jaringan *Wireless*

Jaringan *wireless* adalah bidang disiplin yang berkaitan dengan komunikasi antar sistem komputer tanpa menggunakan kabel. Jaringan *wireless* sering digunakan untuk jaringan komputer baik pada jarak yang dekat (beberapa meter, memakai alat/pemancar *bluetooth*) maupun pada jarak jauh (menggunakan satelit). Bidang ini erat hubungannya dengan bidang telekomunikasi, teknologi informasi, dan teknik komputer. Jenis jaringan yang populer dalam kategori jaringan nirkabel ini meliputi: Jaringan kawasan lokal nirkabel (*wireless LAN/WLAN*), dan Wi-Fi.

B. Parameter WLAN

Terdapat beberapa parameter yang digunakan untuk menentukan kualitas jaringan WLAN di antaranya :

1. Transmitter Power Level (Tx Power / Daya Pancar)

Semua radio akan mempunyai daya pancar tertentu. Daya pemancar diukur dalam dua satuan, dengan menggunakan watt (atau milliwatt) atau menggunakan satuan dBm. Daya dalam dBm dihitung dengan $\text{dBm} = 10 \times 10^{\log P}$ (daya dalam milliwatt), sehingga pemancar dari 100mW (0.1Watt) adalah setara dengan 20 dBm [2].

2. Penguat Antena (Gain)

Penguatan antena (*gain*) adalah besarnya penguatan antena yang dapat dilakukan oleh antena pada saat memancarkan dan menerima sinyal sebagai keluaran daya pada arah tertentu dibandingkan dengan keluaran yang dihasilkan pada arah sembarang oleh antena *omnidirectional* sempurna (antena *Isotropic*) [2].

3. Sensitivitas Penerima (Minimal Received Signal Level)

Sensitivitas perangkat (*receiver sensitivity*) merupakan kepekaan suatu perangkat pada sisi penerima yang dijadikan ukuran *threshold*. *Receiver sensitivity* menunjukkan besarnya sensitivitas penerima sebagai tolak ukur penerimaan sinyal yang ditransmisikan [3].

C. Effective Isotropic Radiated Power (EIRP)

EIRP adalah total energi yang dikeluarkan oleh sebuah *access point* dan antena. Saat sebuah *access point* mengirim energinya ke antena untuk dipancarkan, pengurangan besar energi akan terjadi di dalam kabel [4]. Untuk mengimbangi hal tersebut, sebuah antena menambahkan *power* / *gain*, dengan demikian *power* bertambah. Jumlah penambahan *power* tersebut bergantung tipe antena yang digunakan. EIRP inilah yang digunakan untuk memperkirakan area layanan sebuah alat *wireless*. Secara matematis dinyatakan seperti pada persamaan (1) [5].

$$\text{EIRP} = \text{PTx} - \text{LTx} + \text{GTx} \quad (1)$$

dengan: PTx= daya pancar antena pemancar LTx= loss kabel di antena pemancar GTx = gain antena pemancar

D. Free Space Loss (FSL)

Redaman ruang bebas atau *free space loss* merupakan penurunan daya gelombang radio selama merambat di ruang bebas. Redaman ini dipengaruhi oleh besar frekuensi dan jarak antara titik pengirim dan penerima dimana pengaruh difraksi, refraksi, refleksi, absorpsi maupun *blocking* dianggap tidak ada. Nilai *free space loss* dihitung dengan persamaan (3) [6].

$$\text{Lfs} = 32,45 + 20 \log d + 20 \log f \quad (3) \text{ dengan :}$$

Lfs = redaman ruang bebas (dB)
d = jarak antara antena pemancar ke penerima (km)
f = frekuensi (MHz)

E. Received Signal Level (RSL)

Received level signal adalah tingkat sinyal yang diterima di perangkat penerima dan nilainya harus lebih besar dari sensitivitas perangkat penerima (*received sensitivity*). Jika nilai *received level signal* lebih kecil dari sensitivitas penerima berarti sinyal yang dipancarkan tidak dapat diterima dengan baik oleh perangkat penerima. Secara matematis dinyatakan seperti pada persamaan (2) [5]: Rx level = EIRP – FSL + GRx – LRx (2)

dengan : EIRP = *effective isotropic radiated power*
FSL = *free space loss*
GRx = *gain* antena penerima
LRx = *loss* kabel antena penerima

III. METODOLOGI PENELITIAN

A. Alat dan Bahan

Adapun perangkat keras dan perangkat lunak yang dibutuhkan sebagai berikut:

Perangkat Keras:

- Ruckus zf7982 - 6 unit
- Alcatel-Lucent OAW-API05 - 7 unit
- Laptop
- Smartphone
- Meteran Gulung 30m

Perangkat Lunak:

- Netsurveyor
- Vistumbler
- WiFi Overview
- Wifi Analyzer

B. Metode Penelitian

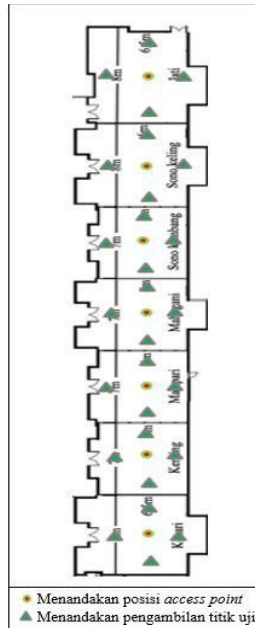
Metode penelitian yang dilakukan penulis dalam penelitian ini dapat dilihat pada *flow chart* berikut:



Gambar 1 Flowchart penelitian

C. Tempat Penelitian

Penelitian ini dilaksanakan di lingkungan *indoor ballroom* di lantai 2 pada ruangan *meeting*. Gambar 2 merupakan denah *Ballroom* lantai 2 yang menjadi tempat penelitian ini.



Gambar 2 Peta lantai 2 Ruang meeting

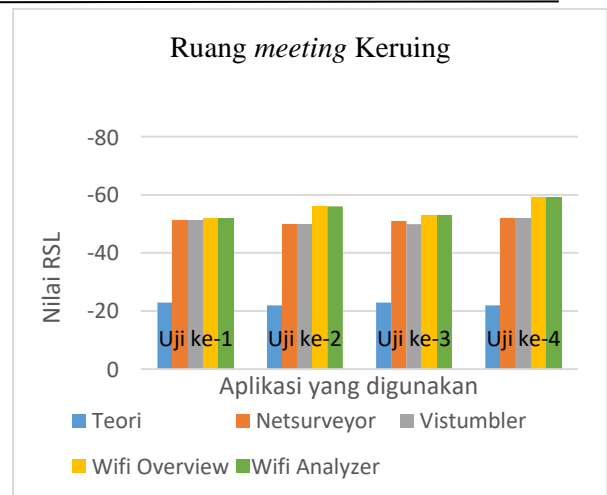
Lantai 2 ini terdiri dari 7 ruangan *meeting* yaitu Kenari, Keruing, Malapari, Mahogani, Sonokembang, Sonokeling dan Jati. Dimana ruangan *meeting* Kenari dan Sonokeling mempunyai luas 48 m^2 , Sementara ruangan *meeting* Keruing, Malapari, Mahogani dan Sonokembang mempunyai luas 42 m^2 serta ruangan *meeting* jati mempunyai luas 58.5 m^2 . Setiap ruangan terdiri dari 1 *access point* sehingga pada lantai 2 ini terdapat 7 *access point* yang akan digunakan sebagai objek penelitian dengan posisi *access point* memiliki ketinggian $\pm 5 \text{ m}$ dari atas lantai.

IV. PERHITUNGAN DAN PENGUKURANNILAI RSL

Setelah dilakukan perhitungan nilai RSL secara teori dengan persamaan 1, 2 dan 3 dan dengan pengukuran di lapangan dengan bantuan aplikasi yang telah disebutkan dapat ditunjukkan oleh gambar

A. Ruang Meeting Kenari

Hasil pengujian pada ruang kenari dapat dilihat pada gambar 3.



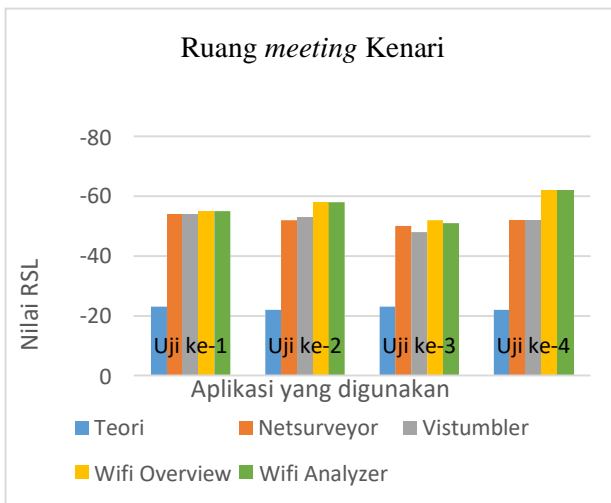
Gambar 3 Hasil pengujian di ruang meeting Kenari

Dari hasil pengujian yang dilakukan 4 titik uji yang ditunjukkan pada gambar 3 dapat disimpulkan bahwa:

- Pada titik uji ke-1 perhitungan RSL menggunakan rumus sebesar -23 dBm , sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -54 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -55 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-2 perhitungan RSL menggunakan rumus sebesar -22 dBm , sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -52 dBm (aplikasi Netsurveyor) dan -53 dBm (aplikasi Vistumbler) sedangkan pada *smartphone* -58 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-3 perhitungan RSL menggunakan rumus sebesar -23 dBm , sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -50 dBm (aplikasi Netsurveyor) dan -48 dBm (aplikasi Vistumbler) sedangkan pada *smartphone* -52 dBm (aplikasi Wifi Overview) dan -51 (aplikasi Wifi Analyzer).
- Pada titik uji ke-4 perhitungan RSL menggunakan rumus sebesar -22 dBm , sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -52 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -62 dBm (aplikasi Wifi Overview dan Wifi Analyzer).

B. Ruang Meeting Keruing

Hasil pengujian pada ruang keruing dapat dilihat pada gambar 4.



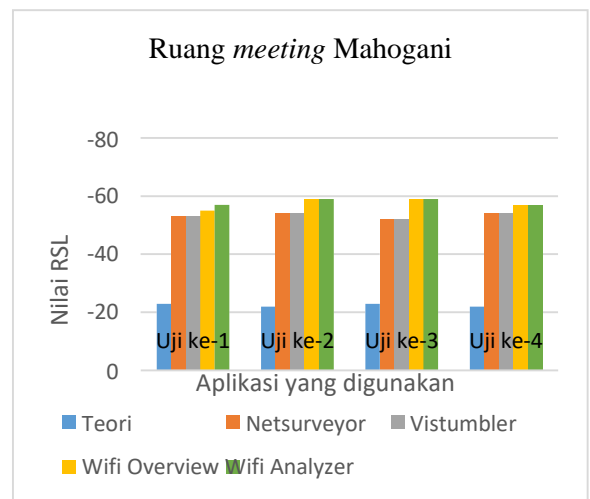
Gambar 4 Hasil pengujian di ruang meeting Keruing

Dari hasil pengujian yang dilakukan 4 titik uji yang ditunjukkan pada gambar 4 dapat disimpulkan bahwa:

- Pada titik uji ke-1 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -51 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -52 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-2 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -50 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -56 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-3 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -51 dBm (aplikasi Netsurveyor) dan -50 dBm (aplikasi Vistumbler) sedangkan pada *smartphone* -53 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-4 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -52 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -59 dBm aplikasi Wifi Overview dan Wifi Analyzer).

C. Ruang Meeting Malapari

Hasil pengujian pada ruang Malapari dapat dilihat pada gambar 5.



Gambar 5 Hasil pengujian di ruang meeting Malapari

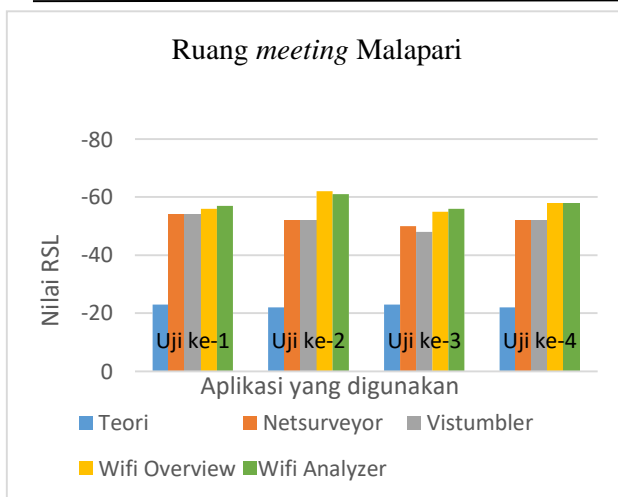
Dari hasil pengujian yang dilakukan 4 titik uji yang ditunjukkan pada gambar 5 dapat disimpulkan bahwa:

- Pada titik uji ke-1 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -54 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -56 dBm (aplikasi Wifi Overview) dan -57 dBm (aplikasi Wifi Analyzer).
- Pada titik uji ke-2 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -52 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -62 dBm (aplikasi Wifi Overview) dan -62 dBm (aplikasi Wifi Analyzer).
- Pada titik uji ke-3 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -50 dBm (aplikasi Netsurveyor) dan -48 dBm (aplikasi Vistumbler) sedangkan pada *smartphone* -55 dBm (aplikasi Wifi Overview) dan -56 dBm (aplikasi Wifi Analyzer).

Pada titik uji ke-4 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -52 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -58 dBm (aplikasi Wifi Overview dan Wifi Analyzer).

D. Ruang Meeting Mahogani

Hasil pengujian pada ruang Mahogani dapat dilihat pada gambar 6.



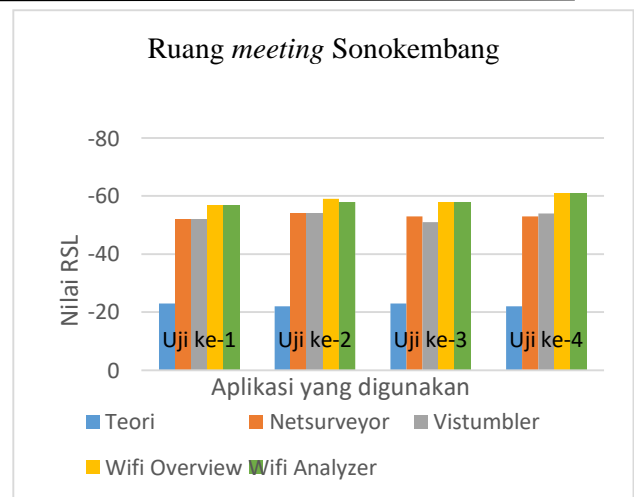
Gambar 6 Hasil pengujian di ruang meeting Mahogany

Dari hasil pengujian yang dilakukan 4 titik uji yang ditunjukkan pada gambar 6 dapat disimpulkan bahwa:

- Pada titik uji ke-1 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -52 dBm (aplikasi Netsurveyor dan Vistumbler), sedangkan pada *smartphone* -55 dBm (aplikasi Wifi Overview) dan -57 dBm (aplikasi Wifi Analyzer).
- Pada titik uji ke-2 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -53 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -59 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-3 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -52 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -59 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-4 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -54 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -57 dBm (aplikasi Wifi Overview dan Wifi Analyzer).

E. Ruang Meeting Sonokembang

Hasil pengujian pada ruang Sonokembang dapat dilihat pada gambar 7.



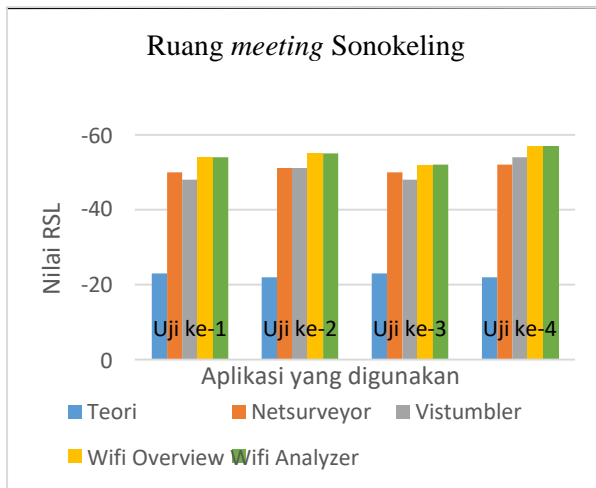
Gambar 7 Hasil pengujian di ruang meeting Sonokembang

Dari hasil pengujian yang dilakukan 4 titik uji yang ditunjukkan pada gambar 7 dapat disimpulkan bahwa:

- Pada titik uji ke-1 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -52 dBm (aplikasi Netsurveyor dan Vistumbler), sedangkan pada *smartphone* -57 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-2 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -54 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -59 dBm (aplikasi Wifi Overview) dan -58 dBm (aplikasi Wifi Analyzer).
- Pada titik uji ke-3 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -53 dBm (aplikasi Netsurveyor) dan -51 dBm (aplikasi Vistumbler) sedangkan pada *smartphone* -58 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-4 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -53 dBm (aplikasi Netsurveyor) dan -53 (aplikasi Vistumbler) sedangkan pada *smartphone* -61 dBm (aplikasi Wifi Overview dan Wifi Analyzer).

F. Ruang Meeting Sonokeling

Hasil pengujian pada ruang Sonokeling dapat dilihat pada gambar 8.



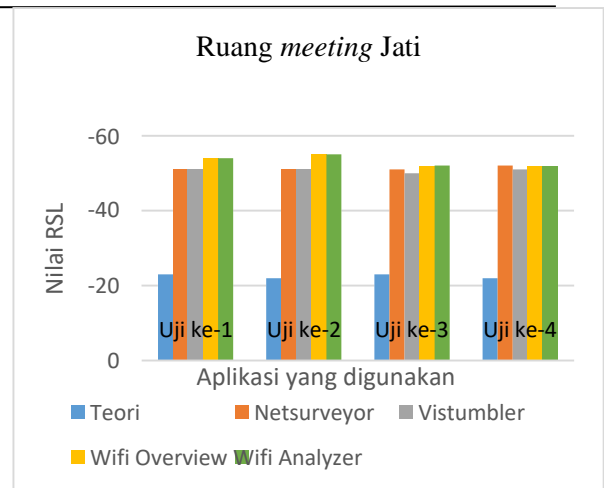
Gambar 8 Hasil pengujian di ruang meeting Sonokeling

Dari hasil pengujian yang dilakukan 4 titik uji yang ditunjukkan pada gambar 8 dapat disimpulkan bahwa:

- Pada titik uji ke-1 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -50 dBm (aplikasi Netsurveyor) dan -48 dBm (aplikasi Vistumbler), sedangkan pada *smartphone* -54 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-2 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -51 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -55 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-3 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -50 dBm (aplikasi Netsurveyor) dan -48 dBm (aplikasi Vistumbler) sedangkan pada *smartphone* -52 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-4 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -52 dBm (aplikasi Netsurveyor) dan -54 (aplikasi Vistumbler) sedangkan pada *smartphone* -57 dBm (aplikasi Wifi Overview dan Wifi Analyzer).

G. Ruang Meeting Jati

Hasil pengujian pada ruang Jati dapat dilihat pada gambar 9.



Gambar 9 Hasil pengujian di ruang meeting Jati

Dari hasil pengujian yang dilakukan 4 titik uji yang ditunjukkan pada gambar 9 dapat disimpulkan bahwa:

- Pada titik uji ke-1 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -51 dBm (aplikasi Netsurveyor dan Vistumbler), sedangkan pada *smartphone* -54 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-2 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -51 dBm (aplikasi Netsurveyor dan Vistumbler) sedangkan pada *smartphone* -55 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-3 perhitungan RSL menggunakan rumus sebesar -23 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -51 dBm (aplikasi Netsurveyor) dan -50 dBm (aplikasi Vistumbler) sedangkan pada *smartphone* -52 dBm (aplikasi Wifi Overview dan Wifi Analyzer).
- Pada titik uji ke-4 perhitungan RSL menggunakan rumus sebesar -22 dBm, sedangkan dalam pengukuran di lapangan nilai RSL untuk perangkat laptop sebesar -51 dBm (aplikasi Netsurveyor) dan -52 dBm (aplikasi Vistumbler) sedangkan pada *smartphone* -52 dBm (aplikasi Wifi Overview dan Wifi Analyzer).

Nilai RSL dari pengukuran di lapangan mempunyai perbedaan jauh dengan perhitungan secara teori, karena perhitungan teori mengabaikan semua faktor redaman di lapangan seperti pengaruh difraksi, refraksi, refleksi, absorpsi maupun *blocking*. Tetapi pengukuran di lapangan secara indikator menunjukkan “sangat bagus” (nilai RSL dibawah 58 dBm) dan “bagus” (nilai RSL antara 58 dBm samapai 70 dbm), serta tidak terdapat indikator “rendah” (nilai RSL diatas 75 dbm), artinya koneksi di setiap ruangan *meeting* sudah baik dan normal.

V. PENUTUP

A. Kesimpulan

Berdasarkan hasil pengamatan dari perhitungan dan

pengukuran nilai RSL yang telah dilakukan selama penelitian, dapat diambil kesimpulan antara lain :

1. Koneksi jaringan *wireless* pada lantai 2 ruangan *meeting* Ruang *meeting* sudah baik dan normal.
2. Beberapa faktor yang mempengaruhi besar nilai RSL antara lain *gain* pemancar, *free space loss*, Tx *power*, jarak pemancar dengan penerima, faktor pengaruh hambatan (difraksi, refraksi, refleksi, absorpsi maupun *blocking*), Rx sensitifitas dan *gain* di sisi penerima.
3. Dari perhitungan secara teoritis nilai RSL setiap ruangan *meeting* adalah -23 dBm (titik uji 1 dan 3) dan -22 dbm (titik uji 2 dan 4).
4. Dari pengukuran di lapangan secara indikator menunjukkan “sangat bagus” dan “bagus”, serta tidak terdapat indikator “rendah”.
5. Adanya perbedaan cukup besar dari hasil perhitungan dan pengukuran nilai RSL disebabkan karena perhitungan teori mengabaikan semua faktor redaman di lapangan seperti pengaruh difraksi, refraksi, refleksi, absorpsi maupun *blocking*.

B. Saran

1. Perlu adanya pengaturan *channel* setiap *access point* untuk menghindari terjadinya interferensi *channel* yang dapat menyebabkan penerunan kualitas sinyal.
2. Perlu adanya pengoptimalan dalam penempatan *access point* agar terhindar dari *obstacle* (hambatan) maupun pengaruh dari difraksi, refraksi, refleksi, absorpsi maupun *blocking*.

DAFTAR PUSTAKA

- [1] I. Kusuharto dan E. Wiyanto, “Simulasi Perancangan Wireless LAN pada Area Perkantoran,” vol. I, p. 12, 2012.
- [2] R. Sallam, “Pengembangan Jaringan Komputer WLAN dengan Menggunakan ServerFreeBSD pada PT. KMK Global,” Jakarta Barat, 2010.
- [3] Telkom Speedy, “Link Budget,” 1 april 2015. [Online]. Available: http://opensource.telkomspeedy.com/wiki/index.php/WiFi:_Menghitung_Link_Bu_dget. [Diakses 22 mei 2017].
- [4] D. Angela, “Optimasi Jaringan Wireless LAN (Studi Kasus di Kampus ITHB Bandung),” 2015.
- [5] F. A. Manurung dan N. Mubarakah, “Analisis Link Budget Untuk Koneksi Radio,” vol. VII, no. 2, 2014.
- [6] H. T. Pambudhi, “Analisis Kekuatan Daya Received Signal Level (RSL) menggunakan Piranti Sagem Link Terminal di PT Pertamina EP Region Jawa,” p. 1, 2010.

ANALISIS TRAFIK DATA PADA SISTEM PEMANTAU ARUS LISTRIK PANEL HUBUNG BAGI

Mohammad Rizki Romadhon, Muhammad Arrofiq
Departemen Teknik Elektro dan Informatika
mhderiski@gmail.com
rofiqm@ugm.ac.id

Abstract – Change and Innovation continues as the times progress. Especially in the field of technology, one of which is the Internet of Things (IoT). With the existence of this technology, electronic devices can be connected to the internet. This will simplify the process of data transmission and device control. With this aim, the authors do research on the implementation of IoT technology on electronic devices, especially in the field of electricity. Electrical energy monitoring systems are used to detect the use of electrical energy used by the load. In this monitoring system, the measurement of the electric current using the current sensor. The measurement data made by the current sensor will be processed by the microcontroller. The research developed arduino-based monitoring systems and databases aimed at detecting and monitoring the use of electric current. In addition, it also performs analysis of data traffic from arduino devices and databases, to find out data delay, throughput or packet loss by using Wireshark.

Keywords : Monitoring, Arduino, electrical current, wireshark

Intisari – Inovasi dan perubahan terus terjadi seiring perkembangan jaman. Terutama dalam bidang teknologi, salah satunya adalah Internet of Things (IoT). Dengan adanya teknologi ini maka perangkat elektronik dapat terhubung dengan internet. Hal ini akan mempermudah proses pengiriman data dan kontrol perangkat. Dengan tujuan tersebut penulis melakukan penelitian terhadap implementasi teknologi IoT pada perangkat elektronik terutama dalam bidang kelistrikan. Sistem pemantau energi listrik digunakan untuk mendeteksi penggunaan energi listrik yang dipakai oleh beban. Dalam sistem pemantau ini dilakukan pengukuran besaran arus listrik menggunakan sensor arus. Data hasil pengukuran yang dilakukan oleh sensor arus akan diolah oleh mikrokontroler. Penelitian ini mengembangkan sistem pemantau berbasis arduino dan database yang bertujuan untuk mendeteksi dan memantau penggunaan arus listrik. Selain itu, juga melakukan analisa terhadap trafik data dari perangkat arduino dan database, untuk mengetahui delay data, throughput ataupun packet loss dengan menggunakan Wireshark.

Kata kunci : Monitoring, arduino , arus listrik, database, Wireshark

I. PENDAHULUAN

Internet of Things (IoT) adalah komunikasi antara satu perangkat dengan perangkat yang lainnya melalui jaringan internet. Dengan adanya *IoT* akan mempermudah pengguna dalam melakukan kontrol terhadap perangkat elektronik yang digunakan. Salah satu contoh implementasi dari *IoT* adalah *home automation* atau lebih dikenal dengan *smart home*. *Smart home* merupakan suatu bangunan yang menggunakan teknologi dan internet untuk mengendalikan dan memantau berbagai peralatan dari jarak jauh (*remote*). Penerapan *IoT* pada perangkat elektronik sangat lah penting terutama pada perangkat pengamanan bangunan. Hal ini akan memberikan rasa nyaman dan aman terhadap pengguna dalam melakukan kegiatan sehari-hari. Segala kemudahan yang ditawarkan oleh *smart home* harus lah diikuti dengan penghematan energi listrik.

Ada beberapa cara yang dapat dilakukan untuk menghemat energi listrik. Di antaranya dengan memasang suatu sistem pemantau penggunaan arus listrik pada suatu bangunan. Dengan menerapkan sistem pemantau berbasis *IoT* maka sistem dapat dipantau secara jarak jauh (*remote*). Penerapan teknologi *IoT* dapat dilakukan dengan menggunakan mikrokontroler. Dengan pemasangan mikrokontroler pada perangkat elektronik maka akan mempermudah koneksi pengiriman dan penerimaan data yang dilakukan. Karena data yang dikirim dan diterima melalui jaringan internet maka perangkat tersebut akan lebih mudah dipantau jarak jauh.

II. TINJAUAN PUSTAKA

2.1 Mikrokontroler

Mikrokontroler adalah sebuah *chip* yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program di dalamnya. Mikrokontroler umumnya terdiri dari *Central Processing Unit (CPU)*, memori, *I/O* tertentu dan unit pendukung seperti *Analog-to-Digital Converter (ADC)* yang sudah terintegrasi di dalamnya. Salah satu jenis mikrokontroler yang sering digunakan adalah arduino. Arduino merupakan papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel. Salah satu jenis arduino yang digunakan dalam penelitian ini adalah Arduino Uno. Arduino uno merupakan arduino yang paling banyak digunakan dengan versi terakhir yaitu Arduino Uno R3. Dalam proses pengiriman datanya mikrokontroler dapat menggunakan berbagai macam cara [1]. Cara yang paling umum digunakan adalah dengan mengirim data menggunakan kabel LAN. Agar arduino dapat mengirim data menggunakan LAN, maka dibutuhkan modul tambahan yang bernama *Ethernet Shield*. *Ethernet Shield* adalah modul yang dapat menghubungkan arduino ke internet. Hal ini didasarkan pada *chip* W5100 Wiznet *Ethernet* yang menyediakan jaringan internet secara *TCP* dan *UDP*. Pemasangan ethernet shield dilakukan dengan menumpuknya di atas arduino, kemudian disambungkan dengan kabel *network*

RJ-45 atau biasa disebut dengan kabel LAN. Di dalam *ethernet shield* tersebut terdapat *slot micro SD* yang berbungsi sebagai tempat penyimpanan *file*. Untuk jenis *arduino board* yang bisa dipasangkan dengan *ethernet shield W5100* yaitu *arduino uno* dan *arduino mega*.

2.2 Internet of Things

Pada era modern ini mikrokontroler telah digunakan pada berbagai macam perangkat. Dengan terkoneksi mikrokontroler dengan internet, membuat perangkat memiliki kemampuan untuk *transfer* data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. Konsep inilah yang disebut dengan *Internet of Things (IoT)* [2]. *Internet of Things (IoT)* merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus. Suatu benda dikatakan *IoT* apabila terdapat pada suatu benda elektronik atau peralatan apa saja yang tersambung ke suatu jaringan lokal dan *global* melalui sensor yang tertanam dan selalu aktif. *IoT* memperkenalkan kesempatan baru seperti kapabilitas untuk memantau dan memanajemen perangkat secara *remote*, menganalisa dan mengambil tindakan berdasarkan pada informasi yang diterima dari berbagai macam koneksi data *real-time* [3]. Sebagai hasilnya, produk *IoT* dapat merubah kota dengan cara meningkatkan infrastruktur, membuat layanan kota lebih efektif dan murah, meningkatkan layanan transportasi dengan mengurangi kemacetan kota dan meningkatkan keamanan penduduk [4]. Terdapat banyak cara dalam penerapan teknologi *IoT*, salah satunya adalah pemantauan lingkungan dengan cara menganalisa dan menyebarkan informasi yang dikumpulkan dari berbagai lingkungan. Arsitektur pada teknologi *IoT* dibagi menjadi 3 lapisan utama. Lapisan tersebut berupa *Application layer*, *Network layer*, dan *Perception layer* [5]. *Perception layer* adalah lapisan paling bawah pada arsitektur *IoT* atau biasa disebut dengan lapisan persepsi dan kontrol. *Perception layer* bertugas dalam pengumpulan informasi objek fisik, transaksi atau proses yang menggunakan sensor alat. *Network layer* adalah lapisan tengah pada arsitektur *IoT* atau biasa disebut dengan lapisan transmisi. *Network layer* bertugas untuk mengirimkan informasi yang diterima oleh *Perception layer* ke *Application layer* dengan aman, cepat dan handal. *Application layer* adalah lapisan paling atas dalam arsitektur *IoT*. Lapisan ini menganalisa dan memproses informasi yang datang dari *Perception layer* dan *Network layer*. Dapat dibilang lapisan ini adalah antarmuka antara perangkat *IoT* dan pengguna.

2.3 Quality of Service

Informasi yang dikumpulkan oleh *Perception layer* harus dapat dikirim dan ditampilkan dengan akurat. Jika terjadi gangguan saat pengiriman data maka data tidak dapat ditampilkan dengan sempurna. karena komunikasi dan kemampuan pengambilan keputusan yang *real-time* dari sensor, memprediksi *Quality of Service (QoS)* perangkat *IoT* sangatlah penting untuk mendeteksi kinerja maksimal dan jadwal waktu mati dan hidup perangkat [6]. Karakteristik aplikasi *IoT* adalah dengan komunikasi antara sensor, *gateway nodes* dan *server* atau *data center*.

Perangkat *IoT* akan mengirimkan informasi berupa paket dengan berbagai waktu sesuai dengan kebutuhan aplikasi. Banyak faktor yang dapat mempengaruhi kinerja aplikasi. Seperti contoh, mobilitas *gateway* yang dapat menimbulkan dampak signifikan pada *throughput* dan *round trip times* serta besar kecil paket yang dikirim dan diterima dari berbagai perangkat. Jadi pemahaman tentang dampak faktor sangatlah penting untuk menentukan kinerja dari jaringan itu sendiri. Berikut beberapa parameter *QoS* :

- Packet loss*, didefinisikan sebagai kegagalan transmisi paket yang disebabkan oleh beberapa kemungkinan, yaitu terjadinya *overload* trafik dan tabrakan dalam jaringan, *error* yang terjadi pada media fisik, dan kegagalan yang terjadi pada sisi penerima. Tabel 2.1 merupakan tabel kategori dari *packet loss*.

Tabel 2.1 *Packet Loss*

Kategori Degradasi	Packet Loss	Indeks
Sangat Bagus	0% - 2%	4
Bagus	3% - 14%	3
Sedang	15% - 24%	2
Buruk	>25%	1

Persamaan untuk menghitung *packet loss* dapat dilihat pada persamaan (1).

$$packet\ loss = \frac{packet\ data\ dikirim - paket\ data\ diterima}{paket\ data\ diterima} \times 100\% \quad (1)$$

- Delay, merupakan waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik yang lain. Tabel 2.2 merupakan tabel kategori delay.

Tabel 2.2 *Delay*

Kategori Latensi	Besar Delay	Indeks
Sangat Bagus	< 150 ms	4
Bagus	150 s/d 300 ms	3
Sedang	300 s/d 450 ms	2
Buruk	>450 ms	1

Delay diperoleh dari selisih waktu kirim antara suatu paket dengan paket lainnya yang direpresentasikan dalam satuan detik.

Persamaan untuk menghitung delay dapat dilihat pada persamaan (2).

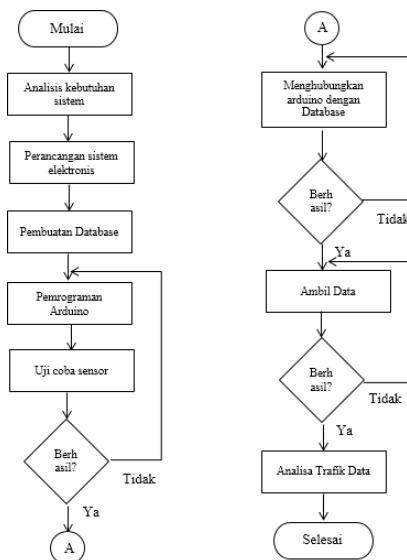
$$delay = waktu\ paket\ terima - waktu\ paket\ kirim \quad (2)$$

- Throughput, merupakan bandwidth yang sebenarnya, diukur dengan satuan waktu tertentu dan pada kondisi jaringan tertentu yang digunakan untuk melakukan pengiriman data dengan ukuran tertentu. Persamaan untuk menghitung *throughput* dapat dilihat pada persamaan (3).

$$throughput = \frac{jumlah\ data\ yang\ dikirim}{waktu\ pengiriman\ data} \quad (3)$$

III. METODELOGI PENELITIAN

Penelitian ini dilakukan dalam beberapa tahap yaitu : analisis kebutuhan, perancangan alat, pengujian, pengambilan data dan analisis data. Analisis kebutuhan merupakan tahap di mana peneliti menganalisa alat dan bahan apa saja yang diperlukan untuk merancang sistem. Perancangan alat merupakan tahap di mana peneliti mulai memasang alat-alat dan melakukan coding pada alat. Pengujian merupakan tahap di mana peneliti melakukan uji coba, apakah alat berjalan sesuai dengan yang diinginkan atau tidak. Pengambilan data merupakan tahap di mana peneliti mengambil data sesuai dengan skenario yang telah direncanakan. Analisa data merupakan tahap di mana peneliti menganalisa data yang telah diambil sebelumnya. Adapun alur penelitian dipaparkan pada gambar 3.1.



Gambar 3.1 Bagan Alur Metode Penelitian

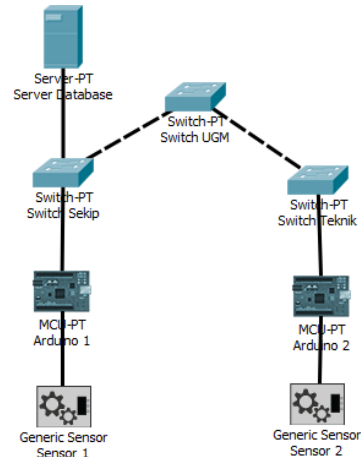
3.1 Analisis Kebutuhan

Pada penelitian ini, alat dan bahan yang dibutuhkan cukup beragam, yakni terdiri dari perangkat keras dan perangkat lunak. Perangkat keras yang digunakan yaitu 2 buah arduino untuk mengolah data sensor, 2 buah sensor arus untuk membaca arus listrik, 2 buah sensor RTC untuk mencatat waktu, 2 buah ethernet shield untuk mengirimkan data sensor ke database dan 1 buah komputer yang digunakan sebagai server database. Sedangkan perangkat lunak yang digunakan untuk menjalankan dan mengambil data yaitu xampp untuk layanan database dan webserver, arduino ide untuk memprogram arduino dan wireshark untuk mengambil dan menganalisa data.

3.2 Perancangan Topologi

Pengujian dalam penelitian ini dilakukan dalam 4 skenario, ke empatnya menggunakan 2 arduino dan 3 buah

sensor untuk masing-masing arduino untuk mengirimkan data ke server database. Perbedaan dari skenario terletak pada interval waktu pengambilan data. Topologi yang digunakan dapat dilihat pada gambar 3.2.



Gambar 3.2 Infrastruktur Jaringan IoT

3.3 Skenario Pengujian

Pada penelitian ini melakukan pengujian pada Quality of Service (delay, packet loss, throughput). Nilai yang dihasilkan nanti akan digunakan dalam analisis. Topologi yang digunakan sesuai pada gambar. Terdapat 4 skenario yang dilakukan. Pada skenario pertama, waktu pengambilan data dilakukan selama 5 menit. Pada Skenario kedua, waktu pengambilan data dilakukan selama 10 menit. Pada skenario ketiga, waktu pengambilan data dilakukan selama 30 menit. Dan pada skenario terakhir, waktu pengambilan data dilakukan selama 60 menit.

Pengukuran QoS dilakukan untuk mengukur performa jaringan dari perangkat IoT yang diterapkan pada jaringan UGM. Pengambilan data pada alat 1 dan 2 dilakukan secara bersamaan.

IV. ANALISA DAN PEMBAHASAN

Berdasarkan pengambilan data yang telah dilakukan menggunakan tools berupa wireshark, didapatkan data berupa :

4.1 Skenario 1

Pengujian dilakukan dalam waktu kurang lebih 5 menit dan dapat dilihat pada tabel 4.1.

Tabel 4.1 Waktu Skenario 1

Alat	Data Awal	Data Akhir
Alat 1 – Gedung Sekip	12.39.18 PM	12.45.01 PM
Alat 2 – Gedung Grafika		

Gambar 4.1 merupakan keterangan dari data alat 1 dan alat 2 skenario 1 yang tertangkap pada wireshark setelah melakukan pengujian skenario 1, terdapat 5 data yang berasal dari alat 1 dan 6 data dari alat 2.

Measurement	Marked	Marked
Packets	5 (0.0%)	6 (0.0%)
Time span, s	239.976	300.216
Average pps	0.0	0.0
Average packet size, B	60	60
Bytes	300 (0.0%)	360 (0.0%)
Average bytes/s	1	1
Average bits/s	10	9

Gambar 4.1 Wireshark Skenario 1

Pengujian skenario pertama terlihat pada tabel 4.2, nilai *QoS* yang didapat dari 5 data pada alat 1 dan 6 data pada alat 2 adalah sebagai berikut :

Tabel 4.2 Hasil Data Skenario 1

Parameter <i>QoS</i>	Hasil	
	Alat 1	Alat 2
<i>Delay</i>	0.040 detik	0.047 detik
<i>Packet Loss</i>	0%	0%
<i>Throughput</i>	10 <i>bps</i>	9 <i>bps</i>

- Delay*, untuk alat 1 sebesar 0.040 detik dan alat 2 sebesar 0.047 detik. *Delay* pada alat 2 lebih besar jika dibanding dengan alat 1, ini dikarenakan alat 2 berada pada lokasi yang berbeda dan lebih jauh dari *server database* dibanding dengan alat 1.
- Packet loss*, pada kedua alat *packet loss* tercatat sebesar 0%. Artinya tidak ada data yang hilang ketika kedua alat mengirimkan data ke *server database*.
- Throughput*, tercatat pada alat 1 nilai *throughput* sebesar 10 *bps* dan pada alat 2 sebesar 9 *bps* yang berarti beban trafik tidak besar. Hal ini dikarenakan data yang dikirim ke *server database* hanya berupa variabel-variabel nilai dalam bentuk *file txt/html*.

4.2 Skenario 2

Pengujian dilakukan dalam waktu kurang lebih 10 menit dan dapat dilihat pada tabel 4.3.

Tabel 4.3 Waktu Skenario 2

Alat	Data Awal	Data Akhir	
Alat 1 – Gedung Sekip			
Alat 2 – Gedung Grafika	12.50.04 PM	13.00.01 PM	

Gambar 4.2 merupakan keterangan dari data alat 1 dan alat 2 skenario 2, terdapat 10 data yang berasal pada alat 1 dan 10 data yang berasal dari alat 2.

Measurement	Marked	Marked
Packets	5 (0.0%)	6 (0.0%)
Time span, s	239.976	300.216
Average pps	0.0	0.0
Average packet size, B	60	60
Bytes	300 (0.0%)	360 (0.0%)
Average bytes/s	1	1
Average bits/s	10	9

Gambar 4.2 Wireshark Skenario 2

Pengujian skenario kedua terlihat pada tabel 4.4, nilai *QoS* yang didapat dari 10 data pada alat 1 dan 10 data pada alat 2 adalah sebagai berikut :

Tabel 4.4 Hasil Data Skenario 2

Parameter <i>QoS</i>	Hasil	
	Alat 1	Alat 2
<i>Delay</i>	0.037 detik	0.041 detik
<i>Packet Loss</i>	0%	0%
<i>Throughput</i>	8 <i>bps</i>	8 <i>bps</i>

- Delay*, untuk alat 1 sebesar 0.037 detik dan alat 2 sebesar 0.041 detik. *Delay* pada alat 2 lebih besar jika dibanding dengan alat 1, ini dikarenakan alat 2 berada pada lokasi yang berbeda dan lebih jauh dari *server database* dibanding dengan alat 1.
- Packet loss*, pada kedua alat *packet loss* tercatat sebesar 0%. Artinya tidak ada data yang hilang ketika kedua alat mengirimkan data ke *server database*.
- Throughput*, tercatat pada alat 1 nilai *throughput* sebesar 8 *bps* dan pada alat 2 sebesar 8 *bps* yang berarti beban trafik tidak besar. Hal ini dikarenakan data yang dikirim ke *server database* hanya berupa variabel-variabel nilai dalam bentuk *file txt/html*.

4.3 Skenario 3

Pengujian dilakukan dalam waktu kurang lebih 30 menit dan dapat dilihat pada tabel 4.5.

Tabel 4.5 Waktu Skenario 3

Measurement	Marked	Marked
Packets	10 (0.0%)	10 (0.0%)
Time span, s	540.376	539.885
Average pps	0.0	0.0
Average packet size, B	60	60
Bytes	600 (0.0%)	600 (0.0%)
Average bytes/s	1	1
Average bits/s	8	8

Alat	Data Awal	Data Akhir
Alat 1 – Gedung Sekip	13.05.01 PM	13.35.37 PM
Alat 2 – Gedung Grafika		

Gambar 4.3 merupakan keterangan dari data alat 1 dan alat 2 skenario 2, terdapat 31 data yang berasal pada alat 1 dan 30 data yang berasal dari alat 2.

Gambar 4.3 Wireshark Skenario 3

Pengujian skenario ketiga terlihat pada tabel 4.6, nilai *QoS* yang didapat dari 31 data pada alat 1 dan 30 data pada alat 2 adalah sebagai berikut :

Tabel 4.6 Hasil Data Skenario 3

Parameter <i>QoS</i>	Hasil	
	Alat 1	Alat 2
<i>Delay</i>	0.041 detik	0.043 detik
<i>Packet Loss</i>	0%	0%
<i>Throughput</i>	8 <i>bps</i>	8 <i>bps</i>

- a. Delay, untuk alat 1 sebesar 0.041 detik dan alat 2 sebesar 0.043 detik. Delay pada alat 2 lebih besar jika dibanding dengan alat 1, ini dikarenakan alat 2 berada pada lokasi yang berbeda dan lebih jauh dari server database dibanding dengan alat 1.
- b. Packet loss, pada kedua alat packet loss tercatat sebesar 0%. Artinya tidak ada data yang hilang ketika kedua alat mengirimkan data ke server database.
- c. Throughput, tercatat pada alat 1 nilai throughput sebesar 8 bps dan pada alat 2 sebesar 8 bps yang berarti beban trafik tidak besar. Hal ini dikarenakan data yang dikirim ke server database hanya berupa variabel-variabel nilai dalam bentuk file txt/html.

4.4 Skenario 4

Pengujian dilakukan dalam waktu kurang lebih 60 menit dan dapat dilihat pada tabel 4.7.

Tabel 4.7 Waktu Skenario 4

Alat	Data Awal	Data Akhir
Alat 1 – Gedung Sekip	14.30.02 PM	15.30.03 PM
Alat 2 – Gedung Grafika		

Gambar 4.4 merupakan keterangan dari data alat 1 dan alat 2 skenario 2, terdapat 61 data yang berasal pada alat 1 dan 60 data yang berasal dari alat 2.

Measurement	Marked	Marked
Packets	61 (0.0%)	60 (0.0%)
Time span, s	3597.648	3542.437
Average pps	0.0	0.0
Average packet size, B	60	60
Bytes	3660 (0.0%)	3600 (0.0%)
Average bytes/s	1	1
Average bits/s	8	8

Gambar 4.4 Wireshark Skenario 4

Pengujian skenario keempat terlihat pada tabel 4.8, nilai QoS yang didapat dari 61 data pada alat 1 dan 60 data pada alat 2 adalah sebagai berikut :

Tabel 4.8 Hasil Data Skenario 4

Parameter QoS	Hasil	
	Alat 1	Alat 2
Delay	0.035 detik	0.038 detik
Packet Loss	0%	0%
Throughput	8 bps	8 bps

- a. Delay, untuk alat 1 sebesar 0.035 detik dan alat 2 sebesar 0.038 detik. Delay pada alat 2 lebih besar jika di banding dengan alat 1, ini dikarenakan alat 2 berada pada lokasi yang berbeda dan lebih jauh dari server database dibanding dengan alat 1.
- b. Packet loss, pada kedua alat packet loss tercatat sebesar 0%. Artinya tidak ada data yang hilang ketika kedua alat mengirimkan data ke server database.

- c. Throughput, tercatat pada alat 1 nilai throughput sebesar 8 bps dan pada alat 2 sebesar 8 bps yang berarti beban trafik tidak besar. Hal ini dikarenakan data yang dikirim ke server database hanya berupa variabel-variabel nilai dalam bentuk file txt/html.

V. KESIMPULAN

Berdasarkan hasil analisis pengujian terhadap QoS dari sistem pemantau arus listrik pada panel hubung bagi, maka dapat diambil kesimpulan sebagai berikut :

1. Parameter QoS yaitu Delay dapat dipengaruhi oleh jarak antara sensor dengan server. Kualitas jaringan internet yang dipakai serta infrastruktur jaringan saat dilakukan pengujian. Karena pengujian dilakukan menggunakan 2 buah sensor yang terletak pada gedung yang berbeda. Maka pengiriman data dipengaruhi oleh infrastruktur dan jarak dari jaringan tersebut.
2. Parameter QoS yaitu Throughput terpengaruhi oleh banyaknya trafik data yang terjadi antara sensor dengan server. Sebagai contoh pada skenario 1 tercatat nilai throughput sebesar 10 bps untuk sensor 1 dan 9 bps untuk sensor 2. Lain halnya dengan skenario 2, 3 dan 4 yang tercatat nilai throughput sebesar 8 bps untuk sensor 1 dan 8 bps untuk sensor 2. Nilai throughput dari semua skenario terlihat hampir sama, karena jumlah data yang dikirim dan waktu pengambilan data sebanding.
3. Parameter QoS yaitu Packet Loss terpengaruhi oleh kualitas jaringan yang digunakan sensor untuk server, sebagai contoh pada skenario 1,2,3 dan 4 tidak ada data yang hilang. Ini dikarenakan pada semua skenario diterapkan pada jaringan local ugm. Jika percobaan dilakukan pada jaringan local, sedikit kemungkinan terdapat paket data yang hilang.

DAFTAR PUSTAKA

[1] R. Goyal and D. L. Arya, "Home Automation and Intelligent Light Control System using Microcontroller," *IEEE*, pp. 997-1000, 2017.

[2] L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A survey," *ScienceDirect*, pp. 2787-2805, 2010.

[3] N. M. Morshed, G. Muid-Ur-Rahman, M. R. Karim and H. U. Zaman, "Microcontroller Based Home Automation System Using Bluetooth, GSM, Wi-Fi and DTMF," *IEEE*, pp. 101-104, 2015.

[4] B. Hammi, R. Khatoun, S. Zeadally, A. Fayad and L. Khoukhi, "IoT technologies for smart cities," *IET*

Networks, vol. VII, no. 1, pp. 1-13, 2017.

- [5] C.-l. Zhong, Z. Zhu and R.-g. Huang, "Study on the IOT Architecture and Access Technology," *IEEE*, pp. 113-116, 2017.
- [6] S. Sankaran, "Modeling the Performance of IoT networks," *IEEE*, pp. 1-6, 2016.

PERBANDINGAN KINERJA ANTARA MODUL XBEE S2 DENGAN MODUL ESP8266 (12E) PADA SISTEM PEMANTAU KUALITAS UDARA

Ricky Hastomo E, Muhammad Arrofiq
Departemen Teknik Elektro dan Informatika Sekolah Vokasi
Universitas Gadjah Mada
Ricky.hastomo.e@ugm.ac.id, rofiqm@ugm.ac.id

Abstarct - The development of wireless sensor network (WSN) technology has progressed very rapidly. These developments also triggered the development of wireless modules used in WSN. Xbee and ESP8266 are wireless modules that are often used to design wireless sensor networks. Xbee uses the IEEE 802.15.4 standard while ESP8266 uses the IEEE 802.11 b / g / n standard. In this research will be comparison of wireless modules between Xbee and ESP8266 to test the performance of the wireless module. Comparison is performed by testing the value of throughput, delay packet, and packet lost. The test will be applied to the air quality monitoring system using the MQ135 sensor. The result of the research is the comparison of parameters tested from both wireless modules.

Keyword: Xbee, WSN, ESP8266, module wireless, MQ135

Intisari - Perkembangan teknologi jaringan sensor nirkabel (JSN) mengalami kemajuan yang sangat pesat pada bekangan ini. Perkembangan tersebut juga memicu berkembangnya modul *wireless* yang digunakan dalam JSN. Xbee dan ESP8266 merupakan modul *wireless* yang sering digunakan untuk merancang jaringan sensor nirkabel. Xbee menggunakan standar IEEE 802.15.4 sedangkan ESP8266 menggunakan standar IEEE 802.11 b/g/n. Dalam penelitian ini akan dilakukan perbandingan modul *wireless* antara Xbee dan ESP8266 untuk menguji kinerja dari modul *wireless* tersebut. Perbandingan dilakukan dengan cara menguji nilai throughput, delay packet, dan packet lost. Pengujian akan diterapkan pada sistem pemantauan kualitas udara dengan menggunakan sensor MQ135. Hasil dari penelitian merupakan perbandingan nilai paramater yang diuji dari kedua modul *wireless* tersebut.

Kata Kunci : Xbee, JSN, ESP8266, modul *wireless*, MQ135

I. PENDAHULUAN

Perkembangan teknologi jaringan sensor nirkabel (JSN) mengalami kemajuan yang pesat, hal ini dikarenakan dengan menggunakan teknologi nirkabel pada sensor mempunyai banyak keuntungan seperti praktis, sensor bersifat *mobile*, meningkatkan efisiensi secara operasional, dan dapat mengumpulkan data dalam jumlah besar [1]. Oleh karena itu, aplikasi JSN sudah banyak digunakan dalam berbagai sistem sensor seperti sensor kebakaran, sensor bencana alam, sistem deteksi tempat parkir dan sebagainya.

JSN merupakan teknologi nikabel yang terdiri dari kumpulan *node* sensor yang ditempatkan di suatu area tertentu. Teknologi nirkabel yang diterapkan dalam JSN antara lain modul Xbee dan ESP8266. Dua modul ini menggunakan standar yang berbeda, Xbee menggunakan standar Zigbee 802.15.4 sedangkan ESP98266 menggunakan standar *WiFi* 802.11 b/g/n [2].

Dalam penelitian ini akan dilakukan perbandingan kinerja dari modul *wireless* yang menggunakan standar Zigbee dan *WiFi* yaitu Xbee S2 dan ESP8266 pada sistem pemantau kualitas udara yang menggunakan sensor MQ135. Nilai parameter yang akan diuji adalah *throughput*, *packet loss* dan *delay*. Sehingga dari hasil pengujian kinerja kedua modul *wireless* tersebut dapat diketahui manakah modul *wireless* yang lebih baik dalam perancangan jaringan sensor nirkabel.

II. TEORI PENDUKUNG

A. Jaringan Sensor Nirkabel

Jaringan *wireless* adalah bidang disiplin Jaringan nirkabel telah banyak digunakan untuk membangun sensor, yang disebut jaringan sensor nirkabel (JSN) atau bisanya disebut dengan *wireless sensor network* (WSN). Jaringan sensor nirkabel (JSN) terdiri dari sejumlah *node* sensor yang dilengkapi dengan modul *wireless* yang bekerja

sama untuk memantau kondisi lingkungan tertentu. *Node* Sensor berkomunikasi dan mengirim data melalui jaringan nirkabel. JSN dapat membentuk jaringan yang efisien dan kuat, sehingga meningkatkan kinerja secara signifikan [3]. Gambar 2.1 merupakan contoh Arsitektur Jaringan Sensor Nirkabel.

B. Modul *Wireless* Pada JSN

Dalam perancangan jaringan sensor nirkabel diperlukan modul *wireless* untuk mengirim data dari sensor ke server (pengolah data). Modul *wireless* yang biasa digunakan pada jaringan sensor nirkabel antara lain ESP8266 dan Xbee. ESP8266 merupakan modul *WiFi* yang memiliki standar IEEE 802.11 b/g/n dengan frekuensi 2.4 GHz, sedangkan Xbee merupakan modul zigbee yang memiliki standar IEEE 802.15.4 dengan frekuensi 2.4 GHz. Seiring berkembangnya modul *wireless*, banyak modul yang digunakan untuk merancang jaringan sensor nirkabel antara lain Modul ESP8266(12E) dan Modul Xbee S2 [2].

C. Sistem pemantauan kualitas udara

Sistem pemantuan kualitas udara merupakan sistem yang dapat memonitoring kualitas udara secara *real time* pada suatu area / tempat, sistem ini biasanya bekerja dengan sebuah sensor yang dapat mendeteksi kandungan udara yang buruk seperti CO₂. Peralatan yang digunakan adalah arduino uno sebagai perangkat yang mengolah data dari sensor, dan sensor yang digunakan untuk mendekteksi kandungan gas adalah MQ135[4].

D. Quality of Service

Kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan bandwidth, mengatasi jitter dan delay. QoS merupakan kualitas atau jaminan terhadap layanan yang diberikan kepada pengguna jaringan. Dalam penelitan parameter QoS yang diuji adalah *Throughput*, *delay*, dan *packet loss* [5].

Throughput yaitu kecepatan(*rate*) transfer data efektif, yaitu di ukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut [5].

Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik kongesti atau juga waktu proses lama [5].

Packet loss adalah suatu parameter yang menggambarkan suatu kondisi jumlah total paket yang hilang pada suatu jaringan [5].

III. METODOLOGI PENELITIAN

A. Alat dan Bahan

Adapun perangkat keras dan perangkat lunak yang dibutuhkan sebagai berikut:

Perangkat Keras:

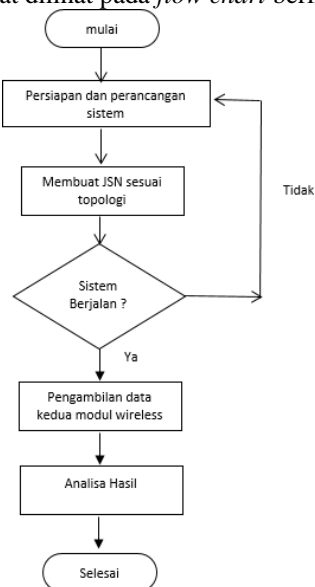
- Arduino uno R3 – 1 unit
- Xbee S2 – 1 unit
- Laptop - 2 Unit
- Xbee Shield – 1 unit
- Wemos d1 *retired*
- Meteran Gulung 30m-1unit

Perangkat Lunak:

- Arduino IDE 1.8.2
- X-CTU 6.3.5
- Xampp Control Panel 3.2.2
- Wireshark 2.4.5
- Docklight 1.6

B. Metode Penelitian

Metode penelitian yang dilakukan penulis dalam penelitian ini dapat dilihat pada *flow chart* berikut:

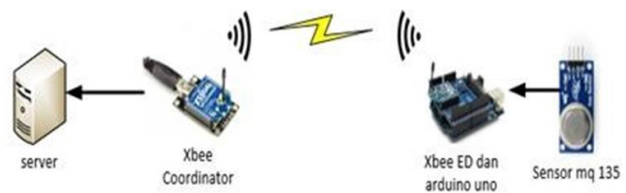


Gambar 1 Flowchart penelitian

C. Perancangan Topologi

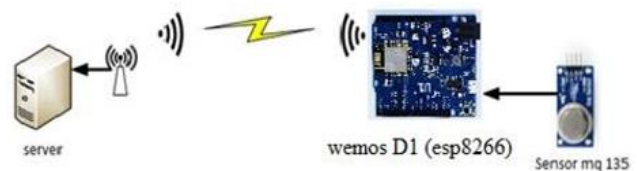
Topologi yang digunakan untuk menganalisa perbandingan kinerja modul *wireless* pada sistem pemantauan kualitas udara adalah *peer to peer* yang dibagi atas 2 skenario yaitu topologi menggunakan modul Xbee S2 dan ESP8266. Adapun skenario yang dimaksud disajikan pada Gambar 2 dan Gambar 3.

Pada konfigurasi perancangan topologi jaringan dengan menggunakan modul Xbee S2, untuk pengiriman data dari sensor menggunakan dua buah Xbee S2. Sensor mq 135 dihubungkan pada arduino yang telah terpasang Xbee shield dan Xbee End Device. Pada server terpasang Xbee Coordinator untuk menerima data yang dikirim dari sensor.



Gambar 2 Perancangan Topologi Jaringan Xbee S2

Pada konfigurasi perancangan topologi jaringan dengan menggunakan modul ESP8266 (12E), untuk pengiriman data dari sensor menggunakan jaringan Wi-fi yang terpasang pada wemos D1. Perangkat Wemos D1 dan Server akan terkoneksi pada jaringan Wi-Fi yang sama, sehingga kedua perangkat dapat berkomunikasi.

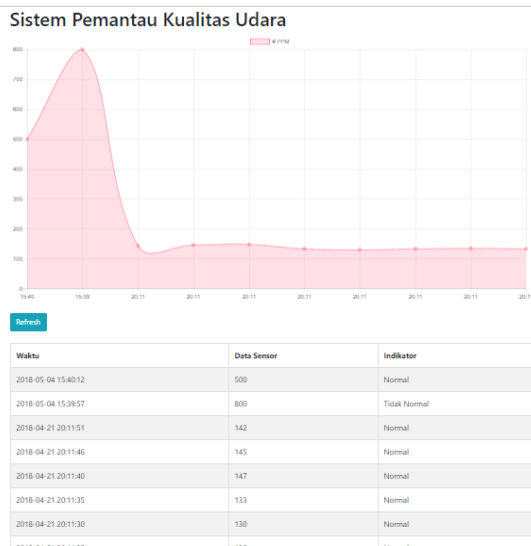


Gambar 3 Perancangan Topologi Jaringan ESP8266

IV. ANALISIS DAN PENGUJIAN

Analisi data dilakukan berdasarkan kumpulan data yang telah diambil berdasarkan parameter pada tiap skenario pengujian yang telah ditentukan. Nilai parameter disajikan dalam bentuk grafik kemudian dianalisis.

Sistem yang dibuat akan mengumpulkan informasi kandungan CO2 dari sensor MQ 135 yang kemudian akan dikirimkan ke server dengan menggunakan modul ESP8266(12e) dan Xbee S2. Sesampainya di *server*, data tersebut diteruskan ke *database* untuk disimpan. Data dari sensor yang sudah disimpan akan ditampilkan pada sebuah *web*. *Web* tersebut akan menampilkan grafik dan tabel yang datanya diambil dari *database* seperti yang terlihat pada Gambar 4.



Gambar 4 Tampilan Web

Grafik dan Tabel akan diperbarui secara otomatis bila ada data baru yang dikirim dari sensor, sehingga mempermudah untuk melihat perubahan nilai kandungan CO2 pada lingkungan sekitar sensor.

A. Pengujian Sistem

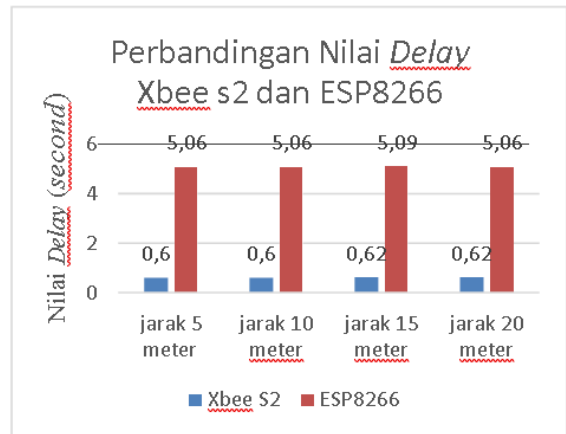
Pada tahap ini melakukan pengujian sistem untuk mengecek apakah sistem berjalan dengan baik atau tidak. Pengujian pertama dilakukan pada modul ESP8266 dengan melakukan tes ping pada IP modul ESP8266, hal ini untuk memastikan bahwa koneksi antara server dan modul ESP8266 pada kondisi yang bagus.

Pengujian kedua dilakukan pada modul Xbee S2 dengan menggunakan aplikasi XTCU untuk melihat komunikasi antara *Coordinator* dengan *End Device*.

Setelah melakukan Pengujian Sistem pada pengiriman data kedua modul Xbee S2 dan ESP8266, Selanjutnya melakukan Pengujian QoS berdasarkan parameter *delay*, *throughput*, dan *packet loss*. Nilai parameter akan dibanding pada sebuah Grafik kemudian dianalisis sehingga dapat dilihat perbandingan nilai QoS antara modul Xbee S2 dan ESP8266.

B. Pengujian Delay

Pada Gambar 6 merupakan hasil pengujian dari *delay* pengiriman data dengan menggunakan modul Xbee S2 dan modul ESP8266(12E).



Gambar 6 Perbandingan Nilai Delay Xbee s2 dan ESP8266

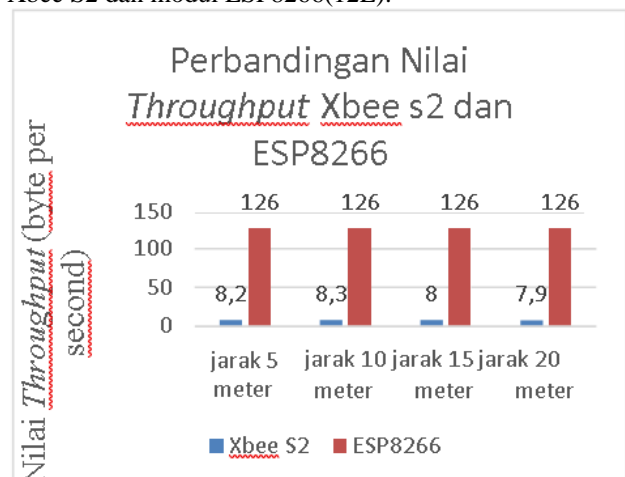
Nilai parameter *delay* yang didapat dari hasil pengujian *delay* pengiriman data dengan menggunakan modul Xbee S2 dan ESP8266(12E) sebagai berikut :

- Pada jarak 5 meter modul Xbee S2 mempunyai delay sebesar 0,6 detik, sedangkan modul ES8266(12E) mempunyai delay sebesar 5,06 detik.
- Pada jarak 10 meter modul Xbee S2 mempunyai delay sebesar 0,6 detik, sedangkan modul ES8266(12E) mempunyai delay sebesar 5,06 detik.
- Pada jarak 15 meter modul Xbee S2 mempunyai delay sebesar 0,62 detik, sedangkan modul ES8266(12E) mempunyai delay sebesar 5,09 detik.
- Pada jarak 20 meter modul Xbee S2 mempunyai delay sebesar 0,62 detik, sedangkan modul ES8266(12E) mempunyai delay sebesar 5,06 detik.

Dari data di atas membuktikan bahwa *delay* pengiriman Xbee lebih cepat dibandingkan dengan ESP8266 yang mempunyai delay lebih dari 5 detik. Baik Xbee dan ESP8266 mempunyai nilai *delay* pengiriman yang hampir sama, untuk Xbee sebesar 0,6 detik, sedangkan EPS8266 sebesar 5,09 detik.

C. Pengujian Throughput

Pada Gambar 7 menunjukkan bahwa hasil pengujian *throughput* pengiriman data dengan menggunakan modul Xbee S2 dan modul ESP8266(12E).



Gambar 7 Perbandingan Nilai *Throughput* Xbee s2 dan ESP8266

Nilai parameter *throughput* yang didapat dari hasil pengujian *throughput* pengiriman data dengan menggunakan modul Xbee S2 dan ESP8266(12E) sebagai berikut :

- Pada jarak 5 meter modul Xbee S2 mempunyai *throughput* sebesar 8,2 byte/s, sedangkan modul ES8266(12E) mempunyai *throughput* sebesar 126 byte/s.
- Pada jarak 10 meter modul Xbee S2 mempunyai *throughput* sebesar 8,3 byte/s naik 0,1 byte/s, sedangkan modul ES8266(12E) mempunyai *throughput* sebesar 126 byte/s.
- Pada jarak 15 meter modul Xbee S2 mempunyai *throughput* sebesar 8 byte/s turun sebesar 0,2 byte/s, sedangkan modul ES8266(12E) mempunyai *throughput* sebesar 126 byte/s.
- Pada jarak 20 meter modul Xbee S2 mempunyai *throughput* sebesar 7,9 byte/s turun sebesar 0,1 byte/s, sedangkan modul ES8266(12E) mempunyai *throughput* sebesar 126 byte/s.

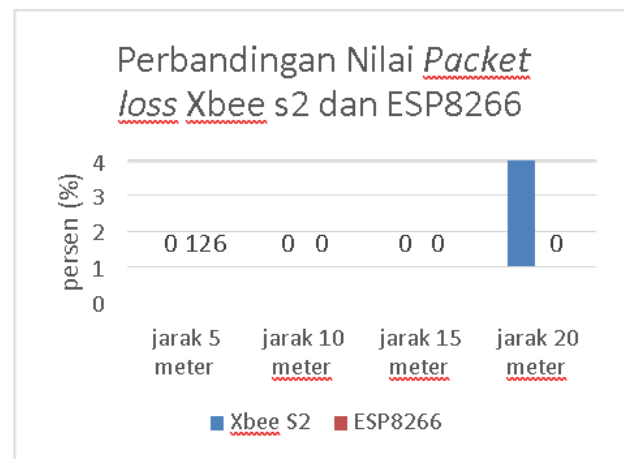
Dari data di atas membuktikan bahwa *throughput* pengiriman ESP8266 lebih besar dibandingkan dengan Xbee S2. *Throughput* pada Xbee S2 mempunyai nilai yang hampir sama yaitu sebesar 8 byte/s, sedangkan EPS8266 mempunyai nilai yang konsisten yaitu sebesar 126 byte/s.

D. Pengujian *Packet loss*

Pada Gambar 8 merupakan hasil pengujian *packet*



loss pengiriman data dengan modul Xbee S2 dan ESP8266.



Gambar 8 Perbandingan *Packet loss* Xbee s2 dan ESP8266

Nilai parameter *packet loss* yang didapat dari hasil pengujian *delay* pengiriman data dengan menggunakan modul Xbee S2 dan ESP8266(12E) sebagai berikut :

- Pada jarak 5 meter modul Xbee S2 dan ES8266(12E) mempunyai *packet loss* sebesar 0 %, hal ini membuktikan dengan pada jarak transmisi 5 meter, data dari sensor berhasil sampai ke server tanpa kehilangan paket data.
- Pada jarak 10 meter modul Xbee S2 dan ES8266(12E) mempunyai *packet loss* sebesar 0 %, hal ini membuktikan dengan pada jarak transmisi 10 meter, data dari sensor berhasil sampai ke server tanpa kehilangan paket data.
- Pada jarak 15 meter modul Xbee S2 dan ES8266(12E) mempunyai *packet loss* sebesar 0 %, hal ini membuktikan dengan pada jarak transmisi 15 meter, data dari sensor berhasil sampai ke server tanpa kehilangan paket data.
- Pada jarak 20 meter modul Xbee S2 mempunyai *packet loss* sebesar 3 %, sedangkan modul ES8266(12E) mempunyai *packet loss* sebesar 0 %.

Dari data di atas membuktikan bahwa pengiriman data dengan modul ESP8266 dan Xbee S2 mempunyai kategori *packet loss* yang bagus, meskipun pada modul Xbee terdapat *packet loss* sebesar 3 % pada jarak transmisi 20 meter [5].

E. Pengujian pengiriman data

Dalam pengujian dilakukan pengiriman data dari sensor secara bersamaan dalam waktu hampir 2 jam, Xbee dapat mengirimkan 1103 data, sedangkan ESP8266 dapat mengirimkan 1100 data. Hal ini dapat dilihat pada *database server* seperti yang ditunjukkan pada Gambar 8.

Gambar 9 jumlah data yang dikirim ESP8266 dan Xbee

V. PENUTUP

A. Kesimpulan

Berdasarkan hasil analisis pengujian terhadap QoS dari

sistem monitoring kualitas udara, maka dapat diambil kesimpulan sebagai berikut :

1. Parameter *Delay* pengiriman data dengan modul Xbee lebih cepat dari pada modul ESP8266.
2. Parameter *Throughput* pengiriman data dengan modul ESP8266 lebih cepat dari pada modul Xbee.
3. Parameter *Packet loss* pengiriman data dengan modul ESP8266 dan Xbee dalam kondisi bagus karena nilai *packet loss* tidak lebih dari 3 %.
4. Pengiriman data modul Xbee hanya mengirimkan data sensor, sedangkan ESP8266 selain mengirimkan data sensor juga dipengaruhi data lain seperti komunikasi antar jaringan.

B. Saran

Berikut adalah beberapa saran yang dapat digunakan untuk mengembangkan penelitian mengenai perbandingan kinerja antara modul Xbee dan ESP8266.

1. Melakukan perbandingan pengujian QoS pada area Indoor dan Out-door sehingga dapat menambah parameter pengujian untuk dianalisis.
2. Melakukan pengujian QoS pada sistem deteksi sensor lainnya seperti pada monitoring suhu, kelembapan udara, dan lainnya.

DAFTAR PUSTAKA

- [1] N. H. Dini, R. Munadi and A. Suryadi, "Analisis Performansi *Wireless Sensor Network* Dengan Mekanisme *Csma/Ca* Pada Standar Ieee 802.15.4/Zigbee," 2011.
- [2] T. Thaker, "ESP8266 based Implementation of *Wireless Sensor Network* with Linux Based Web-Server," IEEE, 2016.
- [3] Jusak, "Implementasi Zigbee Ieee 802.15.4 Untuk Pemantauan Suhu Dan Kelembaban Udara," Seminar Nasional Sistem & Teknologi Informasi (SNASTI) , 2013.
- [4] A. D. Nur, "Purwarupa Kendali Otomatis Kipas Pembuangan Berbasis Arduino Uno," 2014.
- [5] F. Z. Nasihin, A. B. P. Negara and A. Irwansyah, "Studi Perbandingan Performa QoS (Quality of Service) Tunneling Protocol PPTP Dan L2TP Pada Jaringan VPN Menggunakan Mikrotik," vol. 4, no. 1, 2015.

VISUALISASI DATA PENGGUNA LABORATORIUM VIRTUAL PADA KOMPONEN TELEMETERING OPENSTACK BERBASIS WEB MONITORING

Prajaka Brian Lizadi, Hidayat Nur Isnianto
Departemen Teknik Elektro dan Informatika Sekolah Vokasi
Universitas Gadjah Mada
prajaka.brian.l@mail.ugm.ac.id, hnisnianto@mail.ugm.ac.id

Abstract – Currently the use of Cloud computing to meet human necessity was become more common. Cloud computing more comfort provide given because of his model is placed in a container and allows easily accessible according to necessity in various locations. One of the Cloud computing platform that is widely started using are Openstack , because of the ability to manage and regulate the entire system through a web-based dashboard, the command line and through RESTful API. One of the supporting components computing system on a platform Openstack is a project that serves as a Ceilometer telemetering data to present the results of the monitoring, but it is still the raw data that require interface for communication data and human beings. This research created a system of data visualization as an interface to use by admin and users in monitoring data usage per instance on a virtual laborortium. The result from this research is web monitoring interface with differents graphic type and specification to engage admin requirement for analisis in virtual laboratory activity.

Keywords : Cloud computing, Openstack, Ceilometer, data visualization

Intisari – Saat ini penggunaan komputasi awan untuk memenuhi kebutuhan manusia sudah menjadi hal yang sangat umum. Komputasi awan dapat memberikan kenyamanan yang lebih karena modelnya yang ditempatkan dalam satu wadah dan memungkinkan untuk dapat diakses sesuai kebutuhan di berbagai lokasi. Salah satu platform komputasi awan yang mulai banyak digunakan adalah Openstack, karena memiliki kelebihan dalam mengelola dan mengatur keseluruhan sistemnya melalui dashboard berbasis web, perangkat perintah baris dan melalui RESTful API. Salah satu komponen pendukung sistem komputasi pada platform Openstack adalah Ceilometer yang berfungsi sebagai proyek telemetering untuk menyajikan data hasil monitoring, tetapi masih merupakan data mentah sehingga memerlukan interface untuk komunikasi data dan manusia. Pada penelitian ini dibuat suatu sistem visualisasi data sebagai interface untuk digunakan admin dan pengguna dalam memantau data penggunaan setiap instance pada laborortium virtual. Hasil yang diperoleh berupa tampilan halaman web monitoring dengan tipe grafik dan spesifikasi data berbeda – beda untuk memenuhi kebutuhan analisis admin dalam kegiatan laboratorium virtual

Kata kunci : Komputasi awan, Openstack, Ceilometer, Visualisasi data.

I. PENDAHULUAN

Perkembangan teknologi yang sangat pesat menyebabkan perubahan signifikan pada kebiasaan manusia. Kebiasaan yang dilakukan secara manual, secara bertahap dikembangkan agar dapat menjadi otomatis. Perwujudan tersebut adalah sesuatu yang diharapkan agar dapat meringankan kegiatan dalam lingkungan kerja. Penggambaran data dengan bentuk monitoring dalam suatu sistem kerja adalah salah satu otomatisasi yang dibentuk oleh manusia untuk memudahkan pengaturan dan analisa kegiatan dalam lingkungan kerja.

Dalam lingkungan kampus D-IV Teknologi Rekeyasa Internet, kebutuhan terhadap infrastruktur jaringan dalam kegiatan praktikum, adalah sesuatu yang seharusnya wajib untuk disediakan. Hal ini menunjukkan, bahwa kebutuhan terhadap ketersediaan komputasi awan pada lingkungan kampus semakin diperlukan karena memberikan kemudahan dalam mengelola infrastruktur jaringan secara menyeluruh. Salah satu layanan komputasi awan yang baru – baru ini populer digunakan adalah Openstack. Layanan Openstack menjadi platform yang mulai banyak digunakan karena memberikan fasilitas layanan yang aman, handal, fleksibel, *high-efficiency*, dan dapat dikembangkan [1]. Pada lingkungan kampus, salah satu keuntungan menggunakan platform Openstack adalah pengguna lebih fleksibel dalam membangun infrastruktur terkait kebutuhan

praktikum komputer jaringan [2]. Platform Openstack pada kegiatan praktikum bekerja dengan membentuk suatu *project* yang digunakan untuk membangun *instance virtual*, sehingga kegiatan praktikum dapat disebut sebagai

laboratorium *virtual* karena *instance virtual* dapat di akses dari semua jaringan internet UGM.

Sewajarnya, suatu perangkat yang dikembangkan oleh manusia pasti memiliki suatu kekurangan. Pada platform Openstack, kekurangan terdapat pada aplikasi metering yang belum memiliki kemampuan visualisasi data untuk keperluan analisis dan monitoring. Komponen *Ceilometer* pada platform Openstack dijelaskan telah menyajikan data – data monitoring [3], namun pada kenyataannya *Ceilometer* hanya bekerja sebatas pada metering data.

Dalam penelitian ini penulis memberikan solusi yaitu pengembangan komponen *Ceilometer* dengan mengubah data mentah pada aplikasi *metering* menjadi informasi berbentuk visual dalam sebuah halaman web monitoring. Halaman web monitoring akan dibangun dengan menggunakan bahasa *markup web* HTML, bahasa pemrograman PHP, dan *Javascript*. Dengan adanya pengembangan sistem visualisasi data dalam halaman *web* monitoring ini, admin dapat dengan mudah untuk menganalisis dan memonitor data dalam kegiatan laboratorium *virtual*. Selain hal tersebut, pengembangan ini diharapkan dapat mejadi solusi pada aplikasi *Ceilometer* yang belum menyajikan monitoring dari sistem kerja Openstack [4].

II. KAJIAN PUSTAKA

2.1. Komputasi Awan (Cloud computing)

Definisi sederhana dari *cloud computing* adalah suatu model yang memungkinkan jaringan dapat diakses dengan mudah sesuai kebutuhan di berbagai lokasi, dimana model ini dapat mengumpulkan sumber daya

komputasi seperti *network*, *server*, *storage*, aplikasi dan *services* dalam satu wadah.

Terdapat lima karakteristik penting dari *cloud computing* [5];

1. *On-demand self-service*. Pengguna dapat memesan dan mengelola layanan tanpa interaksi manusia dengan penyedia layanan, misalnya dengan menggunakan, sebuah portal *web* dan manajemen antarmuka. Pengadaan dan perlengkapan layanan serta sumberdaya yang terkait terjadi secara otomatis pada penyedia.
2. *Broad network access*. Kemampuan yang tersedia melalui jaringan dan diakses melalui mekanisme standar, yang mengenalkan penggunaan berbagai *platform* (misalnya, telepon selular, laptop, dan PDA).
3. *Resource pooling*. Penyatuan sumberdaya komputasi yang dimiliki penyedia untuk melayani beberapa konsumen menggunakan model multi-penyewa, dengan sumberdaya fisik dan *virtual* yang berbeda, ditetapkan secara dinamis dan ditugaskan sesuai dengan permintaan konsumen.
4. *Rapid elasticity*. Kemampuan dapat dengan cepat dan elastis ditetapkan.
5. *Measured Service*. Sistem komputasi awan secara otomatis mengawasi dan mengoptimalkan penggunaan sumberdaya dengan memanfaatkan kemampuan pengukuran (*metering*) pada beberapa tingkat yang sesuai dengan jenis layanan (misalnya, penyimpanan, pemrosesan, *bandwidth*, dan *account* pengguna aktif). Penggunaan sumberdaya dapat dipantau, dikendalikan, dan dilaporkan sebagai upaya memberikan transparansi bagi penyedia dan konsumen dari layanan yang digunakan.

2.2. Layanan Cloud computing

Pada *Cloud computing* terdapat tiga jenis model layanan dijelaskan sebagai berikut [5];

1. *Cloud Software as a Service (SaaS)*. Kemampuan yang diberikan kepada konsumen untuk hanya menggunakan aplikasi yang disediakan pada layanan *cloud*. Aplikasi tersebut dapat diakses dari berbagai perangkat client melalui antarmuka seperti *web browser* (misalnya, email berbasis *web*).
2. *Cloud Platform as a Service (PaaS)*. Kemampuan yang diberikan kepada konsumen untuk menjalankan aplikasi yang dibuat konsumen atau memperbolehkan konsumen untuk mengolah menggunakan bahasa pemrograman dan peralatan yang didukung oleh provider pada infrastruktur komputasi awan.
3. *Cloud Infrastructure as a Service (IaaS)*. Kemampuan yang diberikan kepada konsumen untuk memproses, menyimpan, dan melakukan komputasi sumberdaya pada layanan komputasi awan, dimana konsumen dapat menjalankan perangkat lunak secara bebas, dapat mencakup sistem operasi dan aplikasi.

2.2.1. Infrastructure as a Service (IaaS)

Secara definisi *IaaS* adalah layanan komputasi awan yang umumnya berupa seperangkat hardware komputer yang berupa “*virtualisasi*”, bersama dengan jaringan internet, *bandwidth*, dukungan alamat IP, keseimbangan beban, jaminan online

secara realtime (terus menerus koneksi ke internet) dan keamanan dalam “ruang lingkup” satu unit layanan *IaaS* [6].

Penerapan *Infrastructure as a Service (IaaS)* memiliki beberapa keuntungan, diantaranya sebagai berikut [7]:

1. Mudah dan cepat tersedia, dengan ketersediaan infrastruktur yang dapat dipesan sesuai kebutuhan.
2. Tidak perlu beli perangkat TI yang biasanya menguras biaya modal cukup besar, dan memakan waktu untuk instalasi serta konfigurasi.
3. Pengelolaan infrastruktur dilakukan oleh penyedia *IaaS*, sehingga hemat biaya profesional TI yang tergolong semakin mahal.
4. Dengan layanan infrastruktur (*IaaS*) anda dapat memperbesar dan mengurangi kapasitas *server* atau *storage* lebih cepat.
5. Hemat investasi awal bagi perusahaan startup ataupun bagi para developer aplikasi sebelum aplikasi tersebut dioperasikan secara normal atau dapat memperkecil biaya pra-operasional.

2.3. Openstack

Openstack adalah suatu proyek open source yang digunakan untuk *platform cloud computing*. Secara sederhana, Openstack merupakan *platform* untuk *cloud computing* yang berasal dari berbagai macam tipe *cloud*. Teknologi Openstack terdiri dari serangkaian aplikasi yang saling terkait dan setiap bagian dari aplikasi tersebut berfungsi mengontrol pengolahan, penyimpanan, serta jaringan sumber daya lintas pusat data. Openstack memiliki kemampuan dalam mengelola dan mengatur keseluruhan sistemnya melalui dashboard berbasis *web*, perangkat perintah baris dan melalui *RESTful API*.

2.3.1. Komponen Openstack

Terdapat komponen-komponen penting untuk mengimplementasikan Openstack agar terbentuk sebuah *cloud computing*;

1. Horizon (*Dashboard*)
2. Keystone (*Identity*)
3. Neutron (*Networking*)
4. Cinder (*Block Storage*)
5. Nova (*Compute*)
6. Glance (*Image*)
7. Swift (*Object Storage*)
8. Ceilometer (*metering*)
9. Heat (*Orchestration*)

2.3.2. Ceilometer

Proyek *Ceilometer* adalah layanan pengumpulan data yang menyediakan kemampuan untuk menormalisasi dan mengubah data di seluruh komponen inti Openstack. *Ceilometer* adalah komponen proyek *telemetering*. Datanya dapat digunakan untuk menyediakan pelanggan *billing*, *resource tracking*, *benchmarking*, *scalability*, dan *statistical* di seluruh semua *platform* Openstack.

Ada beberapa urutan pemahaman data *Ceilometer* pada Openstack yaitu;

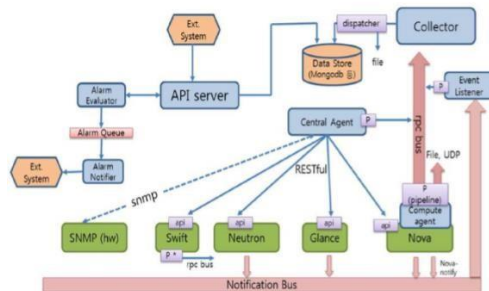
1. Meter. Hanya mengukur aspek tertentu dari penggunaan sumber daya atau kinerja yang

sedang berlangsung. Meter memiliki tiga tipe berbeda yaitu *cumulative*, *delta* and *gauge*. Jenis *cumulative* secara monoton meningkatkan nilai seperti nilai integrasi. *Delta* adalah perubahan nilai dari sebelumnya seperti diferensiasi nilai. *Gauge* adalah nilai yang berdiri sendiri yang hanya berkaitan dengan durasi saat ini.

2. Sample. Bekerja dengan menyederhanakan poin data individu yang berkaitan dengan pengukuran khusus. Sample memiliki atribut yang sama dengan Meter, juga ditambahkan timestamp dan pengukur nilai (disebut Volume).
3. Statistics. Nilai statistik agregasi selama durasi waktu yang ditentukan dan yang bukan seperti Sample adalah satu poin data. Terdapat 5 perbedaan fungsi agregasi (*count*, *max*, *min*, *avg*, *sum*). Artinya adalah jumlah Sample, nilai maksimal, nilai minimal, nilai rata-rata, jumlah total setiap periode.
4. Event. Memberikan informasi kepada pengguna kapan suatu objek berubah dalam Openstack. Sebagai contoh, ketika pengguna mengunduh *image*, pengguna membuat *Virtual Machine* dan ketika pengguna mengeksekusi perintah CRUD (*Create, Read, Update, Delete*) *REST*.

2.3.3. Struktur dan Komponen Ceilometer

Struktur dan komponen *Ceilometer* dapat kita lihat pada Gambar 2.1. terdapat 5 komponen dasar yang mendukung *Ceilometer* yaitu;



Gambar 2. 1 Struktur Komponen *Ceilometer*

1. *Collector*. Komponen penting yang terdapat pada *server* utama. *Notification messages* diproses dan di ubah menjadi data *metering message* yang kemudian dikirim kembali ke *message bus* dengan menggunakan topik yang sesuai. *metering message* secara langsung telah tercatat pada *Data Store* tanpa modifikasi. Pada awalnya, fungsi *pipeline* terletak pada *collector*, namun sekarang telah dipisah.
2. *Central agent*. Komponen yang terdapat pada *server* utama. Bekerja dengan memicu point dari polling. Terutama yang tidak berhubungan dengan fungsi komponen *Nova* yang telah digunakan seperti *Object Storage* (*Swift*), *image* (*Glance*). Akuisisi data ditransfer ke *collector* melalui *RPC Bus*.
3. *Compute agent*. Merupakan modul instalasi tambahan. Meskipun telah tercatat sebelum

push agent compute, *agent* hanya digunakan untuk billing. VM unit penghitungan *monitoring* (*CPU number*, *accumulative CPU time*, *CPU utilization*, *virtual NIC packet rate*, *disk and memory statistics characteristic value*) sangatlah penting. Akan tetapi, *Compute agent* tidak mendukung dimana metode *push agent* digunakan. *Libvirt (default)*, *hyper-V*, dan *Vmware* tersedia pada hypervisor.

4. *Data Store*. Penyimpanan menggunakan *default* *MongoDB*. *PostgreSQL*, *HBase*, *IBM DB2* sedang dalam proses pengembangan.
5. *API server*. *Web server* yang terletak pada *server* pusat. Untuk balasan dari yang dibutuhkan oleh *API*, digunakan mekanisme *RESTful*. *Pecan server* sangat ringan digunakan sebagai *web framework* bahasa pemrograman *python*. *Pecan server* mendukung *REST-style controller*, penggunaan bahasa dan mendukung *JSON*.

2.6. Visualisasi Data

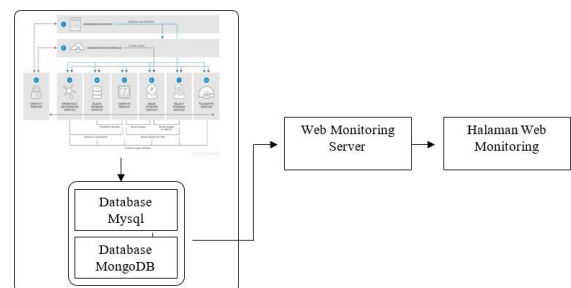
Tujuan utama dari visualisasi data adalah untuk menjembatani informasi secara jelas dan efisien kepada pengguna lewat grafik informasi yang dipilih, seperti tabel dan grafik. Visualisasi yang efektif membantu pengguna dalam menganalisa dan penalaran tentang data dan bukti. Ia membuat data yang kompleks bisa di akses, dipahami dan berguna. Pengguna bisa melakukan pekerjaan analisis tertentu, seperti melakukan perbandingan atau memahami kausalitas, dan prinsip perancangan dari grafik (contohnya, memperlihatkan perbandingan atau kausalitas) mengikuti pekerjaan tersebut. Tabel pada umumnya digunakan saat pengguna akan melihat ukuran tertentu dari sebuah variabel, sementara grafik dari berbagai tipe digunakan untuk melihat pola atau keterkaitan dalam data untuk satu atau lebih variabel [7].

III. METODOLOGI PENELITIAN

Tahap awal pembuatan sistem ini adalah dengan merancang topologi. Topologi yang diterapkan pada sistem ini adalah seperti Gambar 3.1 dan Gambar 3.2.



Gambar 3. 1 Topologi Perangkat



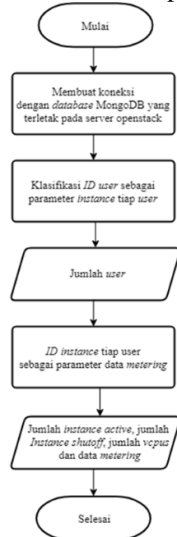
Gambar 3. 2 Topologi Proses Sistem

Platform *Openstack* memiliki aplikasi pendukung untuk *metering* data pengguna. Data hasil *telemetry* akan

dikumpulkan dan dicatat kedalam *database* MongoDB. Kumpulan dari data tersebut kemudian akan di visualisasikan kedalam *web monitoring* berbentuk grafik.

Web monitoring yang akan digunakan menggunakan sistem operasi CentOS dan aplikasi pendukung (*Apache, PHP, Javascript, PHP MongoDB driver*). Ditempatkan pada alamat jaringan yang sama.

Gambar 3.3. merupakan alur dari sistem untuk menghasilkan visualisasasi data berupa grafik.



Gambar 3. 3 Diagram Alir Sistem

IV. ANALISIS DAN PENGUJIAN

4.1. Informasi Data pada Dashboard

Pada halaman dashboard digambarkan jumlah total data dari *user*, core CPU fisik, *instance*, VCPU, VCPU yang digunakan, dan VCPU yang bisa digunakan. Pada Gambar

4.1 terlihat informasi pada total *user* berisi 12 *user* yang terdaftar. Informasi tersebut dapat dibuktikan dengan melihat pada halaman *web* Openstack yang menampilkan informasi data *project* atau *user*.



Gambar 4. 1 Informasi Total Data Instance

Pada Gambar 4.2, urutan daftar yang ada dalam halaman *web* Openstack menunjukkan jumlah total *user* sebanyak 12. Pengujian ini membuktikan informasi jumlah Total *User* sesuai dengan informasi pada Gambar 4.1.

Gambar 4. 2 Halaman Dashboard Openstack

Total VM *active* dan *Shutdown* adalah informasi berbentuk teks dinamis yang selanjutnya akan dijelaskan. Informasi teks dinamis akan berubah sesuai dengan kondisi *instance*. Sesuai dengan Gambar 4.1, terdapat *instance* yang menunjukkan informasi keadaan *active* dan *shutdown*.

Pada Gambar 4.3, dapat dilihat bahwa data yang sedang *active* dan *shutdown*. Pada bagian ini di ambil contoh pada *instance* dengan nama windows dan win – server yang sedang dalam keadaan *shutdown*. Data tersebut terhitung dalam jumlah total data *instance* VM *shutdown* pada Gambar 4.1.

ID Instance	Instance Name	Type	VCPU	Last Update	Status
000001	m1.medium	2		2018-05-19 01:23:34	active
001-b1	m1.medium	2		2018-05-19 01:23:34	active
001-b3	m1.medium	2		2018-05-19 01:23:34	active
publisher-1	m1.small	1		2018-05-19 01:23:34	active
net	m1.medium	2		2018-05-19 01:23:34	active
kebab-guest1	m1.medium	2		2018-05-19 01:23:33	shutdown
001-a4	m1.medium	2		2018-05-19 01:23:34	active
001-a5	m1.medium	2		2018-05-19 01:23:34	active
win10	m1.large	4		2018-05-19 01:23:33	shutdown
001-g1	m1.medium	2		2018-05-19 01:23:34	active
webdev-jobu	m1.large	4		2018-05-19 01:23:33	active
001-b2	m1.medium	2		2018-05-19 01:23:34	active
Windows	m1.large	4		2018-05-19 01:23:33	shutdown
win-server	m1.large	4		2018-05-19 01:23:33	shutdown
centos	m1.large	4		2018-05-19 01:23:34	active

Gambar 4. 3 Daftar Keadaan Instance

Instance pada Gambar 4.3, dapat dibuktikan dalam keadaan *shutdown* dengan melihat Gambar 4.4 pada halaman Openstack berikut.

Gambar 4. 4 Halaman Daftar Instance Pada Dashboard Openstack

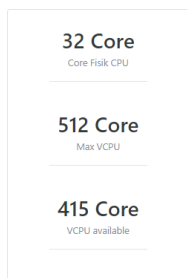
Informasi Total VCPU adalah jumlah dari core CPU yang digunakan oleh tiap *instance*. Data tersebut dibagi menjadi 4 tipe, yaitu adalah *x1.small*, *x1.medium*, *x1.large* dan *x1.xlarge*. Tipe - tipe tersebut mewakili dari penggunaan core yang dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Daftar Tipe Instance

Tipe	Core VCPU	Disk	Memory
<i>x1.small</i>	1	20 GB	2048 MB
<i>x1.medium</i>	2	40 GB	4096 MB
<i>x1.large</i>	4	80 GB	8192 MB
<i>x1.xlarge</i>	8	160 GB	16384 MB

Total VCPU pada Gambar 4.2 adalah hasil dari penjumlahan semua jumlah core dari setiap *instance* ID.

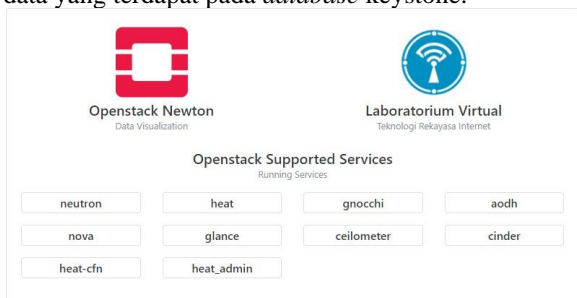
Server yang digunakan oleh *platform* Openstack pada penelitian proyek akhir ini memiliki spesifikasi core sebesar 32 core. Data tersebut, dapat digunakan sebagai parameter untuk menentukan jumlah banyak nya VCPU yang masih dapat digunakan. Informasi akan ditampilkan dengan data teks dinamis seperti Gambar 4.5.



Gambar 4. 5 Informasi Total Core Fisik dan VCPU

4.2. Informasi Aplikasi Pendukung

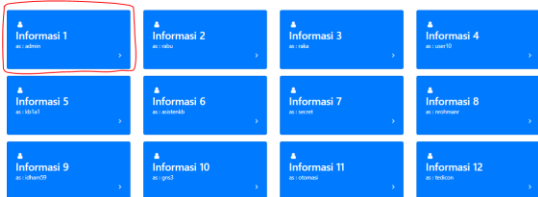
Halaman ini memberikan informasi tentang aplikasi yang mendukung berjalanya sistem dalam platform Openstack. Data *running service* pada Gambar 4.6, adalah teks dinamis yang sesuai dengan data yang terdapat pada *database* keystone.



Gambar 4. 6 Informasi *Running Service*

4.3. Informasi Pada Halaman Pengguna

Jumlah total *user* seperti yang ditampilkan oleh Gambar 4.1, pada halaman ini akan ditampilkan secara deskriptif dengan menampilkan nama dan urutan. Pengguna tersebut ditampilkan menggunakan *card-body* yang diberikan fungsi untuk menghubungkan data dengan halaman data *instance*. Sebagai contoh, pada bagian ini digunakan *user* admin untuk mengakses halaman informasi data *instance* yang terdaftar pada *user* admin. Halaman *user* menampilkan informasi seperti pada Gambar 4.7. Informasi 1 seperti yang dilingkari pada gambar adalah perwakilan dari *user* admin.

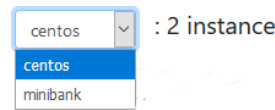


Gambar 4. 7 Informasi Data *User*

4.4. Informasi Data Instance

Halaman data *instance* digunakan untuk menampilkan informasi dari data tiap *instance* yang telah dibuat pada masing – masing *user*. Seperti yang sudah dijelaskan sebelumnya, informasi yang terdapat pada halaman *user* digunakan untuk menghubungkan data kehalaman data *instance*. Dapat disebut juga bahwa data ID *user* digunakan sebagai parameter untuk mengakses data tiap *instance*. Setiap *user* dapat membangun *instance* lebih dari satu sampai batas core VCPU habis digunakan. Oleh karena itu, dibangunlah sebuah fungsi on change dengan dropdown pada halaman *web* untuk memilih *instance* dan

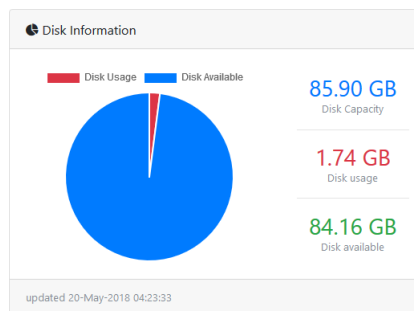
melihat informasi data *instance* yang dipilih. Tampilan tabel dropdown, pada bagian ini mengambil contoh dari *user* dengan nama raka. Informasi tersebut berbentuk seperti Gambar 4.8.



Gambar 4. 8 Informasi Dropdown *Instance*

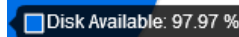
Data *instance* dalam dropdown, ketika dipilih akan menampilkan informasi data yang sudah diklasifikasikan sesuai daftar *instance*. Daftar data yang ditampilkan memiliki daftar seperti pada Lampiran 1. Pada proyek akhir ini, data yang akan ditampilkan pada halaman *web* hanya mencakup data yang tertera pada batasan masalah. Pemilihan data yang akan ditampilkan berdasarkan perbandingan penggunaan data untuk kebutuhan analisis dan *diskusi* terkait data yang lebih penting untuk ditampilkan. Data tersebut ditampilkan dalam beberapa grafik berebeda.

Data *disk* pada halaman ini akan di bagi menjadi 3 bentuk informasi dalam sebuah grafik pie. Sebagai pengujian, digunakan data *instance* centos dari data salah satu *user* bernama raka. *Instance* centos memiliki spesifikasi dengan tipe *x1.large*. Pada Gambar 4.9, bahwa warna merah mewakili jumlah data yang digunakan oleh *instance* centos, kemudian warna biru pada grafik mewakili jumlah *disk* yang masih tersedia.



Gambar 4. 9 Grafik *Disk Information*

Data yang terdapat pada grafik ditampilkan pada fungsi pop up dalam persen. Penampilan data tersebut seperti pada Gambar 4.10 dan 4.11.



Gambar 4. 10 Pop Up *Disk Available*



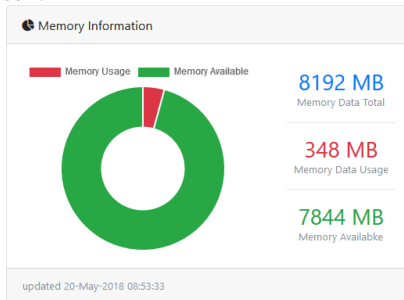
Gambar 4. 11 Pop Up *Disk Usage*

Informasi teks yang terletak disebelah grafik adalah teks dinamis yang menunjukkan data dalam bentuk satuan GB. Data teks tersebut, adalah informasi yang ditampilkan berdasarkan satuan yang digunakan oleh data persen pada grafik. Informasi yang ditampilkan dari data tersebut akan menyesuaikan dengan parameter

nama dan ID pada teks.

Kesimpulan dari informasi diatas, dapat dikatakan bahwa *instance* centos pada *user* raka, pada 20 Mei 2018 pukul 04:23:33 sudah menggunakan *disk* 2.03% atau sebesar 1.74 GB dan masih menyediakan *disk* 97.97% atau sebesar 85.90 GB.

Informasi data *memory*, pada dasarnya memiliki kesamaan tujuan informasi dengan data *disk*. Perbedaannya terletak pada informasi data yang ditampilkan. Pada Gambar 4.12, terlihat pada grafik bahwa warna merah mewakili dari data *memory* usage, kemudian warna hijau mewakili data *memory* available. Informasi data pada grafik juga ditampilkan dalam bentuk persen.



Gambar 4. 12 Grafik Memory Information

Data *instance* dalam dropdown, ketika dipilih akan menampilkan informasi data yang sudah diklasifikasikan sesuai daftar *instance*. Daftar data yang ditampilkan memiliki daftar seperti pada Lampiran 1. Pada proyek akhir ini, data yang akan ditampilkan pada halaman *web* hanya mencakup data yang tertera pada batasan masalah. Pemilihan data yang akan ditampilkan berdasarkan perbandingan penggunaan data untuk kebutuhan analisis dan diskusi terkait data yang lebih penting untuk Data persen dari grafik dapat dilihat seperti pada Gambar 4.13 dan Gambar 4.14.



Gambar 4. 13 Pop Up Memory Usage

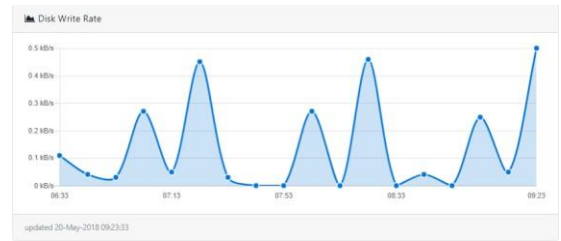


Gambar 4. 14 Pop Up Memory Available

Kesimpulan yang dapat diambil dari informasi yang ditampilkan adalah pada 20 Mei 2018 pukul 08:53:33 data *memory usage* menunjukkan 4.25% atau penggunaan data *memory* sebesar 348 MB dan *memory available* menunjukkan 95.75% atau jumlah *memory* yang tersedia sebesar 7844 MB.

Informasi *disk write bytes rate* dan *disk read bytes rate* pada halaman ini ditampilkan dalam grafik dengan tipe line. Data tersebut adalah hasil mengukur tingkat rata – rata kegiatan membaca dan menulis *disk* dalam rentang waktu 10 menit. Waktu 10 menit ditentukan oleh aplikasi *telemetry* pada platform *Opstack*. Gambar 4.15 menunjukkan informasi dari *instance* centos dalam *user* dengan nama raka. Grafik line di atas menjelaskan informasi tentang *disk write rate* bytes. Data tersebut diperbaharui terakhir pada 20 Mei 2018 pukul 09.23.33. Grafik tersebut memiliki rentang waktu pada label selama

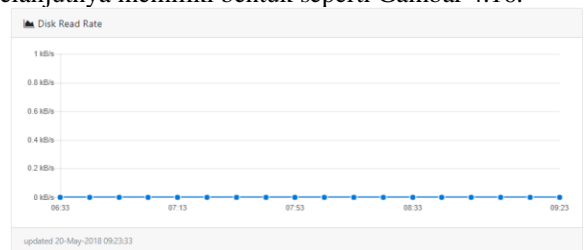
3 jam. Data yang ditampilkan adalah pada saat *instance* centos tidak melakukan aktifitas.



Gambar 4. 15 Informasi Grafik Disk Write Bytes Rate

Kuota yang terdapat disamping grafik, secara dinamis akan mengikuti data terbesar yang ditampilkan dalam grafik. Pada grafik diatas, data terbesar menunjukkan 0.5 kB/s. Jadi data dari rentang waktu 09.13 sampai dengan 09.23, rata-rata *disk* menulis data sebesar 0.5 kB/s.

Grafik *disk read bytes rate* yang akan dijelaskan selanjutnya memiliki bentuk seperti Gambar 4.16.



Gambar 4. 16 Informasi Grafik Disk Read Bytes Rate

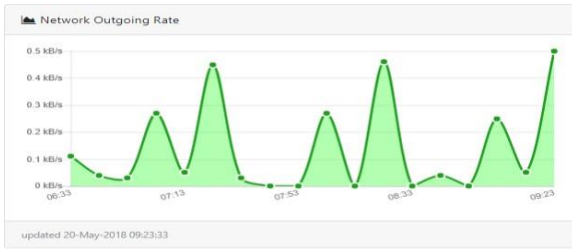
Tidak berbeda dari penjelasan pada *disk write bytes rate*. Pada grafik *disk read bytes rate* data yang tampak memiliki rata – rata sebesar 0 kB/s, kejadian tersebut dikarenakan tidak adanya proses membaca data dalam kurun waktu 3 jam terakhir. Data terakhir diperbaharui pada 20 Mei 2018 pukul 09.23.33.

Dalam grafik ditentukan rentang waktu grafik selama 3 jam. Hal ini dikarenakan dalam kebutuhan analisis kerja efektif pada laboratorium *virtual* adalah selama 2-3 jam.

Informasi yang ditunjukkan *network incoming* dan *outgoing bytes rate* memiliki pendekatan yang sama dengan penjelasan *disk write / read bytes rate*. Tidak ada perbedaan cara menampilkan data, hanya saja fungsi dari data tersebut memiliki tujuan berbeda. Informasi dari *network incoming / outgoing rate* berbentuk seperti Gambar 4.17 dan Gambar 4.18.



Gambar 4. 17 Informasi Grafik Network Incoming Rate



Gambar 4. 18 Informasi Grafik *Network outgoing Rate*

Pengujian menggunakan data *instance* yang sama yaitu centos. Pada saat *instance* tidak melakukan aktifitas, data dari *network incoming* dan *outgoing* menampilkan informasi data yang sama. Dari grafik tersebut, dapat disimpulkan pada *instance* centos, rata – rata data *network* yang masuk terbesar dalam waktu 3 jam terdapat pada rentang waktu 09.13 sampai 09.23 dengan menunjukkan informasi sebesar 0.5 kB/s . Kemudian rata – rata data yang keluar adalah sebesar 0.5 kB/s pada rentang waktu yang sama.

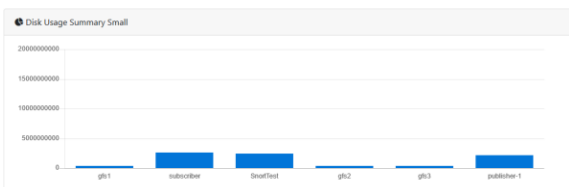
Informasi yang disajikan teks dinamis pada halaman ini ditampilkan serperti pada Gambar 4.19.



Gambar 4. 19 Informasi Data Teks Dinamis

Setiap data yang ditampilkan pada Gambar 4.20 memiliki fungsi tujuan penjelasan sesuai dengan nama yang tertera. Data yang ditampilkan terakhir diperbaharui pada 20 Mei 2018 pukul 11.03.33. Pengujian pada bagian ini dilakukan pada *instance* centos. Informasi CPU Utility menunjukkan waktu penggunaan CPU sebesar 0.03%, kemudian CPU Time *user* sebesar 27 minute, lalu CPU *Delta* sebesar 0.80 second dan Status pada *instance* adalah *active*.

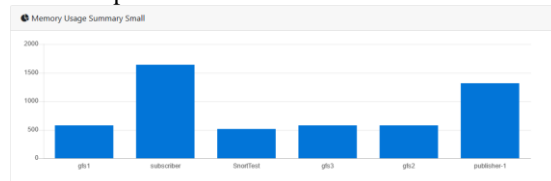
Summary data atau ringkasan data pada proyek akhir ini ditempatkan dalam satu halaman dengan nama halaman *summary* data. Halaman ini dapat di akses melalui Navigation Bar dengan nama *summary* data. Seperti namanya *summary* data adalah ringkasan semua data *instance* dalam satu halaman *web*. Ringkasan tersebut mencakup data – data yang sudah dijelaskan pada tahapan penelitian. Pada halaman ini pengujian dilakukan dengan menampilkan semua informasi dari tiap *instance* yang terdaftar. Informasi tersebut dapat dilihat pada Gambar 4.20.



Gambar 4. 20 Informasi Grafik *Summary Data Disk*

Gambar 4.20 adalah contoh grafik yang digunakan untuk *summary data disk*. *Instance* yang baru dibentuk akan secara otomatis terdaftar pada grafik sesuai dengan tipe *instance*. Grafik yang digunakan adalah grafik dengan tipe bar. Setiap tipe *instance* yang berada dalam grafik, telah dibatasi sesuai dengan kapasitas, sehingga ketika grafik bar sudah menyentuh angka akhir dari kuota yang sudah ditentukan, maka *disk* dari *instance* tersebut dapat dinyatakan penuh. Data *disk* setiap *instance* menggunakan satuan kB dengan alasan agar data dapat ditampilkan sedetail mungkin.

Informasi berikutnya adalah data *memory* yang akan dibentuk seperti Gambar 4.21.



Gambar 4. 21 Informasi Grafik *Summary Data Memory*

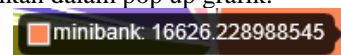
Seperti halnya dengan grafik data *disk*, grafik data *memory* juga memiliki data dengan tipe *instance*, sehingga data *instance* dikelompokkan sesuai dengan tipe *instance*. Gambar 4.21 adalah contoh dari grafik *summary data memory* Data *memory* setiap waktunya dapat berubah – ubah sesuai dengan penggunaan *memory instance*. Grafik yang ditampilkan memiliki data bertipe MB dan total kapasitas di tulis pada kuota. Jika data mencapai batas kuota maka artinya penggunaan *memory* dapat dikatakan penuh.

Informasi ringkasan data selanjutnya adalah pada data *disk write / read bytes rate* dan *network incoming / outgoing rate*. Ringkasan data dibuat dengan menggunakan 2 grafik yaitu grafik pie dan grafik line. Grafik pie digunakan untuk menampilkan data terbaru setiap *instance* yang memiliki kapasitas rata – rata *disk* membaca dan menulis terbesar. Bentuk grafik tersebut seperti pada Gambar 4.22.



Gambar 4. 22 Informasi Grafik *Summary Data Disk Write and Read*

Grafik pie pada Gambar 4.22 menunjukkan sejumlah *instance* ID yang diwakilkan oleh tiap warna. Pada kedua grafik, terlihat warna yang mendominasi grafik pie tersebut. Warna yang mendominasi tersebut adalah rata – rata *disk* yang memiliki tingkat membaca dan menulis paling besar dari semua *disk*. Data *instance* tersebut adalah data pada Gambar 4.23 dan Gambar 4.34 yang ditampilkan dalam pop up grafik.



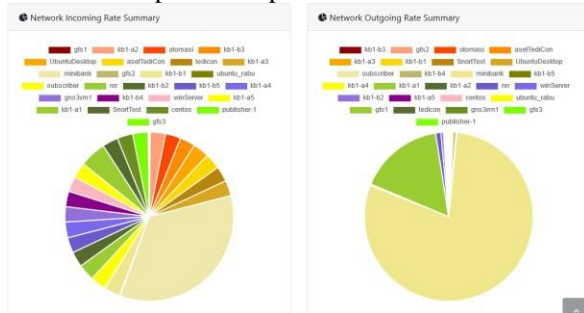
Gambar 4. 23 Pop Up Data *Disk Write* Terbesar



Gambar 4. 24 Pop UP Data *Disk Read* Terbesar

Penggunaan *resource disk write* dan *read rate bytes* terbesar adalah ditempati oleh *instance minibank* dan *kb1- a1*. Data yang ditampilkan menggunakan satuan B/s agar data dapat dianalisa secara detail.

Pada informasi ringkasan *network incoming / outgoing rate bytes* pada dasar nya memiliki fungsi grafik dan tujuan yang sama dengan yang dijelaskan pada ringkasan data *disk write / read rate bytes*. Grafik data tersebut dapat dilihat pada Gambar 4.25.

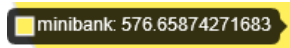


Gambar 4. 25 Grafik Informasi *Network Incoming* and *Outgoing*

Dari grafik tersebut terlihat beberapa warna yang mendominasi. Warna tersebut mewakili dari data yang ditunjukkan Gambar 4.26 dan Gambar 4.27.



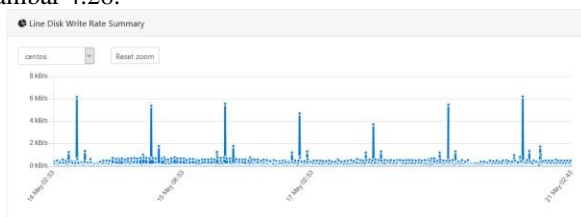
Gambar 4. 26 Pop Up Data *Network Incoming Rate* Terbesar



Gambar 4. 27 Pop Up Data *Network Outgoing* Terbesar

Instance minibank adalah *instance* yang memiliki penggunaan *network incoming* dan *outgoing* terbesar. Data yang ditampilkan menggunakan satuan B/s, agar data dapat dianalisa secara detail. Pengujian pengiriman data

Informasi ringkasan data *disk write / read* dan *network incoming / outgoing* menggunakan grafik dengan tipe line yang digunakan untuk menampilkan ringkasan data dalam kurun waktu 1 minggu. Data tersebut ditampilkan sebagai record data untuk dapat memenuhi kebutuhan analisis. Grafik tersebut ditampilkankan seperti pada Gambar 4.28.

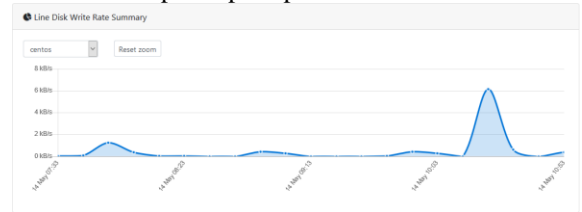


Gambar 4. 28 Informasi Grafik Line *Summary Data Disk Write Rate*

Grafik tersebut memiliki kegunaan yang sama dalam tujuan menampilkan data. Grafik tersebut diberikan

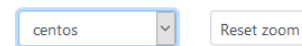
fungsi *zoom in* dan *zoom out* untuk dapat menampilkan data secara detail. Sebagai contoh, pada bagian ini akan digunakan *instance centos* sebagai data yang dipilih untuk dilihat secara detail. Data yang di inginkan adalah data *disk write bytes rate* terletak pada tanggal 14 Mei 2018 pukul

07.33 sampai 10.53 maka pada grafik setelah dilakukan *zoom in* akan tampak seperti pada Gambar 4.29.



Gambar 4. 29 Informasi Data *Summary* Setelah *Zoom In*

Terlihat pada Gambar 4.29, data yang berada pada line chart hanya pada rentan waktu sesuai yang di inginkan. Pada grafik line tersebut juga ditambahkan beberapa fungsi *on change* pada menu dropdown untuk memilih data *instance* dan *button reset* untk mereset grafik agar kembali seperti semula setelah proses *zoom in*. fungsi dropdown dan reset tersebut berbentuk seperti Gambar 4.30.



Gambar 4. 30 Menu Dropdown dan Menu Reset

Pada halaman ini, informasi yang disajikan berupa ringkasan data detail dalam bentuk tabel. Halaman ini digunakan untuk menampilkan masing – masing *instance* yang terdaftar dan keadaan *instance* sesuai dengan waktu diperbaharui. Informasi data yang tertera pada Gambar 4.31 diharapkan dapat digunakan untuk membantu tahap awal sebelum kegiatan analisis.

Instance Name	Type	VCPU	Last Update	Status
gfs1	m1.small	1	2018-05-21 03:03:35	active
asetfedCon	m1.large	8	2018-05-21 03:03:35	active
wireServer	m1.large	4	2018-05-21 03:03:35	active
UbuntuDesktop	m1.medium	2	2018-05-21 03:03:35	active
gns3m1	m1.medium	2	2018-05-21 03:03:35	active
subscriber	m1.small	1	2018-05-21 03:03:35	active
tedcon	m1.large	8	2018-05-21 03:03:35	active
SmartTest	m1.small	1	2018-05-21 03:03:35	active
minibank	m1.large	4	2018-05-21 03:03:35	active

Gambar 4. 31 Daftar Informasi Data Halaman Tabel

Kesimpulan yang dapat diberikan adalah *instance* yang dibangun memiliki tipe spesifikasi *instance* yang berbeda sesuai dengan kebutuhan penggunaan *instance*. Data akan ditampilkan dan diperbaharui setiap 10 menit sekali, dengan tipe grafik dan fungsi analisis yang sesuai dengan data. Setiap informasi yang ditampilkan diharapkan dapat digunakan untuk analisis data dalam kegiatan laboratorium *virtual* atau penelitian yang menggunakan *platfotm* Openstack

DAFTAR PUSTAKA

[1] C. Shao, B. Liang, F. Wang, H. Deng, W. Dai, S. Wei, X. Zhang and Z. Yuan, "Open Stack Platform and its Application in BigData Processing," 2015 8th

- International Conference on Intelligent Networks and Intelligent Systems, pp. 98-101, 2015.
- [2] R. Y. Faisela, PENGEMBANGAN LAYANAN INFRASTRUCTURE AS A SERVICE (IAAS) UNTUK LABORATORIUM *VIRTUAL* MENGGUNAKAN PLATFORM ORCHESTRATION PADA OPENSTACK, Yogyakarta: Universitas Gadjah Mada, 2017.
- [3] E. Qevani, M. Panagopoulou, C. Stampoltas, A. Tsitsipas, D. Kyriazis and M. Themistocleous, "What can OpenStack adopt from a Ganeti-based open-source IaaS?," 2014 IEEE International Conference on Cloud Computing, pp. 833-840, 2014.
- [4] B. Dongmyoung and L. Bumchul, "Analysis of Telemetry Service in OpenStack," ICTC 2015, pp. 272-274, 2015.
- [5] A. Syaikh, "KOMPUTASI AWAN (CLOUD COMPUTING) PERPUSTAKAAN PERTANIAN," Jurnal Pustakawan Indonesia Volume 10 No. 1, pp. 1-12, 2010.
- [6] D. Cloud, "Pengertian Infrastructure As a Service (IaaS)," 6 juni 2016. [Online]. Available: <https://cloud.datacomm.co.id/blog/pengertian-iaas/>.
- [7] FlyingEagle, "Pengertian IaaS – Infrastructure as a Service," 24 Mei 2016. [Online]. Available: <http://www.mobnasesemka.com/pengertian-iaas/>.