EFISIENSI *FLOODING* DENGAN PENGEMBANGAN METODE *RELIABLIE SUBNETWORK* PADA ARSITEKTUR *FIBBING* DI LINGKUNGAN *HYBRID* SDN

Dino Budi Prakoso[1)], Royyana Muslim Ijtihadie [2)], Tohari Ahmad[3)]
[1, 2, 3)] Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember
Jalan Teknik Kimia, ITS, Sukolilo, Surabaya, 60111
e-mail: dinobudiprakoso@gmail.com[1)], roy@if.its.ac.id[2)], tohari@if.its.ac.id[3)]

ABSTRAK

Dalam dunia teknologi khususnya pada bidang jaringan saat ini konektivitas autonomous systems (AS) sangat dibutuhkan. Khususnya terhadap protokol routing dynamic yang sering dipakai dibandingkan protokol routing static. Untuk mendukung jaringan saat ini, dibutuhkan protokol routing yang efisien dan efektif yang mampu mencakup skala yang cukup besar. Software Defined Network (SDN) adalah inovasi teknologi pada dunia jaringan yang mempunyai Control Plane dan Data Plane terpisah yang memudahkan dalam melakukan konfigurasi pada sisi Control Plane. Control Plane merupakan titik pusat terjadinya proses bottleneck dalam arsitektur SDN. Performance merupakan masalah kritis dalam implementasi jaringan skala besar karena beban permintaan yang besar terjadi pada Control Plane dengan menghasilkan nilai throughput yang rendah. Pada penelitian ini akan dilakukan pengujian pada jaringan Hybrid SDN dengan menggunakan protokol routing OSPF, berdasarkan arsitektur Fibbing yang diimplementasikan pada jaringan sistem Hybrid SDN mampu membantu dalam meningkatkan performance, namun terdapat kendala saat pengiriman flooding LSA Type-5 yang digunakan sebagai pembentuk fake node. Banyak node-node yang tidak dilewati sebagai jalur distribusi dalam pembentukan fake node, dalam hal ini tentu akan berdampak pada nilai throughput menjadi tidak stabil dan menurun. Hal tersebut dapat diatasi dengan menggunakan metode Isolation Domain untuk mengatur efisiensi flooding LSA Type-5.

Kata Kunci: *ospf*, *quality of service, reliable subnetwork, software defined network.*

EFFICIENCY OF FLOODING BY DEVELOPING RELIABLE SUBNETWORK METHODS ON FIBBING ARCHITECTURE IN THE HYBRID ENVIRONMENT SDN

Dino Budi Prakoso[1)], Royyana Muslim Ijtihadie[2)], Tohari Ahmad[3)]
[1, 2, 3)] Departement of Informatics, Institut Teknologi Sepuluh Nopember
Jalan Teknik Kimia, ITS, Sukolilo, Surabaya, 60111
e-mail: dinobudiprakoso@gmail.com[1)], roy@if.its.ac.id[2)], tohari@if.its.ac.id[3)]

ABSTRACT

In the technology world especially in the field of current network of Autonomous Systems connectivity (AS) is indispensable. Especially against the dynamic routing protocols that are often used compared to static routing protocols. In supporting this current network, it takes efficient and effective routing protocols capable of covering a sizable scale. Software Defined Network (SDN) is a technological innovation in the network world that has a separate Control Plane and Data Plane that makes it easy to configure on the Control Plane side. Control Plane is the focal point on a process of bottleneck in SDN architecture. Performance is a critical issue in large-scale network implementations because of the large demand load occurring in the Control Plane by generating low throughput value. This research will be conducted testing on the Hybrid network of SDN by using OSPF routing protocol, based on the Fibbing architecture implemented on the system network Hybrid SDN also able to assist in improving performance, but there are constraints when sending flooding which is used as a fake node forming. Many nodes are not skipped as distribution lines in the formation of a fake node, in which case it will certainly affect the value of throughput to be unstable and decrease. This can be overcome by using the Isolation Domain method to manage the LSA Type-5 flooding efficiency.

Keywords: *ospf*, *quality of service, reliable subnetwork, software defined network.*

I. PENDAHULUAN

Pada perkembangan dalam dunia modern komunikasi pada jaringan yang berbasis pada distribusi kontrol dan protokol jaringan transportasi mengalami banyak permasalahan yang cukup kompleks. Seperti penggunaan jaringan IP yang tradisional telah di gunakan secara umum pun belum bisa menangani permasalah yang kompleks dan rumit. Peningkatan permintaan aplikasi secara *real-time* pun cukup membuat kesulitan dalam

mencakup jaringan yang telah ada.

Diusulkan bahwa SDN [1] sangat cocok untuk menjadi pusat data, karena dapat mengembangkan jaringan luas terhadap perusahaan-perusahaan yang berfokus pada teknologi secara global. Dalam klasifikasi *Network Management* terdapat 3 bagan yaitu *Legacy*, SDN dan *Hybrid* SDN [2]. *Legacy* merupakan model jaringan lama yang jarang sekali dipakai saat ini dan tidak termasuk dalam protokol TCP/IP. SDN adalah pendekatan terbaru dalam membangun, mendesain dan mengelola jaringan komputer.

Hybrid SDN mengacu pada paradigma *centralized* dan *decentralized* berdiri berdampingan dan melakukan komunikasi bersama pada berbagai aspek untuk melakukan konfigurasi, mengubah, mengkontrol dan mengelola perilaku jaringan. untuk meningkatkan kinerja dalam satu arsitektur jaringan tersebut. Penggunaan jaringan pada *Hybrid SDN* juga memberikan keuntungan dari beberapa hal diantaranya adalah mudah dalam mengkonfiguraasi *forwading paths* secara deklaratif dan global, mengatur *forwading paths* secara *direct control* dan *installation path* distribusi dengan respon cepat [3]. Untuk membuat suatu jaringan *Hybrid* pada SDN diperlukan konektivitas antara komponen SDN dan *Legacy* salah satunya dengan menggunakan RouteFlow [4]. RouteFlow adalah *open-source* yang berfungsi sebagai penyedia layanan *IP-Routing* secara virtualisasi yang terhubung dengan komponen SDN. RouteFlow tersusun dari beberapa bagian yakni, RouteFlow *server*, OpenFlow Controller (e.g., RYU) dan jaringan virtual yang memberikan konektivitas dan mengoperasikan *IP-Routing* (e.g., Quagga). *IP-Routing* akan memberikan informasi yang berisikan *Forwading Information Base* (FIB) sesuai dengan konfigurasi yang telah dibuat pada *routing protocol* (e.g., OSPF).

Arsitektur *Fibbing* [5] merupakan arsitektur yang menawarkan fleksibilitas pada jaringan konvensional, tetapi dari hal fleksibilitas tersebut perlu mengorbankan tingkat kekuatan *resource* pada protokol terdistribusi. Yang dilakukan arsitektur Fibbing dalam hal fleksibilitas dalam jaringan adalah membuat *fake node* dan *fake link* yang digunakan sebagai memperpendek atau mempercepat jalur pengiriman informasi dari *source node* hinggai *destination node*. Dengan membuat dan memperkenalkan *node* dan *link* yang dibuat secara *virtual* (*fake node*) pada protokol routing berbasis *linkstate*, didapatkan penghitungan dan pengelompokkan perutean pada topologi yang mencakup skala besar. Pada saat *fake node* terbentuk untuk membantu dalam pengelompokkan dan mengumpulkan informasi routing tabel akan cukup efisiensi dalam menanggulangi *flooding* pada saat routing protokol OSPF mengirimkan LSA pada jaringan tersebut. Dalam pembuatan *fake node* pada jaringan akan mempengaruhi kinerja dari protokol OSPF [6]. *Fake node* yang terbentuk dianggap kurang efektif, karena *fake node* yang terbentuk tidak hanya pada jalur yang dilewati, tetapi jalur-jalur yang tidak dilewati akan membuat *fake node* dan *fake link* yang akan berdampak pada nilai *throughput.* Dalam menjalankan arsitektur Fibbing pada lingkungan Hybrid SDN diperlukan RouteFlow yang bertujuan menjembatani jaringan konvesional dan jaringan SDN. RouteFlow akan membantu dalam mensimulasikan jaringan konvesional pada jaringan SDN agar arsitektur Fibbing bisa berjalan pada lingkungan Hybrid SDN.

Dengan beberapa penjelasan tentang terbentuknya *fake node* pada arsitektur Fibbing membuat penulis untuk melakukan pengembangan terhadap Reliable Subnetwork dengan memperhitungkan nilai *metric Cost* sebagai pembentuk area subnetwork baru dalam mengurangi *flooding*. Dengan melakukan uji coba dan analisa menggunakan topologi Mesh [7] yang akan diimplementasikan secara Hybrid dengan menggunakan arsitektur Fibbing. Untuk melakukan simulasi digunakan Mininet dan RouteFlow sebagai pembentuk skenario pada Hybrid SDN. Untuk *controller* pada SDN dalam penelitian ini menggunakan RYU Controller [8].

Dalam makalah ini terdari dari Pendahuluan yang berisikan tentang pengenalan tentang SDN, arsitektur Fibbing dan RouteFlow. Selanjutnya pada Studi Literatur yang berisikan tentang pengembangan dan analisis dari penelitian-penelitian sebelumnya yang berkaitan dengan topik penelitian ini. Kemudian pada Metode yang Diusulkan berisikan tentang penjelasan metode yang diusulkan pada penelitian ini. Selanjutnya untuk Arsitektur dan Topologi yang menjelaskan tentang penjelasan rancangan dan topologi jaringan yang digunakan dalam penelitian ini. Hasil dan Kesimpulan menjelaskan evaluasi dari simulasi yang telah dilakukan dari penelitian ini dan memberikan penjelasan secara ringkas dari hasil evaluasi tersebut.

II. STUDI LITERATUR

Dalam beberapa tahun terakhir pada dunia teknologi di bidang jaringan mengalami perkembangan yang sangat pesat dan paradigma baru pun muncul pada bidang jaringan yaitu, *Software Defined Network* (SDN) [9]. Penelitian yang membahas tentang pengenalan *Software Defined Network* telah dilakukan. Marcin dkk [10] pada tahun 2016 mengusulkan penggunaan *Software Defined Network* (SDN) sebagai solusi yang sangat berguna untuk masa depan. Penelitian yang membahas tentang Hybrid SDN telah dilakukan oleh Vanbever dan Visscchio [11] pada tahun 2014 mengusulkan suatu gagasan untuk pusat control dapat melakukan *distributed routing computation* melalui node palsu (*fake node*).

Kemudian dalam penelitian selanjutnya Visscchio dkk [12] di tahun yang sama mengajuk *central controlled* yang dinamakan "*Fibbing*", dengan memberikan fleksibilitas dalam routing jaringan, seperti *load balancing*, *traffic steering*, dan *back up path.* Dengan memanipulasi input pada protokol jaringan konvensional, manipulasi yang

dilakukan dengan membuat node palsu (*fake node*) di jaringan melalui pembuatan *Link State Advertisement* (LSA) palsu. "*Fibbing*" mengambil beberapa masukan seperti (i) *Path requirements* dari operator jaringan (ii) topologi jaringan dan (iii) graph acyclic untuk setiap tujuan. Berdasarkan masukan yang diambil, LSA Type-5 akan di suntikkan (*injects*) untuk memperkenalkan node palsu dalam topologi jaringan dan akan melakukan *advertising* kepada tujuan. Akan tetapi saat *controller* "*Fibbing"* menyuntikkan node palsu dengam menyebarkan LSA Type-5 pada jalur (*link*) yang tidak digunakan, akan mengakibatkan nilai troughput menjadi lebih banyak [13].

Reliable Subnetwork (RSN) menjadi salah satu hal untuk mengurangi LSA *flooding* [14]. Beberapa penelitian tentang metode *Reliable Subnetwork* telah dilakukan sebelumnya. Miyamura dkk [15] menjelaskan permasalahan tentang skalabilitas pada routing protokol OSPF dan IS-IS untuk jaringan skala besar. Miyamura dkk mengusulkan skema untuk mengurangi *overhead messages* dalam protokol routing tersebut dan melakukan pendekatan dasar dalam menanggulangi *overhead* tersebut dengan membatasi jumlah *neighbors* dimana informasi link-state akan didistribusikan. Hasil dari penelitian tersebut didapatkan jumlah LSA yang dikirim terlalu banyak oleh setiap *router* telah mengalami pengurangan sehingga memberikan *reliability of flooding*.

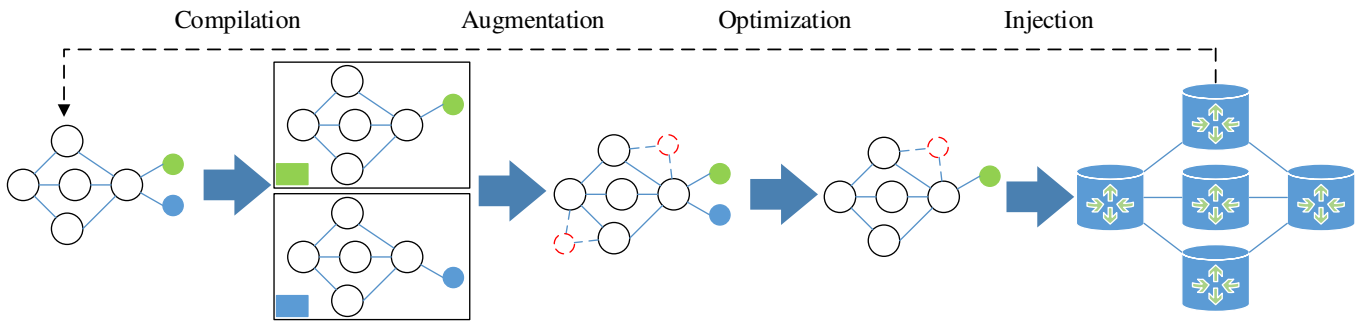
Kemudian pada penelitian Lee SH dkk [16] mengusulkan metode *isolation group* dalam mengatur proses *flooding* agar efisien. Dengan menggunakan arsitektur jaringan *Spanning tree* [17] dan mekanisme *Isolation group* yang akan dibandingkan RSN algorithm untuk mekanisme dalam membatasi *flooding* ke subset pada topologi jaringan. Penelitian menunjukkan hasil baik dengan peningkatan 20% pada rata-rata LSA *traffic* dibandingkan dengan hasil RSN.

III. METODE YANG DIUSULKAN

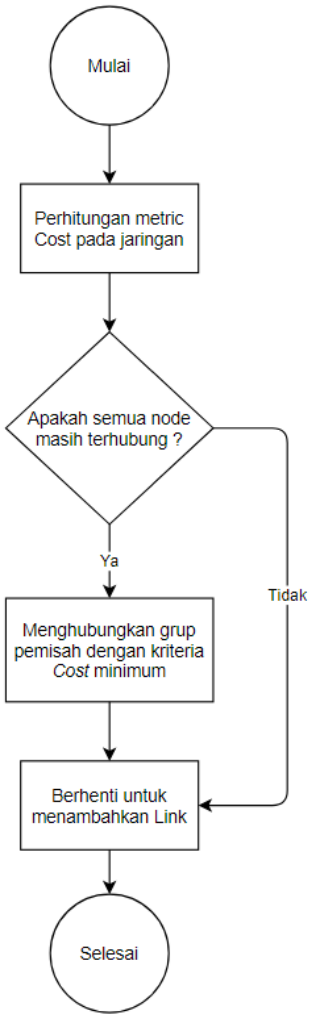
Dalam penelitian ini, arsitektur Fibbing pada tahapan *Augmentation* akan dilakukan modifikasi dengan konsep *Isolation Domain*. Pada tahapan *Augmentation* arsitektur Fibbing akan melakukan penambahan *fake node* dan *fake link* untuk menentukan jalur tercepat dan terpendek seperti pada Gambar 1. Pada saat penambahan *fake node* dan *link node* akan terjadi perhitungan *Cost* dalam penentuan jarak. Jika *Cost* pada *link* yang dihasilkan lebih rendah dibandingkan *Cost* pada *link* yang lain maka *Cost* dengan nilai lebih rendah akan dipilih sebagai jalur sebenarnya.

Proses implementasi arsitektur Fibbing akan dilakukan pada sisi *controller* pada SDN. *Model Control Plane* yang akan digunakan adalah *Hybrid* Control. Sistem kinerja arsitektur Fibbing pada *routing protocol* OSPF akan diujikan terlebih dahulu pada topologi jaringan. Setelah proses implementasi arsitektur Fibbing dengan routing protocol OSPF berjalan di lingkungan sistem *Hybrid* SDN. Pada saat Fibbing melakukan proses *Augmented Topology*, merupakan proses penyebaran *LSA Type-5* untuk membentuk *fake node* secara global. *Isolation Domain* adalah sebuah konsep yang digunakan untuk mengatur penambahan *fake node* dan *fake link* pada jaringan. *Isolation Domain* ini merupakan representasi dari metode *Reliable Subnetwork* (RSN) yang di modifikasi untuk mengatasi terjadinya *flooding* pada routing protokol OSPF.

RSN akan menjadi proses verifikasi untuk penentuan *fake nodes* tersebut. *Fake nodes* yang terbuat hanya rute dari *source* hingga *destination*. RSN didefinisikan sebagai subset atau bagian dari jaringan dengan *cost* minimal yang berisikan semua node pada jaringan yang asli. Jika pada jaringan yanag bukan dari bagian RSN dan terjadi beberapa kegagalan pada rute yang mempunyai nilai *cost* minimal, maka jaringan akan dibagi menjadi beberapa bagian. Beberapa jalur atau rute akan menjadi tidak dapat tersinkronisasi dan tidak dapat menyimpan topologi jaringan kedalam database pada *router* yang lain didalam jaringan tersebut. Mekanisme untuk membatasi terjadinya *flooding* dapat dilihat pada diagram Gambar 2. Sebelum menjelaskan lebih mendalam tentang topologi yang digunakan dalam penelitian ini, sedikit menjelaskan tentang arsitektur yang akan digunakan untuk membangun jaringan SDN dengan *dynamic routing*. RouteFlow akan dijalankan pada virtual yang berbeda dengan virtual untuk jaringan SDN.



Gambar 1. Tahapan pada arsitektur Fibbing [5].



Gambar 2. Diagram Mekanisme *Isolation Domain.*

IV. ARSITEKTUR DAN TOPOLOGI

A. Rancang Bangun SDN

Untuk menjalankan rancang bangun pada penelitian SDN, sebelumnya perlu mendefininsikan SDN terlebih dahulu untuk menjalankan protokol routing. Mininet merupakan *emulator* yang difungsikan untuk mensimulasikan SDN yang dijalankan dengan menggunakan *operating system* yang berbasis Linux. Dalam hal penelitian ini akan menggunakan *operating sytem* Ubuntu pada untuk menjalankan Mininet.

Untuk menjalankan *routing protocol* pada tiap-tiap *device* yang terhubung dengan jaringan SDN diperlukan virtualisasi tambahan dengan menggunakan Quagga. Quagga akan berjalan pada virtualisasi dengan RouteFlow agar setiap *routers* bisa terhubung dengan *controller*. Virtualisasi Quagga akan disimulasikan dengan *operating system* Ubuntu. Pada arsitektur ini juga dibutuhkan satu switch untuk menjembatani komunikasi yang akan mendistribusikan *packet* melalui controller SDN dalam simulasi menggunakan Mininet. *Controller* yang terhubung menggunakan RYU Controller.

B. Topologi

Setelah menjelaskan tentang rancang bangun dalam penelitian ini, gambaran topologi juga diperlukan untuk menampilkan secara visualisasi tentang cara bekerja dan cara komunikasi antara *device legacy* dengan simulasi SDN. Pada Gambar 3, dapat dilihat beberapa router yang terhubung satu sama lain dan router yang terhubung dengan satu switch yang juga terhubung dengan controller. Pembuatan topologi jaringan Mesh ini mempunyai *Spanning Tree* yang akan mendukung dalam pengujian *Isolation Domain* pada arsitektur Fibbing. *Spanning Tree* merupakan sistem untuk menemukan *link* redundant (cadangan) secara dinamis.

Pada topologi jaringan ini, akan dijalankan dengan beberapa skenario diantaranya:

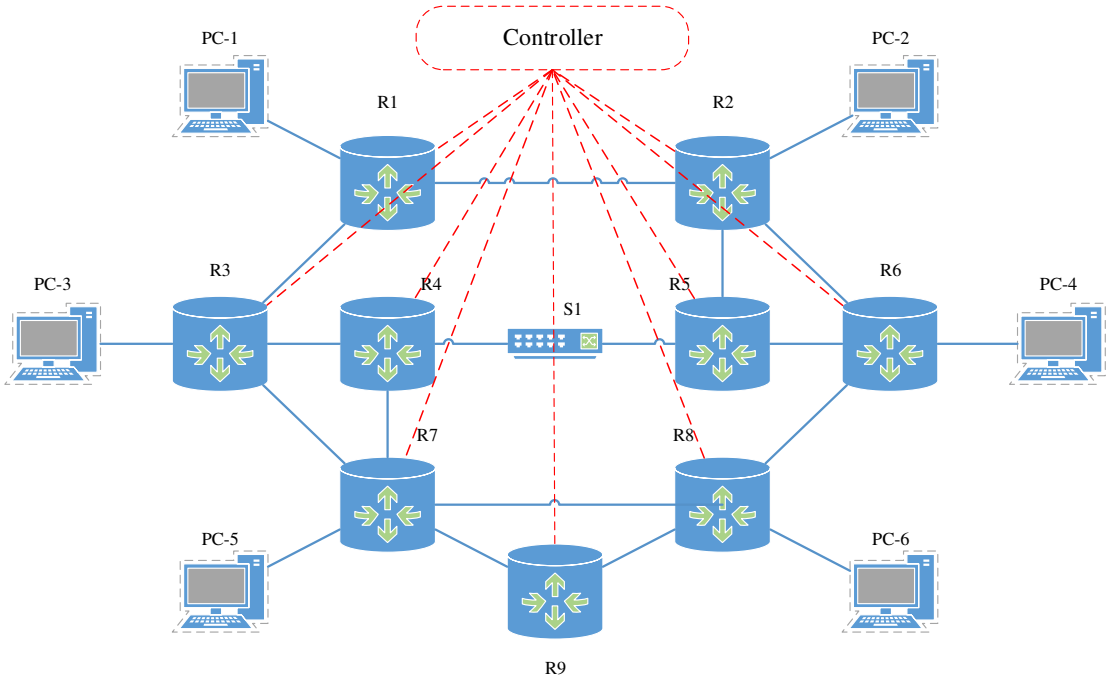
1) Arsitektur Fibbing akan dijalankan menggunakan topologi pada Gambar 3.
2) Dengan menjalankan 2 skenario yaitu skenario pertama menjalankan arsitektur Fibbing dan skenario kedua menjalankan arsitektur Fibbing dengan metode *Isolation Domain*.

3) Dimasing-masing skenario akan dilakukan pengujian *flooding* paket untuk mendapatkan hasil efisiensi *flooding*.

V. HASIL DAN PEMBAHASAN

Hasil dari pengujian pada penelitian ini dilaksanakan dan menjalankan skenario yang telah dibuat dengan menggunakan dua virtualisasi menggunakan Routeflow (Quagga) dan Mininet. Pada virtualisasi Routeflow (Quagga) akan di jalankan skenario dari arsitektur Fibbing, sedangkan untuk virtualisasi Mininet dijalankan guna menjalankan topologi yang ditunjukkan pada Gambar 3 dan untuk menjalankan skenario untuk pengujian. Untuk parameter simulasi yang di jalankan pada virtualisasi Routeflow (Quagga) ditunjukkan pada Tabel I.

Ryu Controller yang terhubung pada jaringan akan disimulasikan menggunakan Routeflow dan akan dihubungkan pada masing-masing router melalui Quagga. Jalur komunikasi antar router yang akan digunakan pada Quagga menggunakan routing protokol OSPF. Quagga akan di jalankan di tiap-tiap router yang telah dibuat pada topologi di Mininet. Parameter simulasi untuk penggunaan Mininet dapat dilihat pada Tabel II dan Python *script* untuk mengimplementasi topologi pada Mininet. Secara singkat virtualisasi untuk menjalankan arsitektur Fibbing ini melalui Quagga dan akan diteruskan melalui Routeflow guna agar bisa berkomunikasi dengan SDN *device*. Skenario yang akan dilakukan pertama kali adalah mengimplementasikan arsitektur Fibbing menggunakan Routeflow dan Ryu Controller yang akan terhubung dengan topologi pada Mininet. Untuk memastikan RouteFlow berjalan pada Mininet dengan menggunakan perintah *pingall* pada Mininet.



Gambar 3. Topologi Mesh.

TABEL I
PARAMETER SIMULASI ROUTEFLOW DAN QUAGGA.

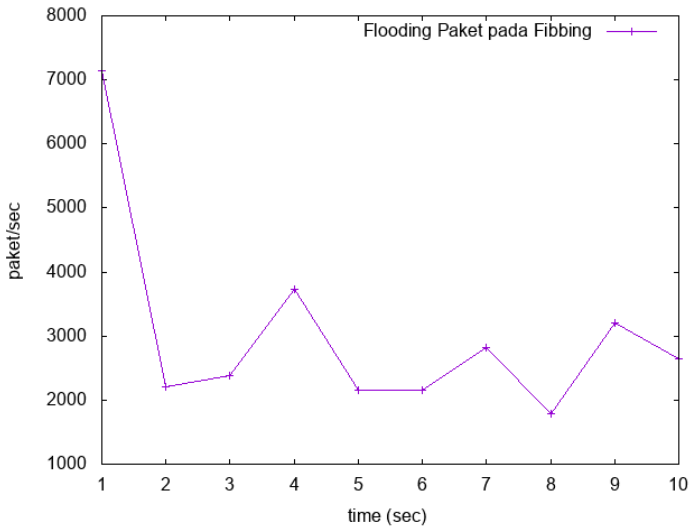
No	Paramater	Spesifikasi
1	Sistem Operasi	Linux Ubuntu 12.04
2	Kebutuhan Memori untuk Simulasi	4 GB
3	Central Processing Unit	Core i7-7700HQ
4	Peralatan Simulasi	VMware Workstation 12
5	Routing Protocol	Open Shortest Path First (OSPF)
6	Controller pada RouteFlow	Ryu Controller

TABEL II
PARAMETER SIMULASI MININET.

No	Paramater	Spesifikasi
1	Sistem Operasi	Linux Ubuntu 14.04
2	Kebutuhan Memori untuk Simulasi	4 GB
3	Central Processing Unit	Core i7-7700HQ
4	Peralatan Simulasi	VMware Workstation 12
5	Total Router	9
6	Total Switch	1
7	Total Personal Computer	6

TABEL III
TOTAL PAKET FLOODING SKENARIO ARSITEKTUR FIBBING.

Waktu	Node	Jenis Paket	Paket yang Terkirim
00.00 – 01.00			7136
01.00 – 02.00	R-1	LSA Type 5	2206
02.00 – 03.00			2386
03.00 – 04.00			3729
04.00 – 05.00	R-2	LSA Type 1	2162
05.00 – 06.00			2151
06.00 – 07.00	R-6	LSA Type 1	2813
07.00 – 08.00			1788
08.00 – 09.00	R-8	LSA Type 1	3197
09.00 – 10.00			2640
Total Paket			30208



Gambar 5. Grafik *flooding packets* pada arsitektur Fibbing.

A. Hasil Efisiensi *Flooding* pada Arsitektur Fibbing

Untuk pengujian akan diberikan *source node* dan *destination node* dengan node source pada PC-1 dan node destination pada PC-6. Uji coba ini dilakukan untuk melihat kinerja arsitektur Fibbing dan pemilihan jalur untuk pengujian dalam topologi yang diajukan, kemudian hasil dari pengujian akan ditampilkan dalam bentuk grafik. Pada Gambar 4 dapat dilihat alur pemilihan jalur yang berjalan pada arsitektur Fibbing. Jalur yang digunakan merupakan jalur terpendek dari node Source (Pc-1) menuju node Destination (Pc-6), dengan melalui area *subnetwork* pada R-1 dan R-2. Setelah perpindahan area subnetwork, informasi terkirim secara lokal melalui R-2 menuju R-6 kemudian R-6 menuju R-8. Dengan durasi waktu pengujian selama 10 detik dan dengan total paket yang terkirim 30208 paket. Paket LSA type 5 terkirim melalu R-1 menuju R-2 sebanyak 11.728 paket selama durasi waktu mulai 00.00 hingga 03.00 detik. Setelah melalu area subnetwork, dapat dilihat pada tabel 4.1, R-2 hingga R-8 mengirimkan paket LSA type 1 sebanyak 18.480 paket dengan durasi waktu mulai 03.00 hingga 10.00. Pengiriman LSA type 1 ini merupakan paket yang dikirimkan oleh router-router yang tergabung dengan area yang sama.

Dengan durasi waktu pengujian selama 10 detik dan dengan total paket yang terkirim 30208 paket. Paket LSA type 5 terkirim melalu R-1 menuju R-2 sebanyak 11.728 paket selama durasi waktu mulai 00.00 hingga 03.00 detik. Setelah melalu area subnetwork, dapat dilihat pada tabel III, R-2 hingga R-8 mengirimkan paket LSA type 1 sebanyak 18.480 paket dengan durasi waktu mulai 03.00 hingga 10.00.

Pengiriman LSA type 1 ini merupakan paket yang dikirimkan oleh router-router yang tergabung dengan area yang sama. Grafik yang ditunjukkan pada Gambar 5, telah menunjukkan hasil *flooding test* dari metode arsitektur Fibbing. Dengan pengujian selama 10 detik dan mengirimkan paket dengan total 30208 ke seluruh jaringan. Dengan pengiriman terbanyak pada saat awal dengan total 7136 paket. Pengiriman paket terbanyak pada awal pengiriman merupakan proses untuk mendapatkan seluruh informasi *forwarding table* pada topologi jaringan. Setelah mendapatkan informasi tersebut, proses pengiriman akan mengalami pengurangan paket karena proses pengiriman paket akan diperbarui dengan *forwarding table* dengan informasi *fake links* dan *fake nodes* yang terbentuk dapat dilihat pada detik 2 dan detik 4 pada Gambar 5.

Paket LSA type 5 yang dikirimkan pada awal seperti pada Tabel III, merupakan proses pengiriman informasi yang telah melewati area yang berbeda dalam satu jaringan. Paket LSA type 5 ini dikirimkan kepada dengan total paket yang banyak karena menyimpan semua informasi perangkat yang terhubung pada area di dalam topologi

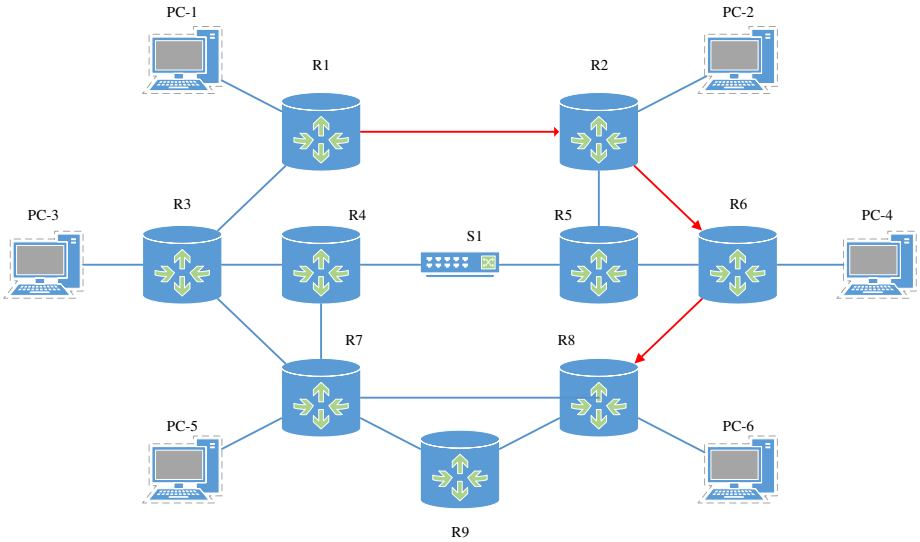
jaringan. Pada saat melakukan pengujian dengan skenario pada arsitektur Fibbing, pemilihan jalur tercepat terpilih tanpa memperhitungkan perbedaan area dari subnetwork dalam satu jaringan, akan tetapi pemilihan jalur tercepat berdasarkan jarak terpendek yang ditentukan dari *source node* hingga *destination node*.

Dari analisa tersebut menunjukkan bahwa pengiriman paket terbanyak pada saat awal pengujian merupakan proses disaat pembentukan fake nodes dan fake links. Dan selanjutnya setelah proses pembentukan *fake nodes* dan *fake links* selesai maka mekanisme untuk Injection pada arsitektur Fibbing bekerja sebagai proses pengiriman packet dari *source* menuju *destination*.

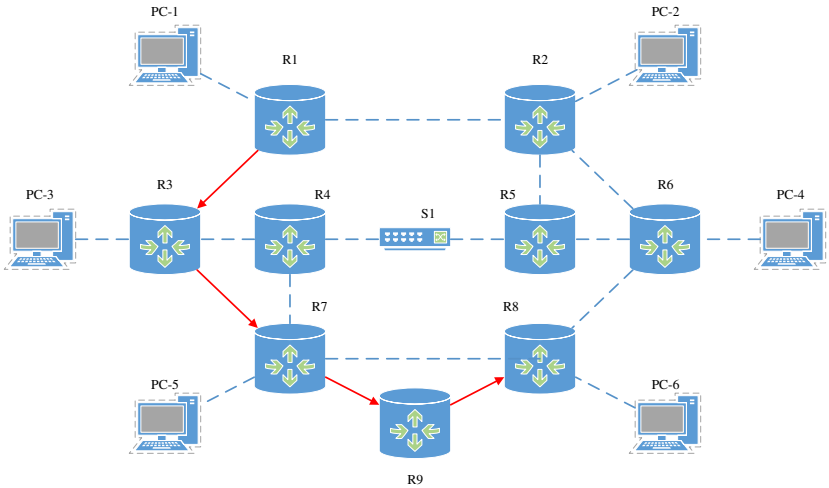
B. Hasil Efisiensi *Flooding* pada Arsitektur Fibbing dengan *Isolation Domain*

Untuk pengujian akan diberikan node source dan node destination dengan node source pada PC-1 dan node destination pada PC-6. Uji coba ini dilakukan untuk melihat kinerja arsitektur Fibbing dengan *Isolation Domain* dan pemilihan jalur untuk pengujian dalam topologi yang diajukan, kemudian hasil dari pengujian akan ditampilkan dalam bentuk grafik. Sebagai *subnetwork* dalam pengujian *subnetwork* pertama pada area PC1(R-1), PC3(R-3), PC5(R-7) dan R-4 kemudian *subnetwork* kedua pada area PC2(R-2), PC4(R-6), PC6(R-8) dan R-5.

Pemilihan subnetwork ini tidak akan mengalami perubahan saat adanya pemilihan node *source* dan node *destination* baru. Berbeda dengan skenario pengujian arsitektur Fibbing jalur yang dilewati untuk skenario Fibbing dengan *Isolation Domain* penentuan jalurnya lebih jauh akan tetapi hasil yang diberikan lebih efisien dibandingkan arsitektur Fibbing. *Isolation Domain* mengutamakan pengiriman pada *domain* nya telebih dahulu sebelum mengirimkan ke *domain* subnetwork yang berbeda. Pada tabel IV total paket yang terkirim pada saat pengujian skenario *Isolation Domain* berjumlah 11170 paket. Total paket yang terkirim saat pengujian skenario *Isolation Domain* mengalami penurunan sebanyak 19038 paket atau mengalam penurunan 36% dari total paket pada saat skenario pengujian Fibbing. Saat *flooding* terjadi hanya diproses pada masing-masing *domain* subunetwork yang telah ditentukan, *flooding* tidak akan melewati area subnetwork yang berbeda.



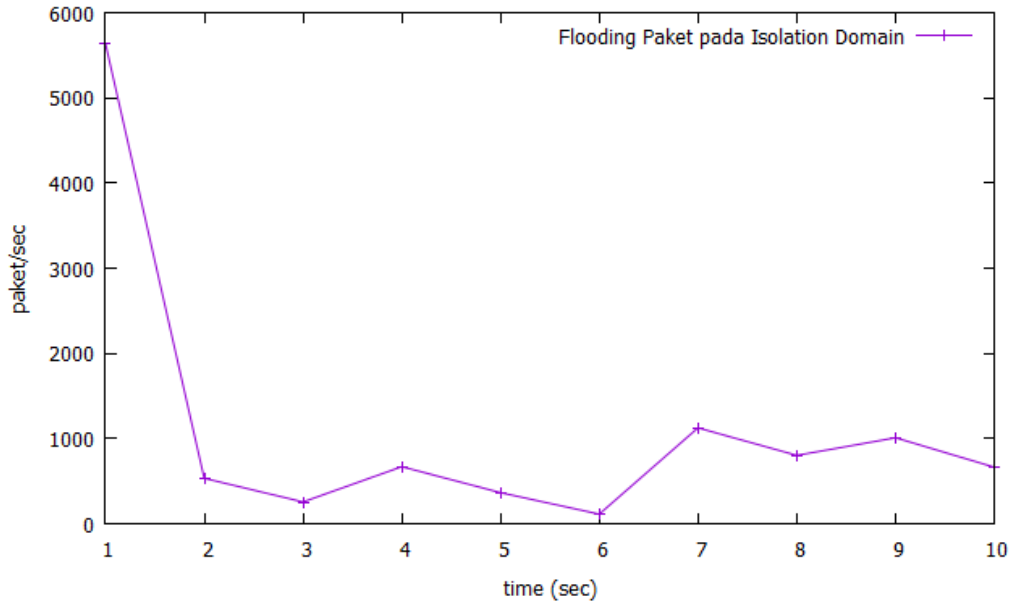
Gambar 4. Alur pengiriman PC-1 menuju PC-6 dengan Fibbing.



Gambar 6. Alur pengiriman PC-1 menuju PC-6 dengan Isolation Domain.

TABEL IV
TOTAL PAKET FLOODING SKENARIO *ISOLATION DOMAIN*.

`Waktu	Node	Jenis Paket	Paket yang Terkirim
00.00 – 01.00	R-1	LSA Type 1	5636
01.00 – 02.00			533
02.00 – 03.00	R-3	LSA Type 1	255
03.00 – 04.00			669
04.00 – 05.00	R-7	LSA Type 5	365
05.00 – 06.00			112
06.00 – 07.00	R-9	LSA Type 5	1125
07.00 – 08.00			802
08.00 – 09.00	R-8	LSA Type 1	1009
09.00 – 10.00			664
Total Paket			11170



Gambar 7. Grafik *flooding packets* pada *Isolation Domain.*

Kemudian pada Gambar 7, menunjukkan hasil dari *flooding test* dengan menggunakan metode *Isolation Domain*. Dengan waktu pengujian 10 detik dan total paket pengiriman sebesar 11170 paket, dibandingkan dengan hasil *flooding test* dari penggunaan metode arsitektur Fibbing penggunaan metode *Isolation Domain* memberikan hasil yang lebih efisien. Dengan awal pengiriman paket yang banyak hampir sama dengan hasil *flooding* paket pada Fibbing, pada Isolation Domain cukup efisien dalam menurunkan jumlah paket-paket yang dikirimkan dalam proses *flooding*. Pada detik ke-4 dan ke-7 mengalami kenaikan pada pengiriman paket dikarenakan informasi baru untuk menambahkan informasi terhadap *fake nodes* dan *fake links* yang telah dibentuk akan memperbarui informasi *forwarding table* pada topologi jaringan.

Hal ini disebabkan karena penggunaan metode *Isolation Domain* ini membantu dalam mengatur pembentukan *fake nodes* dan *fake links* saat arsitektur Fibbing berjalan pada proses *Augmentation*. Dengan metode *Isolation Domain* jalur yang tidak terlewati tidak akan membentuk *fake nodes* dan *fake links* agar memberikan hasil yang lebih efisien.

VI. KESIMPULAN

Arsitektur Fibbing yang telah di modifikasi dengan ditambah *Isolation Domain* telah berhasil disimulasikan dengan menggunakan Routeflow (Quagga) dan Mininet. Arsitektur Fibbing *original* dan arsitektur Fibbing yang telah dimodifikasi dengan *Isolation Domain* menghasilkan nilai pada pengujian *flooding* paket yang berbeda. Pada pengujian menggunakan arsitektur Fibbing menghasilkan total 30208 paket untuk proses pengiriman melalui PC-1 hingga PC-6 dan untuk pengujian menggunakan arsitektur Fibbing menggunakan *Isolation Domain* menghasilkan total 11170 paket yang terkirim. Dengan hasil yang menunjukkan bahwa Isolation Domain mampu memberikan efisiensi terhadap *flooding* paket pada arsitektur Fibbing. Arsitektur Fibbing yang dimodifikasi menggunakan *Isolation Domain* memberikan hasil dan efisiensi *flooding* cukup baik. *Isolation Domain* pun dapat meningkatkan kinerja dari arsitektur Fibbing tanpa menghilangkan proses pembuatan *fake node* dan *fake links* di semua router pada topologi jaringan.

DAFTAR PUSTAKA

[1] F. H. Saputra dan R. M. Ijtihadie, "Survei Mekanisme Congestion Kontrol Pada Transmission Control Protocol di Software Defined Network," *JUTI J Ilm Teknol Inf.* vol. 16, no. 1, 2018.

[2] R. Amin, M. Reisslein M, dan N. Shah, "Hybrid SDN networks: A survey of existing approaches," *IEEE Communications Surveys and Tutorials*, 2018.

[3] O. Tilmans dan S. Vissicchio, "IGP-as-a-backup for robust SDN networks," dalam *Proc. International Conference on Network and Service Management*, 2014.

[4] M. R. Nascimento, C. E. Rothenberg, M. R. Salvador, C. N. A Corrêa, S. C. De Lucena, dan M. F. Magalhães, "Virtual routers as a service: The RouteFlow approach leveraging software-defined networks," dalam *Proc. Int. Conf. Futur. Internet Technol,* 2011.

[5] S. Vissicchio, O. Tilmans, L. Vanbever, dan J. Rexford, "Central Control over Distributed Routing," *Computer Communication Review*, 2015.

[6] A. Rego, S. Sendra, J. M. Jimenez, dan J. Lloret, "OSPF routing protocol performance in Software Defined Networks," dalam *Proc. International Conference on Software Defined Systems*, 2017.

[7] R. M. Negara dan R. Tulloh, "Analisis Simulasi Penerapan Algoritma OSPF Menggunakan RouteFlow pada Jaringan Software Defined Network (SDN)," *J Infotel*, vol. 9, no. 1, 2017.

[8] S. Asadollahi, B. Goswami, dan M. Sameer, "Ryu controller's scalability experiment on software defined networks," dalam *Proc. IEEE International Conference on Current Trends in Advanced Computing*, 2018.

[9] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, dan T. Turletti. "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun Surv Tutorials*, vol. 16, no. 3, 2014.

[10] M. Markowski, P. Ryba, dan K. Puchala, "Software defined networking research laboratory-experimental topologies and scenarios," dalam *Proc. European Network Intelligence Conference*, 2016.

[11] L. Vanbever dan S. Vissicchio, "Enabling SDN in Old School Networks with Software-Controlled Routing Protocols," Present as part Open Netw Summit 2014 (ONS 2014). 2014. [Online]. Tersedia: https://www.usenix.org/conference/ons2014/technical-sessions/presentation/vanbever.

[12] S. Vissicchio S, L. Vanbever, dan J. Rexford, "Sweet little lies: Fake topologies for flexible routing," dalam *Proc. ACM Workshop on Hot Topics in Networks*, 2014.

[13] Sandhya, Y. Sinha, dan K. Haribabu, "A survey: Hybrid SDN," *Journal of Network and Computer Applications* 2017.

[14] A. D. Zinin dan I. M. C. Shand, "Optimizing flooding of information in link-state routing protocol". Patent 10/962842, 2005.

[15] T. Miyamura, T. Kurimoto, dan M. Aoki, "Enhancing the network scalability of link-state routing protocols by reducing their flooding overhead," dalam Proc. *IEEE International Conference on High Performance Switching and Routing*, 2003.

[16] S. H. Lee dan S. H. Rhee, "Efficient flooding for reliability in link-state routing protocols," dalam *Proc. Int Conf ICT Converg*, 2012.

[17] L. Kleinrock dan F. Kamoun, "Hierarchical routing for large networks Performance evaluation and optimization," *Comput Networks*, vol. 1, no. 3, 1977.