



Doi: [10.17933/jppi.v11i2.330](https://doi.org/10.17933/jppi.v11i2.330)

**Analisis Performansi *Baremetal Provisioning* pada *Openstack Platform* Berbasis *Remote Virtualisasi* Menggunakan Layanan *Ironic***  
***Baremetal Provisioning Performance Analysis Of Openstack Platform Based on Remote Virtualization Using Ironic***

**Joey Leomanz Bartolomiussihosa<sup>1</sup>, Agus Virgono<sup>2</sup>, Ridha Muldina Negara<sup>3</sup>**

Fakultas Teknik Elektro<sup>123</sup>  
Universitas Telkom<sup>123</sup>

Jl. Telekomunikasi Terusan Buah Batu Jawa Barat 40257, Indonesia<sup>123</sup>

[joeyleomanzb@student.telkomuniversity.ac.id](mailto:joeyleomanzb@student.telkomuniversity.ac.id)

Naskah Diterima: 27 January 2021 ; direvisi: 26 Desember 2021; Diterima: 27 Desember 2021

### Abstract

Penggunaan *cloud computing* sudah cukup banyak, bahkan banyak dipergunakan untuk melakukan pekerjaan tugas komputasi kinerja tinggi. *Cloud data center* mengandalkan teknologi virtualisasi untuk meningkatkan produktivitas dan mengurangi kompleksitas yang akan diberikan ke pengguna akhir dalam mengakses layanannya yang diberikan. Aplikasi yang berjalan dengan menggunakan teknologi virtualisasi tidak terhindarkan terjadinya penurunan performansi. Kebutuhan konsumen akan komputasi tinggi yang tidak dapat dilakukan pada teknologi virtualisasi dapat disolusikan dengan melakukan komputasi langsung di atas *baremetal* namun tetap pada kompleksitas yang sangat rendah. *Openstack* direpresentasikan sebagai *platform opensource*, populer sebagai *Infrastructure as a Service (IaaS) cloud platform* yang bisa diimplementasikan sebagai *private cloud* maupun publik *cloud*. *Openstack* sudah mendukung dalam melakukan teknik virtualisasi maupun mengakuisisi *baremetal*. Penelitian ini dilakukan uji performansi antara *baremetal* dan virtual mesin dengan melakukan beberapa pengujian. Pengujian performansi berdasarkan sumber daya infrastruktur dari komputasi awan tersebut seperti *CPU processing time*, *network throughput TCP*, *disk I/O*, *memory test throughput*, *jitter*, dan *packet loss*. Hasil pengujian yang didapat, performansi *baremetal ironic* lebih baik dari virtual mesin.

**Kata kunci:** *openstack*, *ironic*, *benchmarking*, *provisioning baremetal*, virtualisasi

### Abstract

*The use of cloud computing is already quite a lot, even many are used to perform high-performance computing tasks. Cloud data centers rely on virtualization technology to increase productivity and reduce the complexity that will be provided to end-users in accessing the services provided. Applications that run using virtualization technology inevitably result in performance degradation. Consumer needs for high computing that cannot be done on virtualization technology can be solved by computing directly on baremetal but still at very low complexity. Openstack is represented as an open-source platform, popular as an Infrastructure as a Service (IaaS) cloud platform that can be implemented as a private cloud or public cloud. Openstack already supports virtualization and acquisition of baremetal. This research was conducted to test the performance between baremetal and virtual machines by doing several tests. Performance testing is based on infrastructure resources from cloud computing such as CPU processing time, network throughput TCP, disk I/O, memory test throughput, jitter, and packet loss. The test results obtained that, baremetal ironic performance is better than virtual machines.*

**Keywords:** *openstack*, *ironic*, *benchmarking*, *provisioning baremetal*, virtualization

# Analisis Performansi *Baremetal Provisioning* pada *Openstack Platform* Berbasis *Remote Virtualisasi* Menggunakan Layanan *Ironic*

## *Baremetal Provisioning Performance Analysis Of Openstack Platform Based on Remote Virtualization Using Ironic*

Joey Leomanz Bartolomiussihosa<sup>1</sup>, Agus Virgono<sup>2</sup>, Ridha Muldina Negara<sup>3</sup>

Fakultas Teknik Elektro<sup>123</sup>  
Universitas Telkom<sup>123</sup>

Jl. Telekomunikasi Terusan Buah Batu Jawa Barat 40257<sup>123</sup>

[joeyleomanzb@student.telkomuniversity.ac.id](mailto:joeyleomanzb@student.telkomuniversity.ac.id)

---

### Abstrak

Penggunaan *cloud computing* sudah cukup banyak, bahkan banyak dipergunakan untuk melakukan pekerjaan tugas komputasi kinerja tinggi. *Cloud data center* mengandalkan teknologi virtualisasi untuk meningkatkan produktivitas dan mengurangi kompleksitas yang akan diberikan ke pengguna akhir dalam mengakses layanannya yang diberikan. Aplikasi yang berjalan dengan menggunakan teknologi virtualisasi tidak terhindarkan terjadinya penurunan performansi. Kebutuhan konsumen akan komputasi tinggi yang tidak dapat dilakukan pada teknologi virtualisasi dapat disolusikan dengan melakukan komputasi langsung di atas *baremetal* namun tetap pada kompleksitas yang sangat rendah. Openstack direpresentasikan sebagai *platform opensource*, populer sebagai *Infrastructure as a Service (IaaS) cloud platform* yang bisa diimplementasikan sebagai *private cloud* maupun publik *cloud*. Openstack sudah mendukung dalam melakukan teknik virtualisasi maupun mengakuisisi *baremetal*. Penelitian ini dilakukan uji performansi antara *baremetal* dan virtual mesin dengan melakukan beberapa pengujian. Pengujian performansi berdasarkan sumber daya infrastruktur dari komputasi awan tersebut seperti *CPU processing time*, *network throughput TCP*, *disk I/O*, *memory test throughput*, *jitter*, dan *packet loss*. Hasil pengujian yang didapat, performansi *baremetal ironic* lebih baik dari virtual mesin.

**Kata kunci:** *openstack*, *ironic*, *benchmarking*, *provisioning baremetal*, virtualisasi

### Abstract

*The use of cloud computing is already quite a lot, even many are used to perform high-performance computing tasks. Cloud data centers rely on virtualization technology to increase productivity and reduce the complexity that will be provided to end-users in accessing the services provided. Applications that run using virtualization technology inevitably result in performance degradation. Consumer needs for high computing that cannot be done on virtualization technology can be solved by computing directly on baremetal but still at very low complexity. Openstack is represented as an open-source platform, popular as an Infrastructure as a Service (IaaS) cloud platform that can be implemented as a private cloud or public cloud. Openstack already supports virtualization and acquisition of baremetal. This research was conducted to test the performance between baremetal and virtual machines by doing several tests. Performance testing is based on infrastructure resources from cloud computing such as CPU processing time, network throughput TCP, disk I/O, memory test throughput, jitter, and packet loss. The test results obtained that, baremetal ironic performance is better than virtual machines.*

**Keywords:** *openstack*, *ironic*, *benchmarking*, *provisioning baremetal*, virtualization

## PENDAHULUAN

Pembangunan infrastruktur *cloud computing* saat ini dikembangkan terus-menerus, bertujuan untuk mendukung layanan bisnis yang dimiliki konsumen pada bagian infrastruktur. Peningkatan akan kebutuhan server *demand*, membuat penyedia layanan *cloud computing* berlomba-lomba membangun infrastruktur untuk bersaing mendapatkan peluang profit yang tinggi (Cisco System. Inc, 2020). *Cloud computing* merupakan penggunaan sumber daya komputasi seperti CPU atau penyimpanan yang disediakan sebagai layanan. Sumber daya komputasi yang dimiliki berada di lokasi penyedia layanan. Akses sumber daya dilakukan dengan kontrol jarak jauh. *Cloud Computing* sering disebut sebagai *Internet Based Service* (Red Hat. Inc, 2014)

Seiring perkembangan zaman, teknologi pada komputer ikut mengalami kemajuan yang cukup pesat yang menghasilkan kecepatan komputasi yang begitu efisien dan mampu melakukan banyak tugas (Qasaimah, *et al.*, 2019). Kemampuan komputasi tersebut membuat penyedia layanan *cloud computing* memanfaatkan teknologi virtualisasi pada perangkat keras yang tersedia guna memaksimalkan sumber daya perangkat keras tersebut. Virtualisasi merupakan teknik membuat versi maya dari sebuah perangkat keras yang dimiliki kemudian dijalankan atau disimpan dalam bentuk beberapa sumber daya maya, yang sering disebut mesin virtual (Kumar & Charu, 2015). Teknik virtualisasi juga diadopsi oleh salah satu sistem *opensource cloud platform* yaitu Openstack

sebagai *Infrastructure as a Service Cloud Computing* (Awasthi & Gupta, 2016).

Konferensi tentang *A Comparison of System Performance on a Private Openstack Cloud and Amazon EC2* menyimpulkan bahwa kemampuan pengelolaan virtual mesin dengan Openstack sudah sangat bagus, menghasilkan mesin virtual dengan performansi yang optimal (Kang, *et al.*, 2017). Tidak semua penggunaan *cloud computing* dengan teknologi virtualisasi dapat dikerjakan, seperti penggunaan *high processing game server hosting* dan *high compute memory* yang menyebabkan penggunaan virtualisasi menjadi tidak ideal (IBM Corporation, 2020). Aplikasi seperti *Enterprise Resource Planning (ERP)*, *Customer Relationship Management (CRM)*, *Supply Chain Management (SCM)*, sangat ideal dan relatif stabil menggunakan *baremetal* (IBM Corporation, 2020). Openstack merupakan jenis layanan dari *cloud computing* berbasis *Infrastructure as a Service (IaaS)* mampu mengatur sebuah perangkat untuk melakukan komputasi dengan teknik virtualisasi dan *baremetal* (Kumar, *et al.*, 2014).

Penelitian ini mirip dengan penelitian yang dilakukan oleh Nicolas Seyvet, Konstantinos Vandikas, Charalampos Gavriil dan Kominos pada tahun 2017 dengan judul *Baremetal, Virtual Machine, and Container in Openstack*, hanya saja yang saja adanya penambahan pengujian *packet loss* maupun *jitter* sebagai kategori QoS di dalam jaringan komputer. *Quality of Service (QoS)* adalah sebuah metode menjadi tolak ukur seberapa baik dari jaringan komputer yang kita miliki dan suatu usaha untuk mendefinisikan karakteristik atau sifat dari suatu servis. Hasil analisis QoS, dapat dijadikan

rekomendasi untuk implementasi fisik jaringan internet (ITU-T Study Group, 2001).

**Tabel 1. Referensi Standarisasi QoS ITU-T G.1010**

Parameter	Data	Voip	Video
			Streaming
<i>Jitter</i>	N.A.	<1 ms	N.A.
<i>Throughput</i>	N.A.	4.64 kbit/s	16.384 Kbps
<i>Packet Loss</i>	0%	<3%	<1%

Sumber: (ITU-T Study Group, 2001)

*Jitter* merupakan variasi *delay*. Pengiriman data berlangsung, paket data yang dikirimkan pengirim dalam bentuk utuh dan dalam waktu yang bersamaan (ETSI, 1999).

$$Jitter = \frac{Total\ Variasi\ Delay}{Total\ Paket\ yang\ di\ terima}$$

Total Variasi *Delay* = *Delay* – (rata – rata *delay*)

**Tabel 2. Referensi Kategori *Jitter* TIPHON**

Kategori <i>Jitter</i>	<i>Jitter</i> (ms)
Sangat Bagus	<0 ms
Bagus	0 ms s/d 75 ms
Sedang	75 ms s/d 125 ms
Jelek	>125 ms s/d 255 ms

Sumber: (ETSI, 1999)

*Throughput* merupakan suatu metode pengukuran, untuk mengukur kecepatan *rate* transfer data efektif dalam satuan *bit per second* (bps). *Throughput* memiliki perhitungan dari jumlah total paket data yang diterima oleh penerima dan sukses pada suatu interval tertentu dibagi oleh durasi interval waktu tertentu (ETSI, 1999).

$$Throughput = \frac{Paket\ data\ diterima}{Lama\ Pengamatan}$$

**Tabel 3. Referensi Kategori *Throughput* TIPHON**

Kategori <i>Throughput</i>	Besar <i>Throughput</i>
Sangat Bagus	>1.200 Kbps
Bagus	700 – 1.200 Kbps
Cukup	338 – 700 Kbps
Jelek	0– 338 Kbps

Sumber: (ETSI, 1999)

*Packet loss* merupakan suatu parameter pada jaringan yang mengukur jumlah paket data yang hilang saat terjadi pengiriman data (ETSI, 1999).

$$Packet\ Loss = \frac{(Packet\ Data\ Kirim - Packet\ Data\ Di\ terima) \times 100\%}{Paket\ Data\ yang\ Dikirim}$$

**Tabel 4. Referensi Kategori *Packet Loss* TIPHON**

Kategori <i>Packet Loss</i>	Besar <i>Packet Loss</i>
Sangat Bagus	0 %
Bagus	3 %
Cukup	15 %
Buruk	25 %

Sumber: (ETSI, 1999)

Penelitian ini menggunakan bantuan layanan dari Openstack bernama Nova yang berfungsi sebagai manajemen sumber daya virtual mesin dengan menerapkan teknologi virtualisasi. Layanan Openstack Ironic merupakan layanan tambahan yang dapat diintegrasikan dengan Openstack untuk mengaktifkan dan mengimplementasikan *baremetal provisioning*. *Baremetal provisioning* adalah proses menjalankan sistem operasi pada perangkat komputer dengan menggunakan bantuan jaringan komputer. Komputer melakukan *network boot*, dengan metode *Pre Boot Execution Environment* (PXE). Sistem operasi berbentuk *image* akan

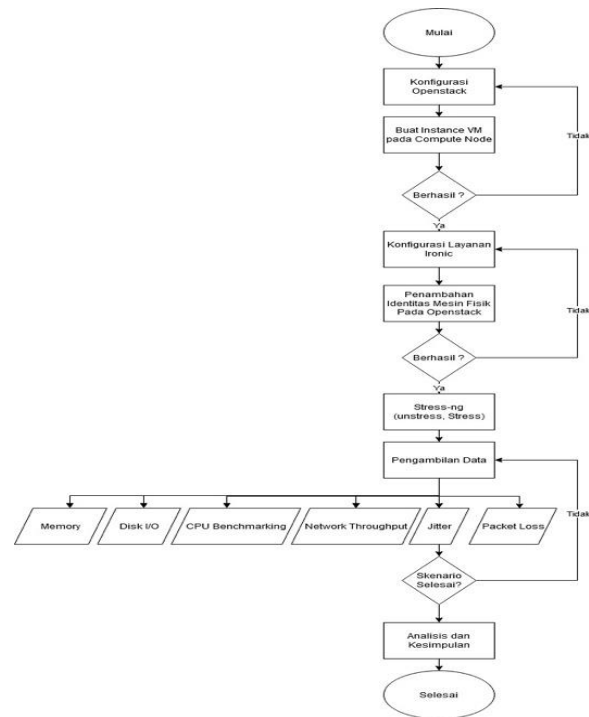
ditransfer menuju komputer tujuan dengan menggunakan bantuan protokol *Trivial File Transfer Protocol* (TFTP) (Nicolas, et al., 2017)

Alasan mendasar menggunakan Ironic, mengikuti tujuan dasar Ironic diciptakan dari *project opensource* Openstack (Openstack, 2019). Setelah penerapan layanan Ironic pada Openstack, penulis melakukan penelitian dan analisis performansi antar virtual mesin dan *baremetal* dengan pengujian parameter *CPU processing time, network throughput TCP, disk I/O, memory test throughput, jitter, dan packet loss*.

## METODE

Penelitian ini diawali dengan melakukan studi literatur, implementasi, pengujian dan pengambilan data. Studi literatur dilakukan dengan cara mencari, mengumpulkan dan memahami baik jurnal, artikel maupun buku referensi sumber – sumber yang berhubungan dengan Openstack dan Ironic. Implementasi sistem pada perangkat keras dirancang menggunakan *cloud platform* yaitu Openstack. Umumnya setelah tahap instalasi selesai, Openstack dapat melakukan pembuatan virtual mesin. Openstack dapat diintegrasikan dengan layanan Ironic, sehingga dapat terciptanya *baremetal provisioning*. Setelah implementasi berhasil, dilakukan pengujian dengan *stress tool* dan pengambilan data.

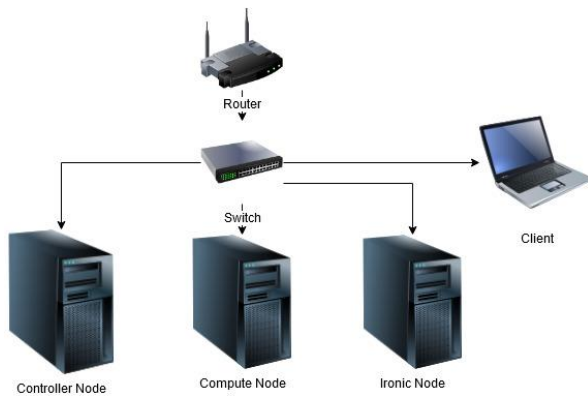
## Alur Perancangan



**Gambar 1. Alur Penelitian Baremetal Provisioning**

Gambar 1 menjelaskan diagram alir pengerjaan penelitian ini. Pengerjaan penelitian ini dimulai dengan menentukan spesifikasi perangkat yang digunakan untuk pengujian. Openstack merupakan *opensource cloud platform* yang digunakan untuk mengatur dan mengelola perangkat keras. Setelah instalasi Openstack selesai dilakukan, maka Openstack sudah dapat digunakan. Pengujian *baremetal ironic* perlu tambahan layanan yang bernama Ironic, guna Openstack dapat melakukan *baremetal provisioning*. Setelah layanan Ironic sudah dilakukan instalasi, selanjutnya dilakukan pengujian dan pengambilan data pada setiap skenario.

## Desain Jaringan



**Gambar 2. Topologi Jaringan Server**

Gambar 2 merupakan gambaran perangkat keras yang disiapkan untuk penelitian ini. Openstack pada Controller node maupun Compute node diinstalasi menggunakan teknik instalasi manual sesuai dokumentasi yang di berikan Openstack (Rackspace Cloud Computing, 2021).

## Spesifikasi Perangkat

Bagian ini berisi spesifikasi perangkat yang digunakan untuk menunjang penelitian.

### a. Perangkat Keras

Tabel berikut merupakan spesifikasi perangkat yang digunakan sistem Openstack dan dilakukan pengujian kemudian pengambilan data.

**Tabel 5. Spesifikasi Controller Node**

<b>Processor</b>	Intel (R) Xeon (R) CPU @2.00Ghz
<b>RAM</b>	8 GB
<b>Harddisk</b>	72 GB

Tabel 5 merupakan spesifikasi perangkat keras yang digunakan sebagai Controller. Controller node berfungsi untuk memerintahkan node lain.

**Tabel 6: Spesifikasi Compute Node**

<b>Processor</b>	Intel (R) Xeon (R) CPU @2.90Ghz
<b>RAM</b>	8 GB
<b>Harddisk</b>	72 GB

Tabel 6 merupakan spesifikasi perangkat yang digunakan sebagai Compute . Perangkat ini yang akan digunakan untuk menjalankan Virtual Mesin. Virtual mesin digunakan sebagai pengujian pada penelitian ini.

**Tabel 7. Spesifikasi Ironic Node**

<b>Processor</b>	Intel (R) Xeon (R) CPU @2.60Ghz
<b>RAM</b>	8 GB
<b>Harddisk</b>	72 GB

Tabel 7 merupakan spesifikasi perangkat keras Ironic Node. Ironic Node akan menjalankan sistem operasi yang diperintahkan oleh Controller Node. Setelah sistem operasi sudah berjalan dengan bantuan PXE *Boot*, tahap selanjutnya, sistem operasi dilakukan pengujian sesuai parameter yang sudah disebutkan sebelumnya pada penelitian ini.

### b. Perangkat Lunak

Tabel berikut adalah spesifikasi pada perangkat lunak pada penelitian ini.

**Tabel 8. Spesifikasi Perangkat Lunak Host**

<b>Software</b>	Nama
<b>Sistem Operasi</b>	Centos 7 Server
<b>Openstack</b>	Rocky
<b>Terminal Emulator</b>	PuTTY

Tabel 8 merupakan spesifikasi perangkat lunak yang dijalankan pada Controller Node maupun Compute Node. Sistem Operasi yang dijalankan adalah Centos 7 Server. Penelitian ini

menggunakan Openstack versi Rocky. Saat penelitian ini dikerjakan, Openstack Rocky merupakan versi paling stabil.

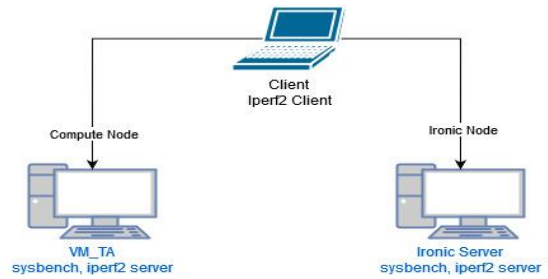
**Tabel 9. Spesifikasi Perangkat Lunak Pengujian**

<i>Software</i>	<b>Nama</b>
<b>Sistem Operasi</b>	Ubuntu Server 18.04 LTS
<b>Traffic Generator</b>	Iperf2
<b>Benchmarking</b>	Sysbench
<b>Load Generator</b>	Stress-ng
<b>Terminal Emulator</b>	PuTTY

Tabel 9 menunjukkan virtual mesin OS untuk pengujian menggunakan Ubuntu 18.04 LTS *server*. Aplikasi Stress-ng sebagai *load generator* digunakan untuk membuat mesin berada pada keadaan sibuk dengan mengatur spesifik *core* yang ingin digunakan. Sysbench digunakan untuk pengukuran performansi CPU pada VM maupun Ironic node. Aplikasi Iperf2 sebagai *traffic generator* untuk pengukuran *network throughput*, *jitter* dan *packet loss*. Penggunaan Putty mempermudah melakukan *remote* pada server untuk pemberian *load* pada virtual mesin dan Ironic node.

**Skenario Penelitian**

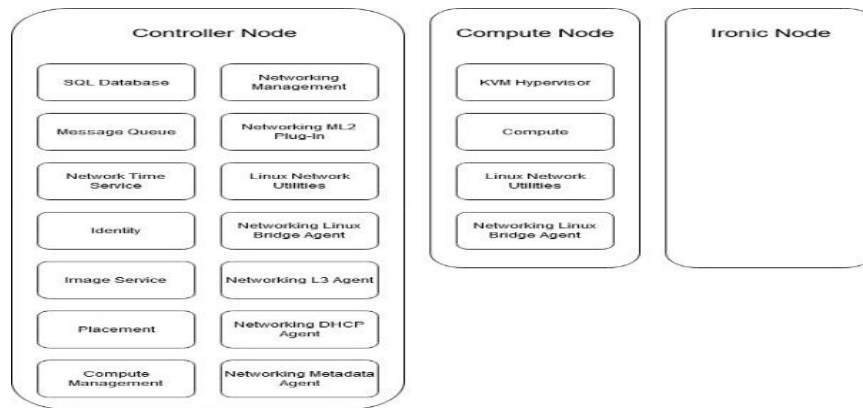
Skenario penelitian dimulai dengan membuat virtual mesin. Saat virtual mesin sudah berjalan dan dapat diakses dengan lancar, dilanjutkan proses instalasi layanan Ironic. Layanan ini berguna untuk melakukan *baremetal provisioning*. Target tujuan *baremetal provisioning* diarahkan ke server Ironic Node. Gambaran pengujian seperti pada gambar 3.



**Gambar 3. Skenario Pengujian**

**Tabel 10. Skenario Pengujian Performansi**

<b>Jumlah Virtual Mesin</b>	1 VM
<b>Jumlah Baremetal</b>	1 Baremetal Node
<b>Jumlah Core @VM</b>	4 vCore (s)
<b>Jumlah Core Baremetal</b>	4 Core
<b>Background Load</b>	0%, 50% , 100%
<b>Background Stress</b>	Unstress, 2 Core Stress, 4 Core Stress
<b>Transport Protokol</b>	TCP
<b>Pengulangan Pengujian @ Skenario</b>	10 kali pengulangan

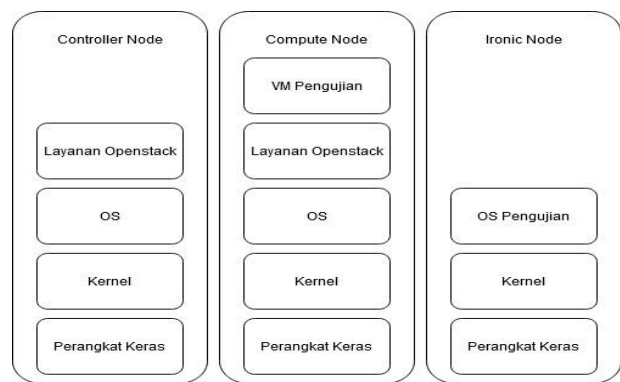


**Gambar 4. Layanan Openstack di instalasi pada masing – masing server**  
 Sumber : (Rackspace Cloud Computing, 2021)

Tabel 10 merupakan rincian skenario pada penelitian ini. Pengujian akan dilakukan dengan 3 macam percobaan yaitu percobaan pertama tanpa diberi beban sama sekali pada virtual mesin maupun Ironic node. Percobaan kedua dengan menambahkan beban *background stress* sebanyak 2 core pada masing-masing virtual mesin dan Ironic node yang diuji. Percobaan ketiga dengan memberikan beban maksimal pada *background stress* sebanyak 4 core pada masing-masing virtual mesin dan Ironic Node. Maksimal 4 core dikarenakan server yang digunakan memiliki total 4 core. Data hasil CPU *Benchmarking, network throughput, memory, disk I/O, jitter* dan *packet loss* dari setiap pengujian skenario dikolektif di laptop klien. Setiap skenario dilakukan pengujian pengulangan pengambilan data sebanyak 10 kali dan diambil nilai rata-rata nya.

**Instalasi Layanan Openstack**

Openstack yang terdiri dari beberapa layanan dilakukan instalasi pada perangkat masing – masing sesuai dokumentasi Openstack berikan (Rackspace Cloud Computing, 2021). Gambar 4 merupakan gambaran masing-masing layanan dilakukan instalasi pada masing–masing server.

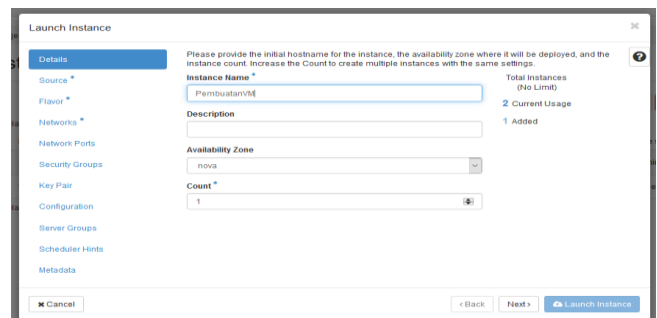


**Gambar 5. Arsitektur Instalasi Openstack**

Sumber: (Rackspace Cloud Computing, 2021)

Gambar 5 menunjukkan VM akan berjalan pada mesin Compute Node dan OS pengujian akan berjalan pada Ironic Node. VM dan OS pengujian yang nanti-nya dilakukan uji performansi.

*Pembuatan Virtual Mesin*



**Gambar 6. Pembuatan VM dengan Dashboard Openstack**

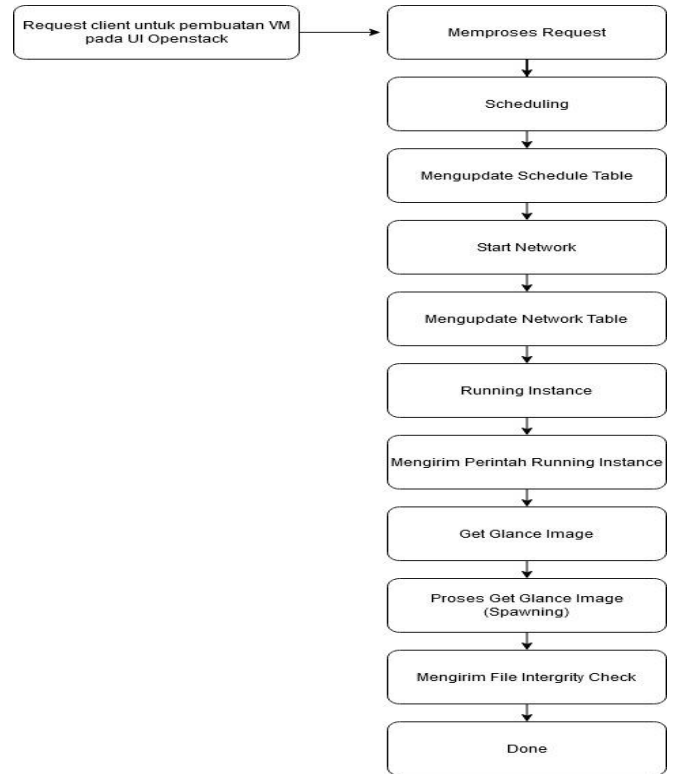


Gambar 6 merupakan tampilan saat pengguna ingin membuat virtual mesin pada Openstack. Pengguna diminta untuk mengisi keterangan pada kolom kosong. Kolom bersifat opsional jika tidak ditandai simbol “\*”.

**Tabel 11. Skenario Metadata untuk Penambahan Virtual Mesin pada Openstack**

Data	Penjelasan
<i>Instance Name</i>	Nama VM
<i>Source</i>	OS yang ingin dilakukan uji coba. Pengujian ini dipilih Ubuntu server 18.04 LTS
<i>Flavor</i>	Spesifikasi VM, pada pilihan ini , disesuaikan dengan spesifikasi Ironic Node
<i>Network</i>	Pengaturan jaringan di dalam Openstack. Disesuaikan dengan <i>network</i> pada Ironic node

Tabel 11 menunjukkan daftar data yang harus diisi saat pembuatan virtual mesin. Setelah data dasar yang sudah terisi, dilakukan *launch instance*. Kemudian, Controller Node akan memberikan perintah kepada Compute Node untuk melakukan pembuatan virtual mesin. Virtual mesin akan berjalan di dalam Compute node.



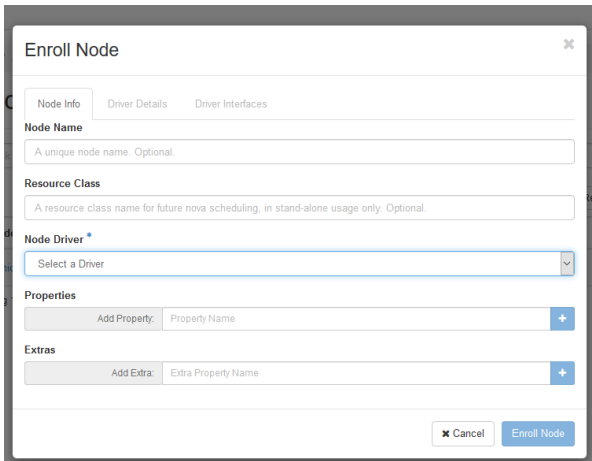
**Gambar 7. Alur data Pembuatan VM**

Sumber: (Zheng, *et al.*, 2017)

Gambar 7 merupakan alur data yang terjadi di dalam Openstack, saat pengguna melakukan *request* pembuatan virtual mesin pada Openstack. Pengguna hanya mengisi data pada kolom yang disediakan, sesuai pada penjelasan sebelumnya, tahap – tahap berikutnya akan dilakukan otomatis oleh komputer.

**Penambahan Identitas Ironic Node**

Openstack tidak dapat melakukan penambahan identitas secara otomatis, perlu pengguna untuk melakukannya secara manual.



**Gambar 8. Penambahan Identitas Ironic Node**

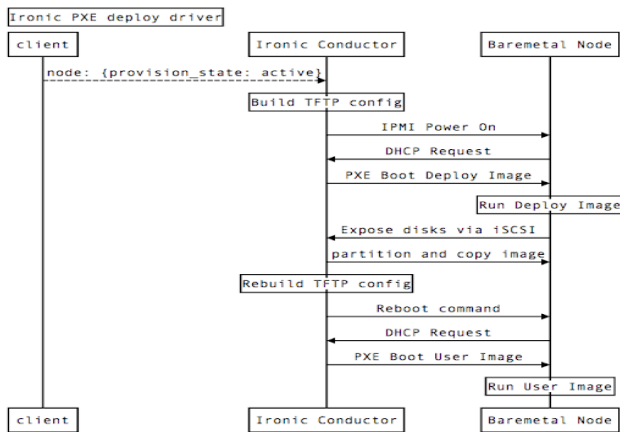
Gambar 8 merupakan tampilan pada Openstack, untuk penambahan identitas Ironic Node. Identitas yang diisi adalah identitas IPMI yang terdapat pada server. Identitas bersifat *case sensitivitas*, sehingga perlu diisi dengan benar.

**Tabel 12. Skenario Metadata untuk Penambahan Ironic Node pada Openstack**

Data	Penjelasan
<i>Node Name</i>	Nama mesin yang ditambahkan
<i>Node Driver</i>	Sistem antar muka untuk mengelola perangkat mesin dari jarak jauh. Tugas IPMI untuk menghidupkan atau mematikan perangkat
<i>IPMI Address</i>	Alamat IP yang sudah diatur terlebih dahulu. Pastikan Openstack dapat mengaksesnya.
<i>IPMI Username</i>	Identitas nama pengguna untuk mengakses perangkat keras tujuan
<i>IPMI Password</i>	Identitas <i>password</i> untuk mengakses perangkat keras tujuan
<i>Deploy ramdisk</i>	Perangkat lunak , yang merepresentasikan seperti <i>harddisk</i> , namun berjalan di dalam <i>memory</i>
<i>Deploy kernel</i>	Sistem operasi , untuk memberikan <i>kernel</i> dasar pada OS untuk melakukan <i>booting</i> dasar pada perangkat mesin tujuan

Tabel 12 menunjukkan daftar data yang harus diisi saat menambahkan perangkat keras yang dijadikan Ironic Node.

Setelah data diisi tahap selanjutnya, tekan tombol *enroll node*. Selesai dari pengisian data, maka mesin fisik sudah terdaftar pada Openstack.



**Gambar 9. Proses Layanan IroniC mengakuisisi IroniC Node**

Sumber: (Browne, 2019)

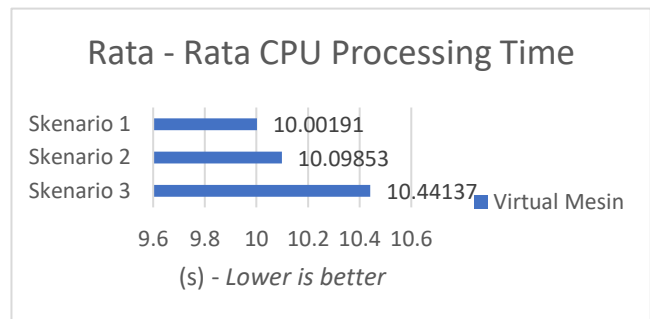
Gambar 9 merupakan tahap proses layanan IroniC mengakuisisi *baremetal node* yaitu IroniC. Tahap pada gambar merupakan urutan yang dilakukan Controller Node sehingga IroniC Node dapat melakukan *boot* dengan PXE. Setelah pembuatan VM untuk dijalankan pada Compute Node berhasil dan IroniC Server sudah berhasil *boot* OS pengujian yang diinginkan, tahap selanjutnya tinggal melakukan pengambilan data pada VM maupun baremetal.

## HASIL DAN PEMBAHASAN

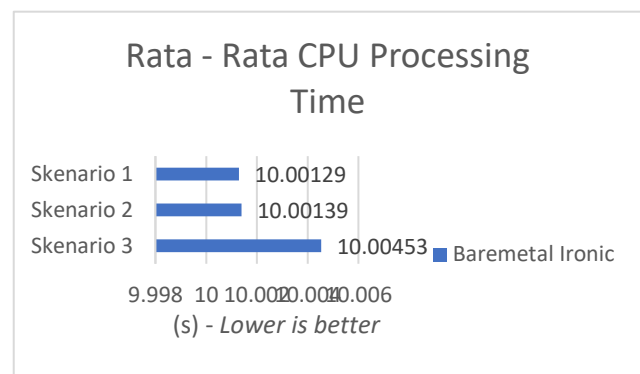
Bab ini membahas mengenai hasil pengujian yang telah dilakukan sebanyak 10 kali disetiap skenario, kemudian hasil tersebut dilakukan rata-rata dari setiap percobaan yang telah dilakukan, berguna untuk mendapatkan hasil tingkat keakuratan sesuai dengan diharapkan. Skenario satu merupakan skenario *unstress* sehingga tidak ada beban *background load* yang diberikan. Skenario dua merupakan skenario dengan *background load* sebesar 50 % dilakukan *stress* CPU sebanyak 2 *core* dari total 4 *core* prosesor yang dimiliki. *Stress*

menggunakan aplikasi *Stress-ng* yang dijalankan pada sistem operasi, dengan waktu *stress* yang tidak ditentukan. *Stress test* diberhentikan secara manual. Skenario tiga merupakan skenario dengan *background load* sebesar 100 %, dilakukan *stress* CPU sebanyak 4 *core* dari total 4 *core* prosesor yang dimiliki. Sama dengan sebelumnya *stress* menggunakan aplikasi *Stress-ng* dengan waktu *stress* yang tidak ditentukan. Tahapan pengujian, sistem dilakukan *stress* disesuaikan disetiap skenario, selanjutnya dilakukan uji performansi menggunakan aplikasi *Sysbench* maupun *Iperf2*.

### CPU Processing Time



**Gambar 10. Grafik Rata - Rata CPU Processing Virtual Mesin**



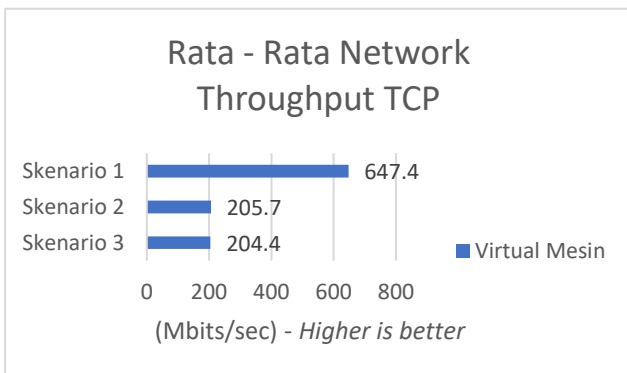
**Gambar 11. Grafik Rata - Rata CPU Processing Baremetal IroniC**

Pengujian *CPU Processing Time* menggunakan aplikasi *Sysbench*. CPU diperintahkan untuk melakukan *compile* Lua JIT code dengan JIT

compiler. Kecepatan eksekusi pada CPU tersebut yang menjadi nilai uji performansi.

Gambar 10 dan 11 merupakan hasil rata – rata pengujian CPU *processing time* pada *baremetal ironic* dan virtual mesin disetiap skenario. Hasil dari evaluasi, *baremetal ironic* mendapatkan performa lebih baik, karena penurunan performa tidak terlalu signifikan. Virtual mesin mendapatkan lonjakan penurunan signifikan pada skenario tiga. Faktor yang mempengaruhi virtual mesin kurang berjalan optimal karena *hypervisor* pada virtual mesin yaitu KVM perlu waktu dalam memproses penjadwalan terhadap prosesor fisik pada *host* yang sedang diuji coba. Ironic Node sebagai tujuan *baremetal provisioning* mendapatkan performansi optimal dibandingkan virtual mesin dikarenakan proses penjadwalan yang diberikan langsung diperintahkan ke CPU fisik yang dimiliki *baremetal ironic*. Saran ke depan untuk meminimalkan penurunan performa pada virtual mesin perlu dilakukan uji coba dengan penambahan KVM CPU *Pinning* (Podzimek, *et al.*, 2015).

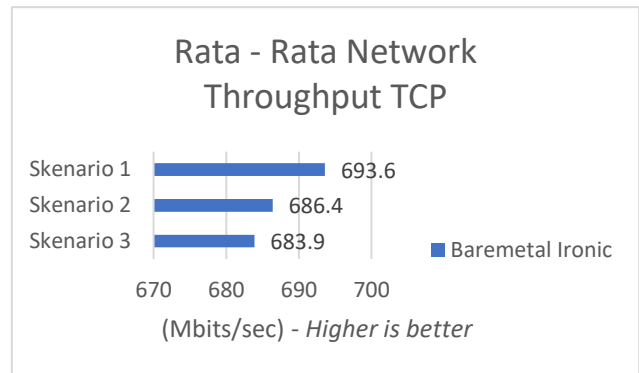
**Network Throughput TCP**



**Gambar 12. Grafik Rata - Rata Network Throughput Virtual Mesin**

**Gambar 13. Grafik Rata - Rata Network Throughput Baremetal Ironic**

*Throughput* merupakan kecepatan transfer data

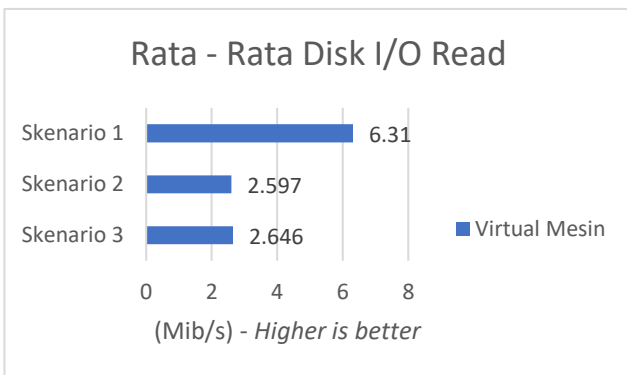


yang diukur dalam satuan *bit per second* (bps) yang dapat dilalui pada sebuah jaringan komputer. Pengujian *throughput* TCP merupakan pengujian kecepatan transfer data dengan metode TCP. Aplikasi yang digunakan untuk pengujian ini menggunakan Iperf2. Iperf2 akan mengukur *throughput* maksimal yang dapat dilalui dari *bandwidth available* yang dimiliki. Nilai *throughput* tersebut yang menjadi nilai uji performansi. Pengujian *bandwidth* perlu dilakukan untuk mengetahui *bandwidth available* pada lingkungan uji yang dimiliki, dengan menjalankan *iperf test* pada server Nova Compute Node dan Ironic Node. *Bandwith available* yang didapatkan sebesar 695 Mbits/sec. Pengujian ini juga berguna menghindari pengetahuan secara teoritis *bandwith available* 1000 Mbits/sec didasarkan menggunakan infrastruktur dengan *gigabit interface*.

Berdasarkan hasil uji performansi menurut referensi TIPHON dapat dikatakan sangat bagus karena besar *throughput* disetiap skenario pengujian antara *baremetal ironic* dan virtual mesin pada nova compute node di atas >1200Kbps. Penurunan performa jaringan terjadi pada virtual mesin jika diberikan beban di atas 50%. Saat dilakukan

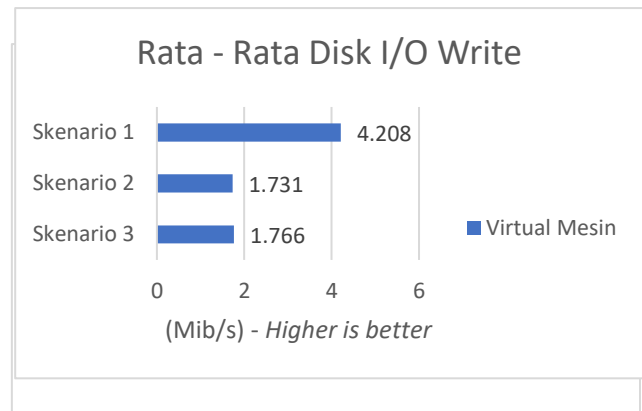
pengecekan ulang dan dicoba konfigurasi ulang pada tap *interface network* atau virtual *network* pada virtual mesin yang disediakan oleh neutron menghasilkan nilai MTU 1450 bytes. Standar MTU pada *interface* adalah 1500 bytes. Penyebab hal tersebut menjadi salah satu faktor dikarenakan perlu *processing packet* pada saat terjadi transmisi data untuk dapat melewati paket dengan MTU 1450 bytes menuju 1500 bytes, maupun sebaliknya. *Processing packet* tersebut juga dipengaruhi oleh beragam *background load* yang diatur disetiap skenario. *Processing packet* perlu dibantu CPU, namun pada virtual mesin, proses tersebut harus dilalui melewati *hypervisor* terlebih dahulu. *Baremetal ironic* sebagai tujuan *baremetal provisioning* lebih optimal karena transmisi data dilalui langsung *physical interface* dan memiliki standar MTU. Mengenai perbedaan MTU virtual mesin belum adanya dokumentasi yang dapat memperbaiki hal tersebut.

**Disk I/O**

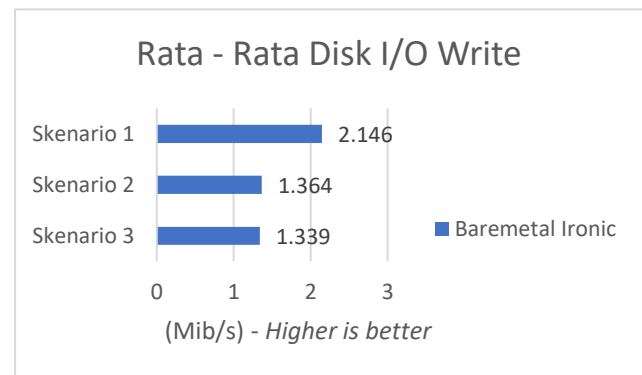


**Gambar 14. Grafik Rata - Rata Disk I/O Read Virtual Mesin**

**Gambar 15. Grafik Rata - Rata Disk I/O Read Baremetal Ironic**



**Gambar 16. Grafik Rata - Rata Disk I/O Write Virtual Mesin**



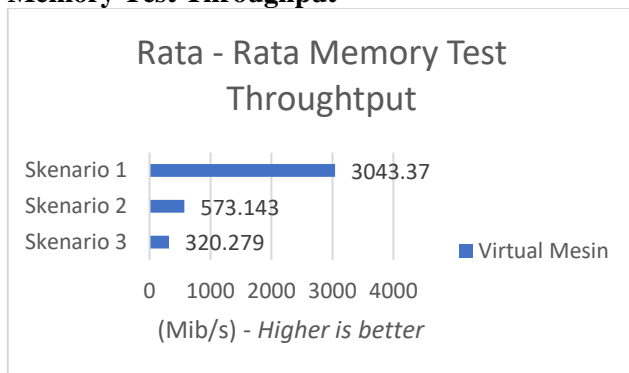
**Gambar 17. Grafik Rata - Rata Disk I/O Write Baremetal Ironic**

Pengujian *disk I/O* menggunakan aplikasi Sysbench. Setelah dilakukan instalasi Sysbench pada virtual mesin dan *baremetal ironic*, langkah selanjutnya memberikan perintah terhadap Sysbench untuk membuat *file* sampel sebesar 8GB untuk keperluan uji *read* dan *write*. Alasan menggunakan *file* sebesar 8GB untuk menghindari *cache* yang tersimpan pada RAM. Perlu diingat layanan Openstack yang berjalan mengambil porsi kurang lebih 5% dari total RAM. Saat pengujian berlangsung, Sysbench akan memerintahkan sistem operasi untuk melakukan *random access* pada *harddisk*. Cara kerja dari *random access*, *file* 8 GB

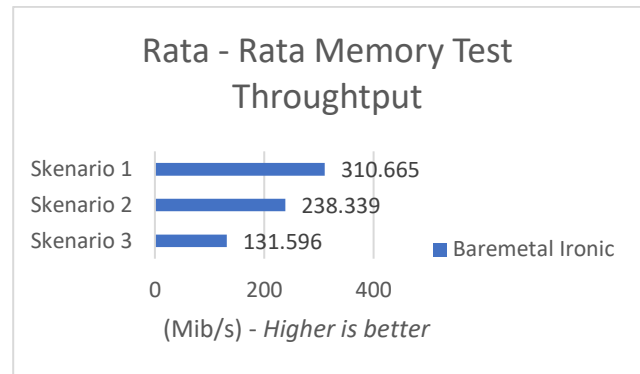
tersebut ditaruh secara acak pada sebuah *block* maupun *track* tertentu pada sebuah *harddisk*. Kecepatan nilai membaca atau menulis tersebut yang menjadi nilai uji performansi.

Gambar 14 dan 16 menunjukkan penurunan *speed disk I/O* virtual mesin pada *read* maupun *write* yang cukup signifikan saat mesin sudah diberikan *load* di atas 50%. Faktor penyebab *load* tersebut membuat *hypervisor* dalam melakukan transfer data translasi antara mesin fisik dengan virtual mesin melambat. Gambar 15 dan 17 merupakan hasil uji performansi *Disk I/O* pada *baremetal ironic*. Penurunan tidak terlalu signifikan karena proses transfer data langsung pada *harddisk* fisik. Perlu tingginya *I/O Operation per second (IOPS)* dan *CPU Instruction* pada perangkat Nova Compute Node untuk meminimalkan penurunan tersebut (Felter, et al., 2015).

**Memory Test Throughput**



**Gambar 18: Grafik Rata - Rata Memory Test Throughput Virtual Mesin**



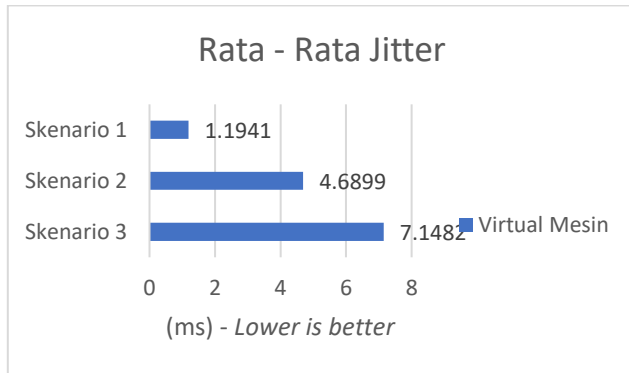
**Gambar 19: Grafik Rata - Rata Memory Test Throughput Baremetal Ironic**

*Memory* adalah tempat penyimpanan data sementara dalam komputer. *Memory* bekerja menyimpan data-data sementara dan menyuplai data-data penting yang dibutuhkan *processor* dengan cepat untuk diolah menjadi informasi. Pengujian *memory* menggunakan aplikasi Sysbench. Sysbench mencoba akses sebuah *file* sementara pada setiap *block memory* secara *sequentially*. Alat pengujian ini menggunakan prosesor Intel Xeon CPU tanpa *hyperthreading*, dengan L1 *cache* 32KB *data / instruction cache*, L2 *cache* 256KB, L3 *cache* 4MB. Pemrosesan data dari *memory* juga melewati L *cache* tersebut. Kecepatan akses *memory / throughput* tersebut yang menjadi nilai uji performansi.

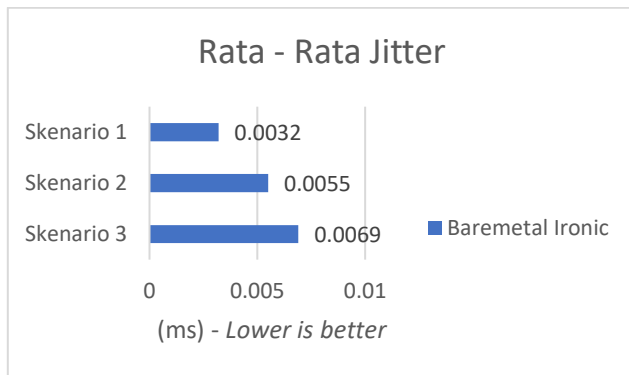
Berdasarkan gambar 18 penurunan performansi *memory* pada virtual mesin sangat jauh saat diberikan *load* di atas 50%. *Baremetal ironic* mengalami rata-rata penurunan performansi kurang lebih 32%. Faktor penurunan disebabkan saat *load* yang diberikan membuat L *cache* tersebut menjadi penuh, sehingga pemrosesan data menjadi lambat. Virtual mesin berjalan di atas *hypervisor* atau KVM.

KVM juga perlu waktu untuk melakukan translasi virtual *memory* terhadap *memory* fisik (Kotra, *et al.*, 2017). *Baremetal ironic* yang lebih optimal disebabkan tidak adanya translasi virtual *memory* terhadap *memory* fisik.

**Jitter**



**Gambar 20. Grafik Rata - Rata Jitter Virtual Mesin**

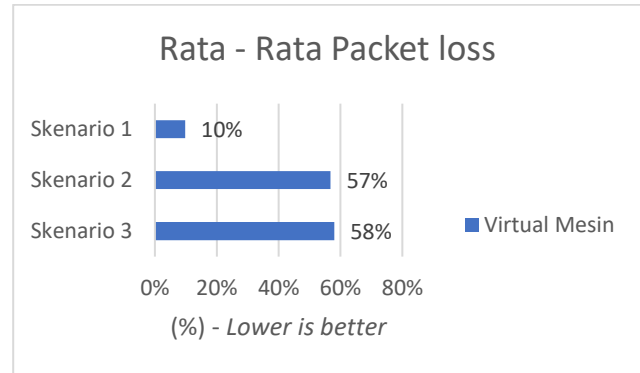


**Gambar 21. Grafik Rata - Rata Jitter Baremetal Ironic**

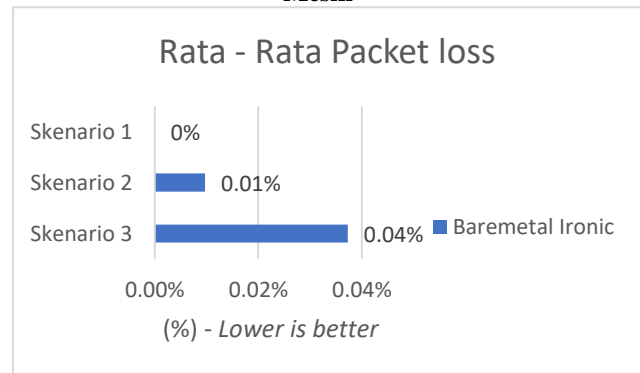
*Jitter* merupakan suatu paket pada transmisi data diakibatkan oleh variasi-variasi dalam waktu *delay*, dalam waktu pengolahan data dan waktu penghimpunan ulang pada suatu paket data. *Jitter* yang diuji sebatas layer 3 pada OSI layer jaringan. Pembacaan *jitter* pada komputer hanya sebatas paket yang diterima. Pengujian *jitter* menggunakan aplikasi Iperf2. Iperf2 akan mencoba melakukan transmisi data antara klien dan server, kemudian dilakukan pengecekan *delay packet* yang diterima pada masing-masing komputer.

Hasil dari pengujian *jitter* atau variasi *delay* pada gambar 20 dan 21, dapat diambil kesimpulan bahwa *jitter* yang dihasilkan bagus < 75 ms berdasarkan acuan dari TIPHON. Virtual mesin mendapatkan *high latency* karena paket transmisi data melewati virtual *interface* selanjutnya diproses oleh KVM. Faktor perbedaan MTU pada virtual *interface* juga menjadi penyebabnya. Gambar 19 merupakan pengujian *Baremetal ironic*, mendapatkan hasil *jitter* yang mendekati 0 ms. Hal tersebut disebabkan *packet* data tidak melewati virtual *interface*.

**Packet Loss**



**Gambar 22. Grafik Rata - Rata Packet loss Virtual Mesin**



**Gambar 23. Grafik Rata - Rata Packet loss Baremetal Ironic**

Pengukuran terakhir pada penelitian ini merupakan *packet loss*. *Packet loss* merupakan suatu parameter yang menggambarkan kondisi total paket hilang saat terjadi pertukaran data antar

pengirim maupun penerima, disebabkan oleh *packet collosion* maupun *congestion* pada jaringan. *Congestion* pada jaringan dapat disebabkan jika *packet* yang dikirim tidak diproses oleh penerima maka *packet* tersebut akan tersimpan sementara pada perangkat *router*. *Buffer packet* pada perangkat *router* akan penuh menyebabkan *packet* akan dilakukan penghapusan, sehingga perlu transmisi ulang. Lambatnya proses pada sisi penerima, disebabkan sedang banyaknya beban yang sedang berjalan pada mesin penerima.

Gambar 22 berdasarkan referensi TIPHON, virtual mesin pada Nova Compute Node saat tidak diberikan *background load* dapat dikatakan bagus. Saat adanya *background load* pada skenario pengujian dua maupun skenario pengujian tiga, virtual mesin pada nova compute node termasuk kategori buruk. Virtual mesin terjadi *packet loss* yang cukup banyak karena data harus melewati *hypervisor*. Penambahannya *background load* menyebabkan adanya waktu tambahan pada pemrosesan data yang dilewatkan *hypervisor*. *Packet* yang tertunda untuk diproses akan dibuang, dan perlu retransmisi ulang. MTU juga menjadi faktor *packet loss* yang terjadi pada virtual mesin. Minimnya *packet loss* pada *baremetal ironic* disebabkan karena tidak ada data yang melewati *hypervisor* dan MTU yang dimiliki bernilai standar.

## Pembahasan

Openstack *flexible* dalam melakukan *provisioning* sebuah perangkat dengan *baremetal provisioning* maupun virtualisasi. Keduanya bagus, namun tergantung saat beban yang diberikan nanti. Penggunaan virtual mesin sangat efisien karena

adanya *resource sharing*, pengguna dapat menjalankan beberapa *guest os* pada sebuah perangkat keras. Namun, beban yang diberikan melebihi setengah dari kapasitas mesin, maka dari itu perlu solusi yang harus dipikirkan untuk menghindari penurunan performa yang signifikan.

## PENUTUP

### Kesimpulan

Tujuan utama dari penelitian ini mengevaluasi performansi CPU, *Memory*, *Disk I/O*, *Jitter* dan *packet loss* pada perangkat *baremetal ironic* maupun virtual mesin dengan tiga skenario pengujian. Berdasarkan pengujian dan analisis yang telah dilakukan, didapatkan kesimpulan sebagai berikut:

1. CPU *processing time* pada *baremetal ironic* memiliki *delay* yang lebih rendah dari virtual mesin. Nilai penurunan rata-rata skenario satu dan dua, *baremetal ironic* dengan nilai selisih 0,001 ms dan virtualisasi nova dengan nilai selisih 0,09662 ms.
2. Pengujian *throughput*, *baremetal ironic* lebih baik dibandingkan dengan virtual mesin. Nilai selisih penurunan skenario satu dan dua, *baremetal ironic* dengan nilai 7,1 Mbits/sec dan virtual mesin dengan nilai 441,7 Mbits/sec.
3. Pengujian *memory*. Selisih nilai penurunan performa *baremetal ironic* lebih kecil dari pada virtual mesin. Nilai selisih penurunan skenario satu dan dua, *baremetal ironic* dengan nilai 72,3 Mib/s dan virtual mesin dengan nilai 2470,4 Mib/s.
4. Pengujian *Disk I/O*. Performa *baremetal ironic* lebih baik dari virtual mesin dengan nilai selisih penurunan skenario satu dan skenario dua



pada *disk I/O read* , *baremetal ironic* dengan nilai 1,17 Mib/s dan virtual mesin dengan nilai 3,72 Mib/s.

5. Pengujian *jitter*, *baremetal ironic* lebih baik dibandingkan dengan virtual mesin. Nilai selisih penurunan skenario satu dan dua, *baremetal ironic* dengan nilai 0,0023 ms dan virtual mesin dengan nilai 3,41 ms.

6. Pengujian terakhir dari penelitian ini adalah *packet loss*. *Packet loss* pada *baremetal ironic* relatif 0 %. Virtual mesin penurunan cukup signifikan dari skenario satu dan skenario dua dengan nilai selisih 47 %. *Baremetal ironic* lebih unggul dalam menangani *packet* dibandingkan virtual mesin.

Poin–poin di atas dapat disimpulkan dari setiap skenario pengujian, jika penggunaan yang memiliki komputasi yang tinggi dan sangat padat, penggunaan *baremetal* sudah sangat layak karena penurunan disetiap skenario tidak terlalu jauh dibandingkan dengan virtual mesin.

## Saran

Solusi alternatif untuk membangun *cloud computing* dengan Openstack. Openstack sudah mendukung dalam *provisioning hardware* dengan *baremetal ironic* maupun mengefisiensikan alat dengan bantuan virtual mesin dengan bantuan Nova. Virtual mesin maupun *baremetal* semua dikontrol dengan Openstack. Solusi bagus untuk diterapkan pada bisnis untuk membangun *data center*.

Batasan untuk saran ke depan, perlu konfigurasi KVM untuk mengaktifkan KVM CPU *Pinning* untuk mengoptimalkan virtual mesin dalam CPU *Processing*. Pengujian *network*, mengalami kendala

pada MTU yang tidak sesuai dengan standar. Perlu investigasi lebih dalam untuk memperbaiki hal tersebut. Berharap ada perbaikan *software* ke depannya.

Penelitian ini, instalasi Openstack dilakukan dengan teknik manual instalasi. Ada cara lain dalam membangun Openstack dengan bantuan Packstack. Packstack memberikan kemudahan untuk pengguna untuk melakukan *deploy* Openstack dengan satu perintah saja. Packstack merupakan kumpulan dari beberapa *script* , dijadikan satu perintah untuk melakukan *deploy* Openstack.

Secara keseluruhan, konsep menggunakan virtual mesin tidak dapat dihilangkan. Virtual mesin sangat bagus untuk mengefisiensikan alat. Teknik virtualisasi membuat sebuah perangkat dapat menjalankan beberapa virtual mesin. Penggunaan *baremetal* juga baik jika ada keperluan *high computing intensity*. Tergantung dalam penggunaan tugas yang diberikan. Openstack dapat melakukannya secara *flexible* terhadap *resource* yang tersedia. Penerapan Openstack perlu konsep *planning deployment hardware* dan efisiensi yang baik untuk menghasilkan penggunaan maksimal dan didunia bisnis meningkatkan pendapatan pemilik layanan *cloud computing*.

## UCAPAN TERIMA KASIH

Puji dan syukur selalu penulis panjatkan kepada Allah SWT berkat atas ridho-nya sehingga penulis dapat menyelesaikan penelitian ini. Penulis juga tidak lupa ucapkan terima kasih kepada pihak–pihak yang membantu penulis dalam membantu penyusunan jurnal ini, doa, maupun kritik pada penelitian ini, antara lain:

1. Bapak Joe, Ibu Asian, Adik Joevy Arzelia selaku anggota keluarga dalam memberikan dukungan serta doa sehingga penulis dapat menyelesaikan penelitian ini.
2. Bapak Agus Virgono, Ir., M.T. selaku pembimbing I dalam mengarahkan praktik dari pengerjaan penelitian, ilmu, masukan dalam pengerjaan penelitian serta sabar dalam membimbing penulis untuk menyelesaikan penelitian ini.
3. Ibu Ridha Muldina Negara, S.T, M.T. selaku pembimbing II dalam mengarahkan memberikan solusi terhadap masalah teknik maupun non teknis dalam menyelesaikan penelitian ini.
4. Bapak Burhanuddin Dirgantoro, Ir., M.T. selaku dosen wali TK-41-04 yang membimbing maupun arahan selama masa perkuliahan.
5. Teman-teman TK-41-04 yang telah menjadi teman seperjuangan dalam proses perkuliahan berlangsung.
6. Teman-teman di Lab Adaptif Network, serta kakak dan adik senior.
7. Seluruh pihak yang tidak dapat disebutkan dalam membantu proses pengerjaan penelitian.

## DAFTAR PUSTAKA

- ITU-T Study Group. (2001). *Recommendation G.1010 - End User Multimedia QoS Categories*. ITU-T.
- Rackspace Cloud Computing. (2021, Januari 6). *Openstack Installation Guide*. (Openstack) Diambil kembali dari Openstack Doc: <https://docs.openstack.org/install-guide/overview.html>
- Openstack. (2019, Desember 19). *Openstack Bare Metal Provisioning Program*. Dipetik

- Desember 11, 2020, dari Openstack Wiki: <https://wiki.openstack.org/wiki/Ironic>
- IBM Corporation. (2020, Oktober 9). *IBM Cloud Bare Metal Servers*. (IBM) Dipetik November 9, 2020, dari IBM: <https://www.ibm.com/cloud/bare-metal-servers/hosting-solutions>
- Cisco System. Inc. (2020). *Cisco Global Cloud Index*. USA: Cisco.
- ETSI. (1999). *TIPHON General Aspects of QoS*. Valbonne: ETSI.
- Zheng, W., Wang, Y., Xia, Y., & Wu, Q. (2017). On Dynamic Performance Estimation of Fault-Prone Infrastructure-as-a-Service clouds. *International Journal of Distributed Sensor Networks*(13).
- Qasaimeh, M., Denolf, K., Lo, J., Vissers, K., Zambreno, J., & Jones, P. H. (2019). Comparing Energy Efficiency of CPU, GPU and FPGA Implementations for Vision Kernels. *IEEE*. Las Vegas.
- Podzimek, A., Bulej, L., Chen, L. Y., Binder, W., & Tuma, P. (2015). Analyzing the Impact of CPU Pinning and Partial CPU Loads on Performance and Energy Efficiency. *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. Shenzhen: IEEE.
- Kumar, R., Gupta, N., Charu, S., & Jain, K. (2014). Open Source Solution for Cloud Computing Platform Using OpenStack. *International Journal of Computer Science and Mobile Computing*. 3, hal. 89-98. IJCSMC.
- Kotra, J. B., Kim, S., Madduri, K., & Kandemir, M. T. (2017). Congestion-aware memory management on NUMA platforms: A VMware ESXi case study. *2017 IEEE International Symposium on Workload Characterization (IISWC)*. Seattle: IEEE.
- Kang, M., Kang, D.-I., Walters, J. P., & Crago, S. P. (2017). A Comparison of System Performance on a Private OpenStack Cloud and Amazon EC2. *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. 10. Honolu: IEEE.

- Felter, W., Ferreira, A., Rajamony, R., & Rubio, J. (2015). An updated performance comparison of virtual machines and Linux containers. *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. Philadelphia: IEEE.
- Browne, P. (2019). *OpenStack Ironic - Multi-Tenant*. Research Computing Platforms.
- Awasthi, A., & Gupta, R. (2016). Multiple hypervisor based Open Stack cloud and VM migration. *2016 6th International Conference - Cloud System and Big Data Engineering*. India.
- Red Hat. Inc. (2014, Agustus 16). *Understanding cloud computing*. (Redhat) Dipetik Februari 8, 2020, dari Redhat: <https://www.redhat.com/en/topics/cloud>
- Kominos, C. G., Seyvet, N., & Vandikas, K. (2017, March). Bare-metal, virtual machines and containers in OpenStack. In *2017 20th conference on innovations in clouds, internet and networks (icin)* (pp. 36-43). IEEE.