

RESEARCH ARTICLE

Analisis Performansi Integrasi Sistem Named Data Networking Dan Software Defined Network Untuk Komunikasi Data

Alya Shafira, Sofia Naning Hertiana* and Leanna Vidya Yovita

Fakultas Teknik Elektro, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia

* Corresponding author: sofiananing@telkomuniversity.ac.id

Received on 13 September 2023; accepted on 17 October 2023

Abstrak

Teknologi jaringan internet saat ini sudah menjadi kebutuhan utama oleh para pengguna yang memakainya. Pertukaran data informasi pada beberapa jaringan internet diusulkan sebagai alternatif *Internet Protocol* (IP). Dalam beberapa tahun terakhir, para peneliti telah menemukan jaringan arsitektur masa depan yaitu *Named Data Networking* (NDN). Arsitektur NDN diusulkan untuk memecahkan masalah komunikasi *Internet Protocol* (IP) saat ini. Tetapi, implementasi pada NDN membutuhkan perubahan dalam infrastruktur jaringan yang ada. Maka dari itu digunakan pendekatan yang layak menggunakan jaringan *Software Defined Network* (SDN), yang dimana *control plane* dapat mengoptimalkan keputusan *routing*. Hasil akhir dari penelitian ini dengan melakukan pengukuran performansi pada integrasi NDN-SDN, NDN, serta SDN dengan melihat pengiriman paket yang diminta oleh sebuah *client* dengan ukuran paket sebesar 25, 50, dan 75 paket dengan nama *prefix* yang berbeda. Kemudian membandingkan keluaran yang dihasilkan dengan parameter seperti *Throughput*, *Round Trip Time* (RTT), dan *CPU Usage* pada skenario 1 dan Skenario 2. Berdasarkan hasil pengujian yang dilakukan, bahwa semakin banyak pengiriman paket yang dilakukan akan mempengaruhi kualitas pada parameter seperti *throughput* dan RTT. Sedangkan pada *CPU usage* menunjukkan saat keadaan aktif *traffic* akan menjalankan semua program yang dilakukan, namun saat dia dalam kondisi tidak aktif maka *traffic* tidak akan menjalankan program yang dilakukan.

Key words: *Internet Protocol* (IP), *Named Data Networking* (NDN), *Software Defined Network* (SDN), *Throughput*, *Round Trip Time* (RTT), *CPU Usage*.

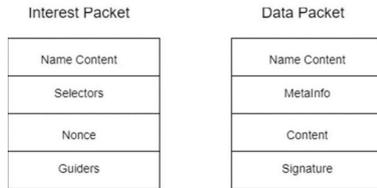
Pendahuluan

Saat ini teknologi jaringan internet berkembang sangat pesat. Pada jaringan internet pertukaran data informasi diusulkan sebagai alternatif *Internet Protocol* (IP). Dalam jaringan IP, sebagian besar model pengiriman datanya mengutamakan data apa yang harus dibutuhkan tanpa memikirkan lokasi *server* dan *host*. Sebaliknya, semakin jauh jarak lokasi *server* dan *host* maka akan berpengaruh juga pada kualitas delay yang akan dihasilkan [1]. Maka dari itu, beberapa arsitektur internet baru telah diusulkan untuk memperbaiki arsitektur jaringan IP (*Internet Protocol*), diantaranya adalah *Named Data Networking* dan *Software Defined Network*. *Name Data Networking* (NDN) merupakan rancangan arsitektur Internet.

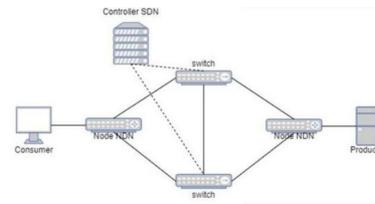
Dimana *user* atau *consumer* tidak perlu lagi memerlukan alamat IP untuk mengakses sebuah konten pada internet. Alamat IP akan diganti dengan nama data (*name*) dari konten tersebut, bukan alamat sumber ataupun tujuan. Dibandingkan dengan IP, jaringan NDN memiliki keunggulan yaitu mampu mengukur performansi jalur yang berbeda dan dapat menggunakan jalur-jalur alternatif ketika terjadi kegagalan [2].

Untuk meningkatkan performansi arsitektur IP, *Software Defined Networking* (SDN) telah dianggap sebagai solusi untuk diintegrasikan dengan NDN [3].

SDN mempunyai keunggulan dengan menerapkan pemisahan antara *control plane* dan data *plane* untuk memudahkan dalam mengelola jaringan [4]. Sehingga jaringan SDN lebih memahami kondisi jaringan secara terintegrasi. Pada penelitian sebelumnya, telah dibahas penelitian dengan membandingkan performansi pada arsitektur NDNS dan NDN *best route* [5]. Namun hanya berfokus pada sebagian performansi yang dilakukan seperti parameter *cache hit*, dan *bootstrap time* dengan menggunakan skenario topologi yang kurang bervariasi. Oleh karena itu, dalam penelitian Tugas Akhir ini akan membahas mengenai analisis performansi integrasi *Named Data Network* dan *Software Defined Network* untuk mengetahui performa pada jaringan dan perangkat menggunakan skenario yang lebih bervariasi dengan beberapa parameter seperti *throughput*, *Round Trip Time*, dan *CPU Usage*.



Gambar 1. Paket pada NDN.



Gambar 2. Desain Sistem.

Tinjauan Pustaka

Named Data Networking(NDN)

Named Data Networking (NDN) merupakan salah satu jaringan arsitektur baru dengan metode pemberian nama pada suatu data. Perubahan NDN memungkinkan adanya data plane baru, dimana suatu consumer sebagai penerima data akan mengirimkan ke paket *interest*, lalu *router* akan meneruskan dan menyimpan kondisi paket *interest*, yang dimana akan digunakan untuk membantu paket data kembali kepada *consumer* [6]. Pada NDN, komunikasi data dalam NDN dikendalikan oleh sebuah penerima data yaitu consumer data dengan melalui 2 tipe paket yaitu *interest packet* dan data *packet* [6]. Paket *interest* akan dikirimkan melalui sebuah consumer dan paket data akan dikirimkan oleh sebuah produser.

Router NDN

Pada setiap *router* NDN membutuhkan perubahan perangkat *router* yang sudah ada, karena *router* NDN mempunyai kesamaan baik dengan dengan *router* IP dan dapat melakukan *upgrade* pada *hardware* maupun *software* sehingga dapat mendukung pengiriman data pada NDN [1]. Sebuah *router* NDN terdiri dari tiga elemen yaitu, *Content Storage* (CS), *Forwarding Information Base* (FIB), dan *Pending interest Table* (PIT) [7].

Software Defined Network (SDN)

Software Defined Network (SDN) merupakan sebuah perangkat yang mendefinisikan sebuah jaringan. Jadi disini perangkat secara fleksibel mengikuti arah konfigurasi yang ditetapkan oleh perangkat itu sendiri, SDN digunakan untuk mendesain, mengelola, mengontrol, dan mengimplementasikan jaringan dalam hal untuk kebutuhan yang lebih kompleks [8]. Konsep dasarnya sendiri merupakan sebuah pemisahan antara *control plane* dan *forwarding plane* [8]. Pada *control plane* diletakkan secara terpusat dan berfungsi untuk berkomunikasi dengan paket-paket yang lain sedangkan *forwarding plane* berfungsi untuk melakukan *forwarding packet* (meneruskan paket) ke tujuan.

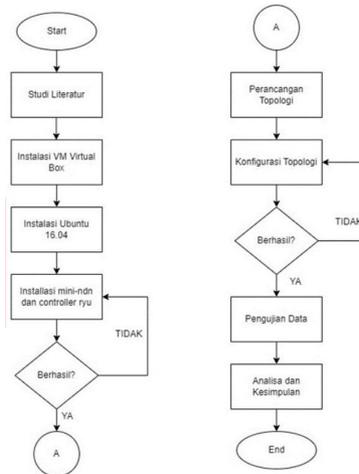
Perbedaan IP dengan NDN

Perbedaan paling mendasar antara IP dengan NDN dilihat dari aturan penamaan. Jika pada arsitektur IP memberikan suatu identitas mengenai alamat sumber dan tujuannya, sedangkan pada NDN menggunakan sistem penamaan atau nama pada suatu paket.

Metodologi Penelitian

Perancangan Sistem

Rancangan sistem berfungsi untuk menentukan spesifikasi pada konsep keseluruhan mengenai komponen penyusun yang dimana akan mendukung pengujian yang dilakukan. Gambaran perancangan sistem dapat dilihat pada Gambar 2. Pada gambar 2 menjelaskan bahwa sistem terdiri dari jaringan NDN dengan *controller* SDN yang terhubung melalui *switch*, dimana *switch* ini akan meneruskan paket ke *node*

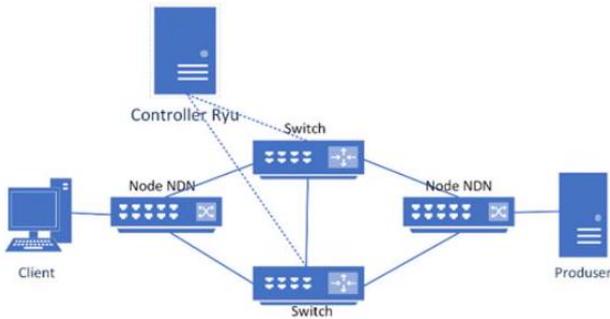


Gambar 3. Diagram Pengerjaan Sistem.

selanjutnya sesuai dengan *route* yang telah ditentukan pada *controller* SDN. Nantinya *node-node* ini akan meneruskan paket data sesuai dengan *route* yang telah ditetapkan oleh *node* NDN-nya dengan *consumer* yang melakukan permintaan data pada *node producer*. Sebuah produser berperan dengan membuat nama data yang akan diberikan sesuai dengan permintaan para *consumer*, dimana nama data yang akan dikirimkan akan melewati sebuah *node-node* NDN dan akan menyimpan nama data tersebut. Namun, jika sebuah *consumer* meminta nama data kembali, maka *consumer* tersebut bisa mendapatkan dari sebuah *node-node* NDN yang paling terdekat dengan nama data yang sudah terlewat pada produser dan *consumer* tersebut tidak perlu lagi memintanya ke produser.

Diagram Alir Pengerjaan Sistem

Untuk membantu perancangan sistem simulasi diperlukan *flowchart* dimana dapat membantu untuk memahami proses perancangan yang akan dibentuk seperti gambar 3. Hal yang pertama dilakukan yaitu dengan studi literatur, dimana penulis akan mendalami teori, lalu melihat dan mempelajari konsep-konsep dari perangkat yang akan digunakan, serta pengumpulan jurnal dan referensi lainnya. Tahap selanjutnya mulai dengan instalasi Mini-NDN dan *Ryu Controller*. Jika instalasi berhasil maka akan dilanjutkan ke tahap selanjutnya yaitu perancangan topologi. Perancangan topologi dikonfigurasi pada emulator mini-NDN yang diinstall dalam Ubuntu 16.04. Bila berhasil maka dilanjutkan dengan pengujian data. Setelah itu data yang sudah diperoleh dapat dianalisis dan menarik kesimpulan. Penelitian ini menggunakan *Virtual Machine* yaitu *VirtualBox* yang didalamnya sudah terinstall sistem operasi Ubuntu 16.04.



Gambar 4. Skenario 1-1 dengan 4 Intermediate Node NDN-SDN.

Skenario Pengujian

Pada bagian ini, akan dilakukan pengujian untuk mengukur kualitas kinerja performansi terhadap integrasi NDN-SDN, NDN, serta SDN untuk mendapatkan hasil dan kualitas yang berbeda. Setiap pengujian akan menggunakan parameter umum dan menggunakan beberapa skenario yang berbeda, adapun aspek pengujian sebagai berikut:

- Skenario Perubahan jumlah Prefix:

Pada skenario ini, akan dilakukan pengujian dengan menambahkan 3 buah nama prefix yang berbeda. Topologi yang digunakan yaitu dengan menggunakan 4 Intermediate Node dan 6 Intermediate Node. Pengujian dilakukan dengan nama prefix yang berbeda dimana pada prefix 1 /contoh, lalu 2 prefix /contoh/contoh1, dan 3 prefix seperti /contoh/contoh1/contoh2 pada sebuah client dengan menggunakan Intermediate Node yang berbeda dan hasilnya dilihat melalui parameter seperti throughput, dan Round Trip Time.

- Skenario Perubahan Ukuran Paket:

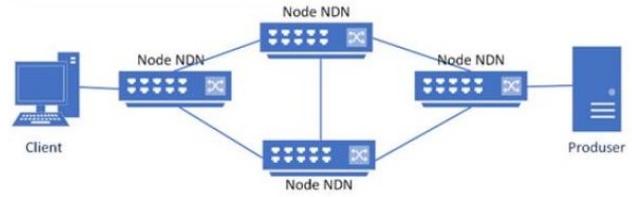
Pada skenario ini, akan dilakukan pengujian dengan memberikan ukuran paket yang berbeda. Topologi yang digunakan yaitu dengan menggunakan 4 Intermediate Node dan 6 Intermediate Node. Pengujian dilakukan dengan mengubah ukuran paket dimulai dari 25 paket, 50 paket, dan 75 paket pada sebuah client dengan menggunakan Intermediate Node yang berbeda dan hasilnya dilihat melalui parameter seperti throughput dan Round Trip Time.

- Skenario pengujian pada CPU Usage

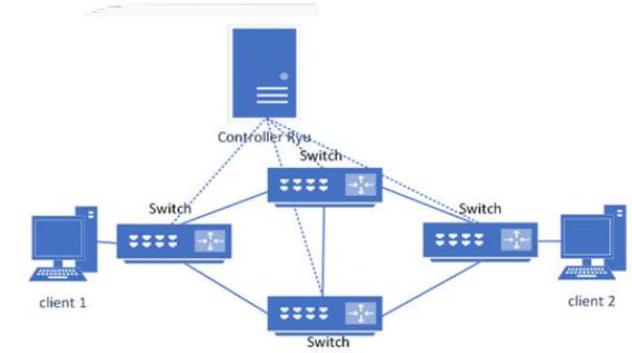
Pengujian cpu usage dilakukan untuk mengetahui konsumsi cpu di masing-masing skenario baik itu di skenario 1 dengan 4 intermediate node dan skenario 2 dengan 6 intermediate node. Pengujian dilakukan dengan melakukan pemantauan pada node/switch yang paling dekat dengan client. Pemantauan akan dilakukan saat ada traffic, dimana saat ada traffic sedang melakukan request paket, sedangkan saat tidak aktif maka client tidak akan melakukan request paket. Pengujian dilakukan selama durasi 2 menit (120 detik).

Skenario 1 dengan 4 Intermediate Node

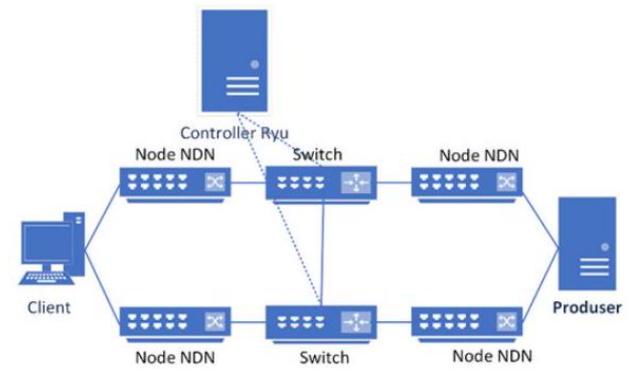
Pada skenario ini, akan dilakukan pengujian dengan menggunakan 4 buah Intermediate Node pada NDN-SDN, NDN, serta SDN. Pengujian ini dilakukan dengan melihat performansi parameter seperti throughput, Round Trip Time, dan CPU Usage yang dimana terdapat perubahan pada tiap jumlah nama prefix dan jumlah ukuran paket. Untuk topologi yang digunakan, dapat dilihat seperti gambar 4 5 6 dibawah ini:



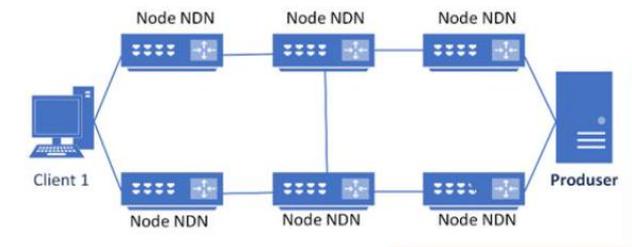
Gambar 5. Skenario 1-2 dengan 4 Intermediate Node NDN.



Gambar 6. Skenario 1-2 dengan 4 Intermediate Node NDN.



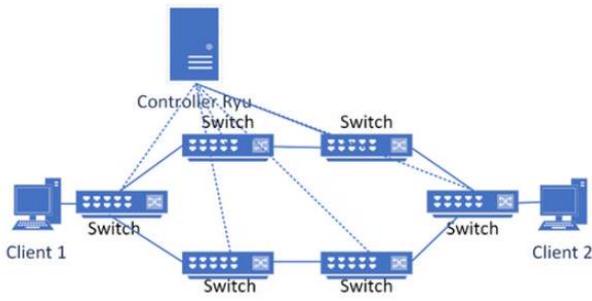
Gambar 7. Skenario 2-1 dengan 6 Intermediate Node NDN-SDN.



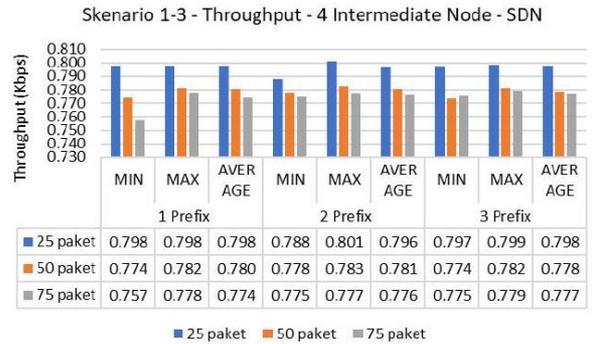
Gambar 8. Skenario 2-2 dengan 6 Intermediate Node NDN.

Skenario 2 dengan 6 Intermediate Node

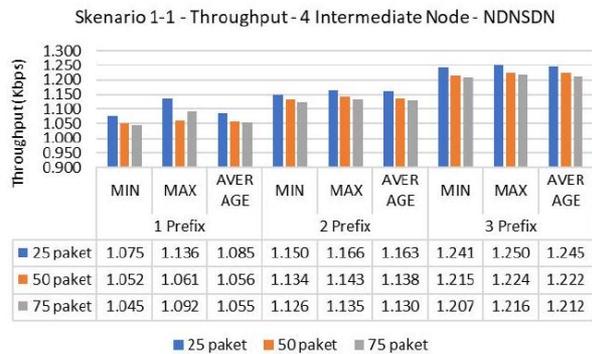
Pada skenario ini, akan dilakukan pengujian dengan menggunakan 6 buah Intermediate Node pada NDN-SDN, NDN, serta SDN. Pengujian ini dilakukan dengan melihat performansi parameter seperti throughput, Round Trip Time, dan CPU Usage yang dimana terdapat perubahan pada tiap jumlah nama prefix dan jumlah ukuran paket. Untuk topologi yang digunakan, dapat dilihat seperti gambar 7 8 9 dibawah ini:



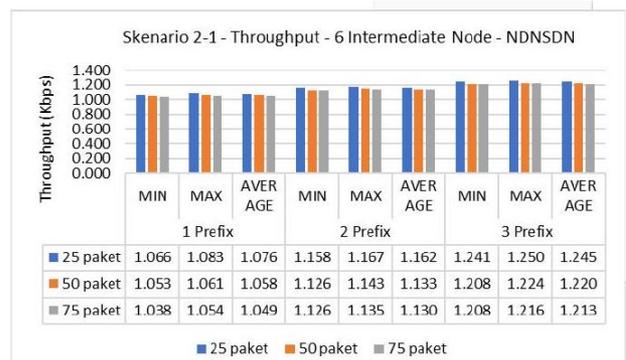
Gambar 9. Skenario 2-3 dengan 6 Intermediate Node SDN.



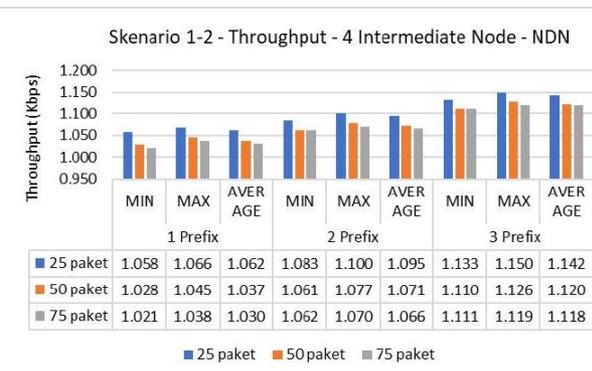
Gambar 12. Grafik throughput SDN 4 Intermediate Node.



Gambar 10. Grafik throughput NDNSDN 4 Intermediate Node.



Gambar 13. Grafik throughput NDNSDN 6 Intermediate Node.



Gambar 11. Grafik throughput NDNSDN 4 Intermediate Node.

Hasil dan Pembahasan

Hasil Pengukuran terhadap Throughput

Pengujian Skenario 14 Intermediate Node terhadap throughput

Hasil pengujian *Throughput* untuk skenario 1 dengan 4 *Intermediate Node* dilakukan dengan menambahkan 3 buah *prefix* dan mengirimkan paket sebesar 25, 50, dan 75 paket selama 1 detik (1 paket) sebanyak 10 kali percobaan.

Berdasarkan grafik pada Gambar 10 menunjukkan hasil pengukuran *throughput* untuk skenario 1 dengan menggunakan jumlah permintaan yang dikirim yaitu 25, 50 dan 75 paket pada NDN-SDN. Nilai rata-rata *throughput* pada semua pengujian *prefix* 1 sampai *prefix* 3 dengan jumlah paket tersebut pada NDN-SDN sebesar 1.145 Kbps.

Berdasarkan grafik pada Gambar 11 menunjukkan hasil pengukuran *throughput* untuk skenario 1 dengan menggunakan jumlah permintaan yang dikirim yaitu 25, 50 dan 75 paket pada NDN. Nilai rata-rata *throughput* pada semua pengujian *prefix* 1 sampai *prefix* 3 dengan jumlah paket tersebut pada NDN sebesar 1.082Kbps.

Berdasarkan grafik pada Gambar 12 menunjukkan hasil pengukuran *throughput* dengan nilai rata-rata pada semua pengujian *prefix* 1 sampai *prefix* 3 dengan jumlah paket tersebut pada SDN sebesar 0.784 Kbps.

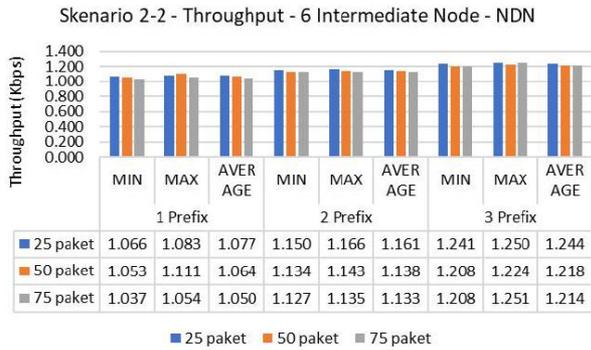
Pengujian Skenario 26 Intermediate Node terhadap throughput

Untuk pengujian *throughput* pada skenario 2 6 *Intermediate node* sama dilakukan seperti skenario 1.

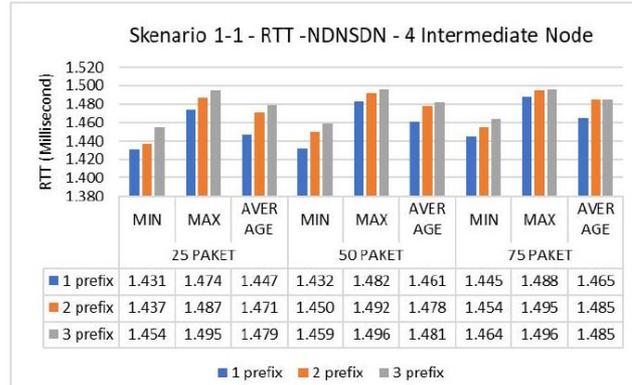
Berdasarkan grafik pada Gambar 13 menunjukkan hasil pengukuran *throughput* untuk skenario 2 dengan menggunakan jumlah permintaan yang dikirim yaitu 25, 50 dan 75 paket pada NDNSDN. Nilai rata-rata *throughput* pada semua pengujian *prefix* 1 sampai *prefix* 3 dengan jumlah paket tersebut pada NDNSDN sebesar 1.143 Kbps.

Berdasarkan grafik pada Gambar 14 menunjukkan hasil pengukuran *throughput* untuk skenario 2 dengan nilai rata-rata pada semua pengujian *prefix* 1 sampai *prefix* 3 dengan jumlah paket tersebut pada NDN sebesar 1.144 Kbps.

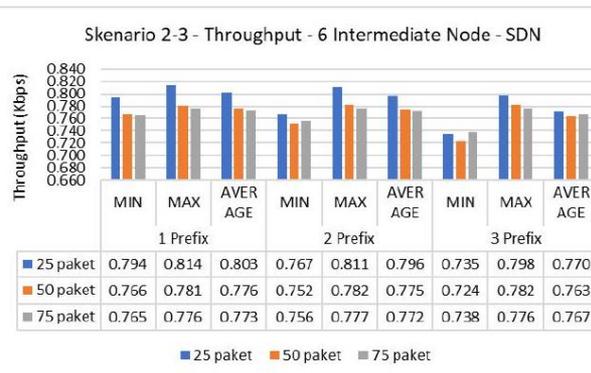
Berdasarkan grafik pada Gambar 15 menunjukkan hasil pengukuran *throughput* untuk skenario 2 dengan nilai rata-rata *throughput* pada semua pengujian *prefix* 1 sampai *prefix* 3 dengan jumlah paket tersebut pada NDN sebesar 0.777 Kbps.



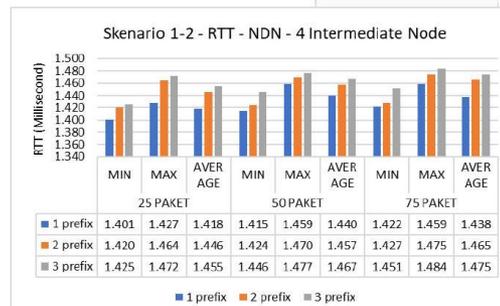
Gambar 14. Grafik throughput NDN 6 Intermediate Node.



Gambar 16. Grafik RTT NDNSDN.



Gambar 15. Grafik throughput SDN 6 Intermediate Node.



Gambar 17. Grafik RTT NDN 4 Intermediate Node.

Hasil Analisis Pada Throughput

Berdasarkan hasil pengujian *Throughput* pada skenario 1 dan skenario 2, dapat dilihat bahwa semakin banyak jumlah *prefix* dan jumlah permintaan yang dikirimkan sebesar 25, 50, dan 75 paket nantinya akan mempengaruhi besar nilai *throughput* yang didapat. Namun semakin besar jumlah permintaan yang dikirimkan dan semakin banyak jumlah *prefix* pada *client* menyebabkan nilai rata-rata *throughput* yang didapatkan semakin kecil. Pengaruh jumlah *prefix* pada NDN-SDN, NDN, dan SDN terhadap nilai *throughput* yang semakin kecil disebabkan karena semakin banyak jumlah *prefix* yang digunakan maka akan meningkatkan pengiriman paket berdasarkan jumlah pengiriman yang telah dilakukan.

Hasil Pengukuran terhadap Round Trip Time(RTT)

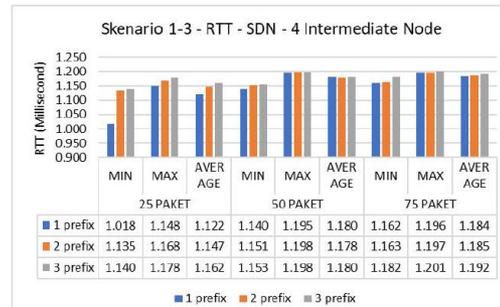
Pengujian Skenario 14 Intermediate Node terhadap RTT

Berdasarkan grafik pada Gambar 16 menunjukkan hasil pengukuran RTT untuk skenario 1 pada NDN-SDN. Dengan menggunakan penambahan jumlah *prefix* pada NDNSDN. Nilai rata-rata RTT pada pengiriman paket 25, 50, dan 75 untuk nilai rata-rata NDNSDN sebesar 1.472 ms.

Hasil Pengukuran terhadap Round Trip Time (RTT)

Pengujian Skenario 14 Intermediate Node terhadap RTT

Pada Gambar 17 menunjukkan hasil pengukuran RTT untuk skenario 1 pada NDN dengan nilai rata-rata untuk keseluruhan untuk NDN sebesar 1.451 ms.

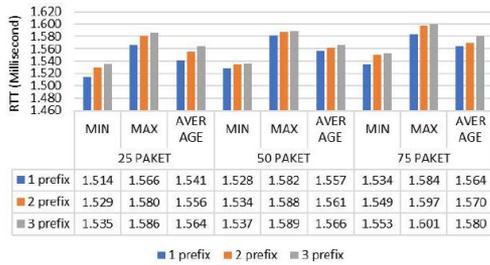


Gambar 18. Grafik RTT SDN 4 Node.

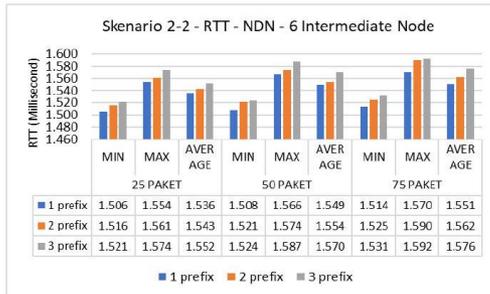
Berdasarkan grafik pada Gambar 18 menunjukkan hasil pengukuran RTT untuk skenario 1 pada SDN. Dengan nilai rata-rata untuk keseluruhan skenario SDN sebesar 1.170 ms.

Pengujian Skenario 26 Intermediate

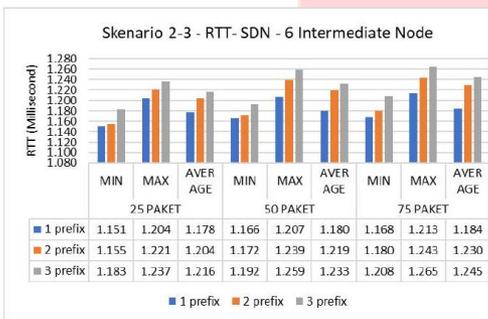
Node terhadap RTT Skenario 2-1 - RT - NDNSDN - 6 Intermediate Node Berdasarkan grafik pada Gambar 19 menunjukkan hasil pengukuran RTT untuk skenario 2 pada NDNSDN. Dengan menggunakan penambahan jumlah *prefix* pada NDNSDN. Nilai rata-rata RTT pada pengiriman paket 25, 50, dan 75 untuk NDNSDN sebesar 1.562 ms. Berdasarkan grafik pada Gambar 20 menunjukkan hasil pengukuran RTT untuk skenario 2 pada NDN. Nilai rata-rata RTT pada pengiriman paket 25, 50, dan 75 untuk NDN sebesar 1.555 ms. Berdasarkan grafik pada gambar 21 menunjukkan hasil pengukuran RTT pada SDN, nilai rata-rata pada pengiriman paket sebesar 1.210 ms.



Gambar 19. Grafik RTT NDNSDN 6 Node.



Gambar 20. Grafik RTT NDN 6 Node.



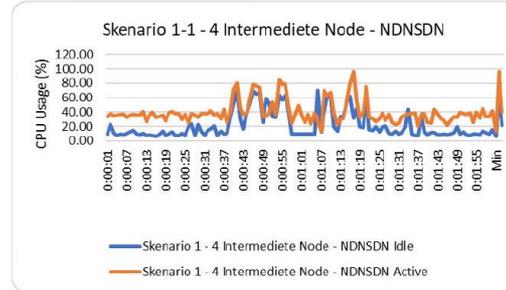
Gambar 21. Grafik RTT NDNSDN 6 Node.

Hasil Analisis RTT

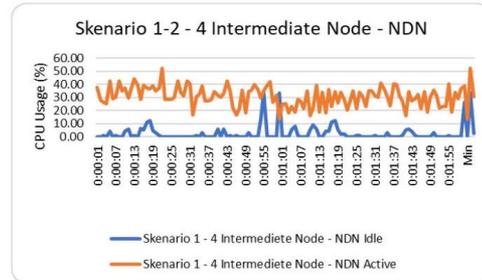
Berdasarkan pengujian hasil dari *Round Trip Time* (RTT) pada skenario 1 dengan 4 *Intermediate Node* dan skenario 2 dengan 6 *Intermediate node* terdapat kenaikan nilai RTT pada pengujian yang dipegaruhi dengan pengiriman paket sebesar 25, 50, dan 75 paket dan jumlah *prefix* yang meningkat. Pada pengujian NDN-SDN membutuhkan waktu untuk melakukan penguraian paket pada NDN secara berulang dan terjadi proses penerusan (*forwarding*) dari *switch* pada *controller* SDN. Sedangkan NDN, semua proses tersebut dilakukan oleh *node* NDN itu sendiri. Pada SDN karena dilakukan pada *switch* sendiri. Hal ini terjadi karena semakin banyaknya jumlah paket dan *prefix* disebabkan karena dengan bertambahnya jumlah paket maka jalur yang ditempuh agar sampai ke tujuan semakin jauh sehingga menyebabkan waktu yang dibutuhkan paket agar sampai ke tujuan menjadi bertambah, dan bisa juga karena banyak faktor yang mempengaruhi, mulai dari proses rute yang dilewati, waktu pengamatan dan lain-lain.

Hasil Pengukuran terhadap CPU Usage

Hasil pengujian *CPU usage* untuk skenario 1 dan 2 akan menunjukkan penggunaan *resource* CPU saat sedang ada *traffic* (*active state*)



Gambar 22. Grafik CPU NDNSDN 4 Intermediate Node.



Gambar 23. Grafik CPU NDN 4 Intermediate Node.

mapupun saat tidak ada *traffic* (*idle state*) pada NDN-SDN, NDN, dan SDN. Pengujian dilakukan dengan melakukan pemantauan pada *node switch* yang paling dekat dengan *client*. Pemantauan akan dilakukan saat ada *traffic*, dimana saat ada *traffic* sedang melakukan *request* paket, sedangkan saat tidak aktif maka *client* tidak akan melakukan *request* paket. Pengujian dilakukan selama durasi 2 menit (120 detik) agar menjadi pembandingan antara integrasi NDN-SDN, NDN, dan SDN.

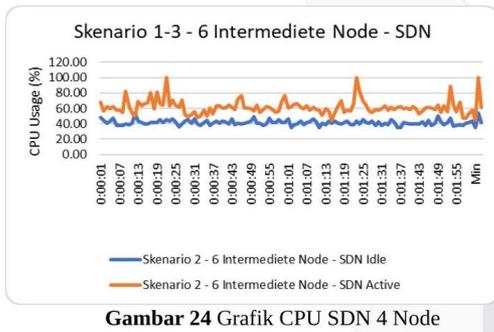
Pengujian Skenario 14 Intermediate Node Terhadap CPU Usage

Pada Gambar 22 menunjukkan hasil pengukuran *CPU Usage* untuk skenario 1 pada NDN-SDN. Pada integrasi NDN-SDN, saat kondisi *idle state CPU usage* berada pada rentang kondisi 6.19% sampai 69.80% dengan perolehan rata-rata 21.54% untuk penggunaan CPU setiap detiknya. Tetapi saat kondisi *active state*, kondisi *CPU usage* berada pada rentang kondisi 13.13% sampai 97.00% dengan perolehan rata-rata 40.01% untuk penggunaan setiap detiknya.

Pada Gambar 23 saat kondisi *idle state CPU usage* berada pada rentang kondisi 0.00% sampai 33.67% dengan perolehan rata-rata 2.41 % untuk penggunaan CPU setiap detiknya. Tetapi saat kondisi *active state*, kondisi CPU berada pada rentang kondisi 13.27% sampai 52.04% dengan perolehan rata-rata 30.85% untuk penggunaan setiap detiknya. Pada Gambar 24 menunjukkan hasil pengukuran CPU saat kondisi *idle state CPU usage* berada pada rentang kondisi 34.69% sampai 53.61% dengan perolehan rata-rata 41.41% untuk penggunaan CPU setiap detiknya. Tetapi saat kondisi *active state*, kondisi *CPU usage* berada pada rentang kondisi 44.33% sampai 100.00% dengan perolehan rata-rata 61.46% untuk penggunaan setiap detiknya.

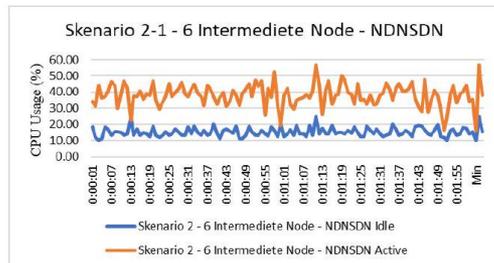
Pengujian Skenario 26 Intermediate Node Terhadap CPU Usage

Pada Gambar 25 menunjukkan hasil pengukuran CPU saat kondisi *idle CPU usage* berada pada rentang kondisi 10.10% sampai 25.00% dengan perolehan rata-rata 15.20% untuk penggunaan CPU setiap detiknya. Tetapi saat kondisi *active state*, kondisi *CPU usage* berada pada rentang kondisi 16.33% sampai 57.00% dengan perolehan rata-rata 38.10% untuk penggunaan setiap detiknya.

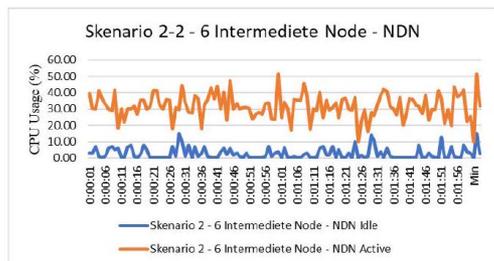


Gambar 24. Grafik CPU SDN 4 Node.

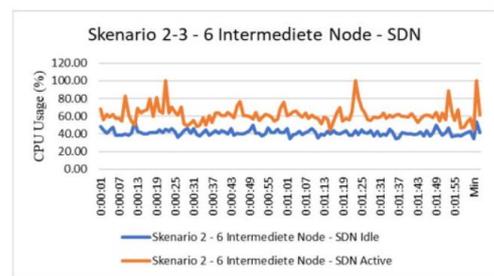
Gambar 24. Grafik CPU SDN 4 Node.



Gambar 25. Grafik CPU NDNSDN 6 Node.



Gambar 26. Grafik CPU NDN 6 Node.



Gambar 27. Grafik CPU SDN 6 Node.

Pada Gambar 26 menunjukkan hasil pengukuran *CPU Usage* untuk skenario 2 pada NDN. Pada NDN, saat kondisi *idle state CPU usage* berada pada rentang kondisi 0.00% sampai 15.00% dengan perolehan rata-rata 2.66% untuk penggunaan CPU setiap detiknya. Tetapi saat kondisi *active state*, kondisi CPU berada pada rentang kondisi 10.20% sampai 51.49% dengan perolehan rata-rata 31.79% untuk penggunaan setiap detiknya. Pada Gambar 27 menunjukkan hasil pengukuran *CPU usage* untuk skenario 2 pada SDN. Pada SDN, saat kondisi *idle state CPU usage* berada pada rentang kondisi 34.69% sampai 53.61%

dengan perolehan rata-rata 41.41% untuk penggunaan CPU setiap detiknya. Tetapi saat kondisi *active state*, kondisi CPU berada pada rentang kondisi 44.33 % sampai 100.00% dengan perolehan rata-rata 61.46% untuk penggunaan setiap detiknya.

Hasil Analisis *CPU Usage*

Berdasarkan pengujian hasil dari *CPU usage* untuk semua skenario, pada integrasi NDN-SDN NDN-SDN *switch node* memiliki penggunaan CPU yang cukup baik dikarenakan memerlukan informasi ketersediaan rute tetangga, proses pemilihan *route* yang diteruskan ke *controller* SDN untuk ditangani, *switch* sendiri difungsikan untuk meneruskan (*forward*) paket berdasarkan hasil *controller* SDN. Pada SDN sendiri seiring dengan bertambah jumlah *switch*, maka akan meningkatkan persentase kerja CPU. Sedangkan pada NDN, semua proses tersebut dilakukan oleh *node* NDN itu sendiri, maka dari itu *node* NDN memiliki penggunaan CPU yang lebih kecil. Oleh karena itu semakin besar atau banyaknya program yang sedang dijalankan, pengaruh *CPU usage* akan berdampak jika *CPU usage* pada sebuah *consumer* dalam suatu jaringan melebihi batas maksimal. Tetapi jika *CPU usage* sedang dalam keadaan tidak aktif, maka semua program akan pada batas normal.

Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, didapatkan beberapa kesimpulan sebagai berikut.

1. Pada integrasi NDN-SDN mempunyai proses penguraian paket, dan juga informasi ketersediaan rute tetangga (*face create*), dan proses pemilihan *route* yang diteruskan ke *controller* SDN untuk ditangani, *switch* sendiri difungsikan untuk meneruskan (*forwarding*) paket berdasarkan hasil *controller* SDN.
2. *Throughput* yang didapatkan pada penelitian ini menunjukkan bahwa semakin banyak jumlah paket dan *prefix* yang dikirimkan maka nilai *throughput* yang didapatkan akan semakin kecil.
3. Pada aspek RTT, bertambahnya jumlah paket dan *prefix* disebabkan karena semakin banyak jumlah paket, jalur yang ditempuh untuk sampai ke tujuan semakin jauh sehingga menyebabkan waktu yang dibutuhkan paket untuk sampai ke tujuan semakin besar.
4. Penggunaan *CPU usage* saat terdapat *traffic* dan saat tidak ada *traffic* lebih baik saat *traffic* sedang berjalan. Karena saat keadaan aktif *traffic* akan menjalankan semua program dan semakin banyak program yang dijalankan maka pengaruh *CPU usage* akan berdampak jika *CPU usage* pada sebuah *consumer* dalam suatu jaringan melebihi batas maksimal.

Daftar Pustaka

1. Melati SR, Yovita LV, Mayasari R. Caching Performance of Named Data Networking with NDNS. In: International Conference on Information Networking. IEEE; 2021. p. 261-6.
2. Amadeo M, Campolo C, Ruggeri G, Molinaro A, Iera A. Understanding Name-based Forwarding Rules in Software-Defined Named Data Networking. In: IEEE International Conference on Communications; 2020. .
3. Zhang QY, Wang XW, Huang M, Li KQ, Das SK. Software Defined Networking Meets Information Centric Networking: A Survey. IEEE Access. 2018;6:39547-63.
4. Hidayat I. Arsitektur Software Defined Network: Implementasi Pada Small Network. Jurnal Jaringan Komputer dan Keamanan. 2020;1(Februari):1-13.
5. Kalghoum A, Gammar SM. Towards a New Information Centric Networking Strategy Based on Software Defined Networking.

- In: IEEE Wireless Communications and Networking Conference (WCNC); 2017. .
6. Zhang L, et al. Named data networking. ACM SIGCOMM Computer Communication Review. 2014;44(3):66-73.
 7. Kalghoum A, Gammar SM, Saidane LA. Towards a novel forwarding strategy for named data networking based on SDN and bloom filter. In: IEEE/ACS International Conference on Computer Systems and Applications (AICCSA); 2018. p. 1198-204.
 8. sufyaldy. Software Defined Network 'Inovasi dan Masa Depan Network Science'; 2015. Available from: <https://sufyaldy.wordpress.com>.