RETROTEKIN
JURNAL REKAYASA TROPIS, TEKNOLOGI, DAN INOVASI
e-journals2.unmul.ac.id/index.php/retrotekin
e-ISSN: 3031 - 2531

RETROTEKIN Vol. 1 No. 2, November 2023, 40-56, DOI:

Performa Klaster *Hadoop MapReduce* pada *Private Cloud Computing* untuk Komputasi *Skyline Query*

Annisa Octavyanti Hakim[1)], Heri Wijayanto[1)], I Gde Putu Wirarama[1)]
[1)] Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram
E-mail: annisaoctavya@gmail.com

ABSTRAK

Untuk mengoptimalkan pemrosesan data besar dengan *Hadoop*, komputasi awan menyediakan infrastruktur yang mudah digunakan, menggabungkan layanan *private cloud* dengan *Infrastructure as a Service* (*IaaS*). Dalam penelitian ini, terdapat proses karakterisasi dan penilaian kinerja eksekusi data besar pada instan klaster virtual *Hadoop MapReduce* yang dibangun di *private cloud* Universitas Mataram. Dengan algoritma *Skyline Query*, klaster diuji dengan variasi data, mesin, dan ukuran blok *HDFS* pada 3 jenis data sintetis: *anti-correlated*, *correlated*, dan *independent*. Parameter waktu eksekusi digunakan untuk membandingkan hasil dengan klaster *Hadoop* pada infrastruktur fisik. Hasil pengujian klaster *private cloud* menunjukkan peningkatan waktu komputasi saat data meningkat dari 1,5 juta menjadi 12 juta pada 4 mesin: data *anti-correlated* (168%), *correlated* (194%), dan *independent* (126%). Tren serupa terjadi pada klaster *Hadoop* fisik. Pada skenario lainnya, klaster *private cloud* menunjukkan kinerja yang lebih baik dengan penambahan mesin hingga 7, sementara klaster *Hadoop* fisik mengalami *overhead communication* antar-*node* ketika mesin diskalakan menjadi 7 mesin. Pemrosesan data 12 juta dengan ukuran blok *HDFS* 512 MB dan 7 mesin merupakan *block size* paling optimal karena menghasilkan waktu eksekusi terpendek. Berdasarkan uji statistik t menggunakan rata-rata waktu komputasi, disimpulkan bahwa klaster *Hadoop* di *private cloud* dengan spesifikasi *Intel(R) Xeon (R) E3-1225 v5 @ 3,30 GHz RAM 16 GB* lebih unggul dalam mengeksekusi aplikasi *Skyline* dibandingkan klaster *Hadoop* fisik dengan spesifikasi *Intel Core i5 CPU @ 3,00GHz RAM 4GB*.

Kata Kunci: *Hadoop MapReduce*, *cloud computing, private cloud*, IaaS, *skyline query* terdistribusi, *MR-BNL*

ABSTRACT

To optimize significant data processing through Hadoop, cloud computing offers user-friendly infrastructure by combining private cloud services and Infrastructure as a Service (IaaS). In this research, the characterization and evaluation of large data execution on virtual Hadoop MapReduce clusters within Mataram University's private cloud. Using the Skyline Query algorithm, the cluster is tested with variations in data, machines, and HDFS block sizes on three synthetic data types: anti-correlated, correlated, and independent. Execution time parameters are used to compare results with Hadoop clusters on physical infrastructure. Test outcomes of the private cloud cluster exhibit increased completion time as data scales from 1,5 million to 12 million on four machines: anti-correlation data (168%), correlation data (194%), and independent data (126%). A parallel performance trend is observed in the physical Hadoop cluster. In a separate scenario, the private cloud cluster demonstrates superior performance with the addition of up to 7 machines, while the physical Hadoop cluster encounters communication overhead with 7 machines. Processing 12 million data using an HDFS block size of 512 MB and 7 machines produces the shortest execution time. Based on t-statistical tests concerning average processing time, the conclusion is that the Hadoop cluster within the private cloud, featuring Intel(R) Xeon (R) E3-1225 v5 @ 3,30 GHz and 16 GB RAM, surpasses Skyline application execution compared to a physical Hadoop cluster with specifications of Intel Core i5 CPU @ 3,00 GHz and 4 GB RAM

Keyword: Hadoop MapReduce, cloud computing, private cloud, IaaS, distributed skyline query, MR-BNL

1. Pendahuluan

Teknologi internet mengalami perkembangan yang sangat pesat sejak kemunculannya pada tahun 1960-an (Nuriadin dkk., 2021). Pengguna internet yang terus meningkat bersamaan dengan layanan aplikasi yang semakin beragam menjadi pemicu konsumsi data yang terus bertambah tanpa henti. Data dalam jumlah besar tanpa mekanisme pengelolaan yang baik hanya akan menjadi objek pasif yang tidak dapat dimanfaatkan lagi. Oleh karena itu, dibutuhkan suatu arsitektur yang mumpuni untuk mengelola data besar (Subagya dkk., 2021).

Salah satu *framework big data* yang paling populer saat ini adalah *Hadoop*. *Hadoop* merupakan sebuah arsitektur yang menggunakan konsep paralel terdistribusi untuk mengolah data bervolume besar menggunakan model pemrograman *MapReduce* melalui sekumpulan komputer yang terhubung satu sama lain melalui jaringan (klaster) (Ryanto, 2017). *MapReduce* membagi data menjadi banyak pecahan yang mana tiap pecahan akan diproses di tiap *node* pada sebuah klaster.

Pada saat ini, mengelola data besar menggunakan *Hadoop* memiliki tantangan tersendiri dalam hal penyediaan, pengaturan, dan perawatan infrastruktur skala besar. Diperlukan biaya investasi awal dalam hal infrastruktur, operasional, pakar TI dan pemeliharaan berkelanjutan yang tentunya tidak sedikit. Hal ini membuat implementasi *Hadoop* dengan *physical machine* terbatas dilakukan. Untuk menyelesaikan tantangan ini, *cloud computing* menawarkan konsep pengolahan sumber daya komputasi melalui jaringan internet (*cloud*) dengan biaya sebesar yang digunakan pengguna saja. Ini dapat membantu pengguna untuk lebih berfokus dalam pekerjaannya dibandingkan mengkhawatirkan masalah ketersediaan infrastruktur, sumber daya, dan pakar TI.

Pada *cloud computing*, terdapat beberapa model penyebaran, salah satunya yakni *private cloud*. Layanan ini banyak digunakan oleh pengguna seperti perusahaan dan universitas yang menginginkan *control* eksklusif. *Private cloud* memberikan kontrol penuh kepada penggunanya dengan memberikan akses khusus ke jaringan dan infrastruktur yang dapat dikostumisasi. Implementasi *private cloud* dengan *Infrastructure as a Service* (*IaaS*) akan disediakan dalam bentuk *virtual instance* atau infrastruktur virtual yang bisa diminta (*request*) sesuai kebutuhan internal. Infrastruktur virtual ini bekerja seperti mesin dengan komponen penyimpanan, RAM, *disk space*, sistem operasi, *network,* dan kekuatan pemrosesan *CPU* (Subramanian & Gouda, 2015). *IaaS* dengan *virtual machine* dapat dikelola secara fleksibel dan secara teknis dapat menggantikan *server* fisik, sumber daya pusat data, *network tools,* dan komponen fisik lainnya (Prabowo dkk., 2015).

Keunggulan *private cloud* ini dapat dimanfaatkan untuk mengatasi keterbatasan penyediaan mesin fisik dengan spesifikasi yang ideal untuk komputasi *big data* berskala kompleks pada laboratorium Prodi Teknik Informatika, Universitas Mataram. Saat ini, Laboratorium 2, Prodi Teknik informatika memiliki PC sejumlah 21 buah dengan spesifikasi *Processor Intel(R) Core (TM)* i5-9500 CPU @ 3,00GHz, RAM 4,00 GB (3,78 GB *usable*) dan memori SSD 500 GB. Berdasarkan (Azizah & Saptono, 2020), spesifikasi tersebut merupakan spesifikasi *minimum* untuk komputasi *Hadoop*. Sehingga untuk komputasi *Hadoop* yang lebih baik, teknologi virtualisasi dengan *private cloud* akan sangat baik diimplementasikan.

Fokus penelitian ini terletak pada unjuk kerja klaster *virtual* pada *private cloud* yang ada di *server* Universitas Mataram dalam menyelesaikan komputasi *Hadoop MapReduce*. Ekosistem *Hadoop* yang dibangun, akan digunakan untuk mengeksekusi sejumlah data berukuran besar dengan beberapa skenario pengujian salah satunya algoritma *Skyline Query* guna menilai kinerja klaster *Hadoop*. *Skyline Query* merupakan metode pencarian sekumpulan objek penting yang memiliki kriteria lebih baik dari pada objek lainnya dalam himpunan data. Algoritma ini dipilih karena merupakan algoritma yang memiliki kompleksitas yang sangat bergantung pada jumlah dimensi dan besar *dataset* yang digunakan (Wibawa dkk., 2018). Sebagai bahan perbandingan, *requirement* serupa juga akan dijalankan pada mesin fisik. Kedua implementasi ini akan dibandingkan kinerjanya dalam hal waktu kecepatan eksekusi atau *running time* saat menjalankan komputasi.

2. Metode Penelitian

A. *Klaster Hadoop*

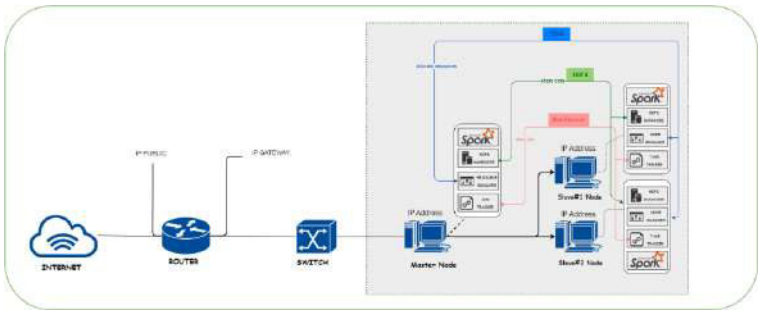
Pada penelitian ini, Klaster *Hadoop* dibangun di atas layanan *Infrastructure as a Service* pada *Private Cloud* menggunakan *server* Laboratorium 2, Laboratorium Komunikasi Data dan Sistem

Tertanam yang diletakan di UPT PUSTIK Universitas Mataram. Klaster yang akan diluncurkan berbentuk *instance virtual* atau *virtual machine* sebanyak 1 buah *node* sebagai inisialisasi awal. Setiap *node* akan dilakukan instalasi *Linux Ubuntu* 22.04.1 LTS sebagai basis sistem operasi Klaster *Hadoop*.

Distribusi *Klaster Hadoop* berbentuk mesin virtual akan dibuat menggunakan komputer *server* PUSTIK Universitas Mataram dengan spesifikasi:

1. *Processor* : *Intel*(R) *Xeon* (R) E3-1225 v5, 4 Core, CPU @3,30 GHz
2. RAM : 16 GB
3. Memori : HDD, 1TB

Sebagai bahan perbandingan, dilakukan konfigurasi *Hadoop* dengan *requirement* yang serupa pada klaster virtual menggunakan komputer fisik pada Laboratorium 2, Laboratorium Komunikasi Data dan Sistem Tertanam Prodi Teknik Informatika. Klaster komputer ini akan diberikan perlakuan yang sama dengan *klaster virtual*. Arsitektur Klaster *Hadoop* yang akan dibangun ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur Klaster *Hadoop*

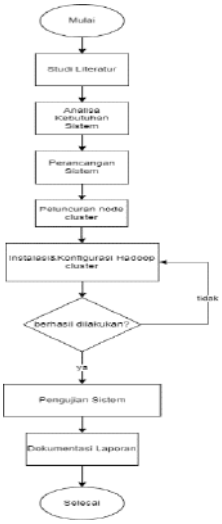
B. Data

Pada penelitian ini, data yang digunakan adalah data sintetis. Data sintetis mengacu pada data artifisial yang dibuat (*generate*) dengan suatu *tools* menggunakan suatu algoritma untuk melakukan pengujian tertentu. Dalam distribusi data, penulis menggunakan berbagai jenis data sintetis yakni *correlated*, *uncorrelated,* dan *independent* dengan masing-masing berukuran 1 GB. Ketiga distribusi data yang dimaksud seperti berikut (Börzsönyi dkk., 2001):

1. *Independent*: semua nilai atribut dibuat secara *independent* menggunakan sebuah distribusi *uniform*.
2. *Correlated*: mengacu pada data yang memiliki korelasi atau terhubung satu sama lainnya. Data ini merepresentasikan lingkungan di satu titik yang tidak hanya baik pada satu dimensi, namun juga dimensi lainnya
3. *Uncorrelated*: data yang tidak saling berhubungan satu sama lain. Dalam artian, data yang dimiliki hanya baik pada satu dimensi saja, tidak pada dimensi lainnya.

C. Alur Penelitian

Subbab ini berisi langkah-langkah yang akan dilakukan peneliti dalam menyelesaikan pokok permasalahan yang diangkat pada penelitian ini. Diagram alir penelitian dapat dilihat pada Gambar 2.



Gambar 2 Diagram alir penelitian

D. Pengujian Sistem

Dalam penelitian ini, dikarenakan fokus penelitian terletak pada pengujian performa Klaster *Hadoop* menggunakan layanan IaaS *Private Cloud*, maka rancangan skenario pengujian perlu disusun. Pengujian akan dilakukan menggunakan beberapa kasus seperti pada Tabel 1.

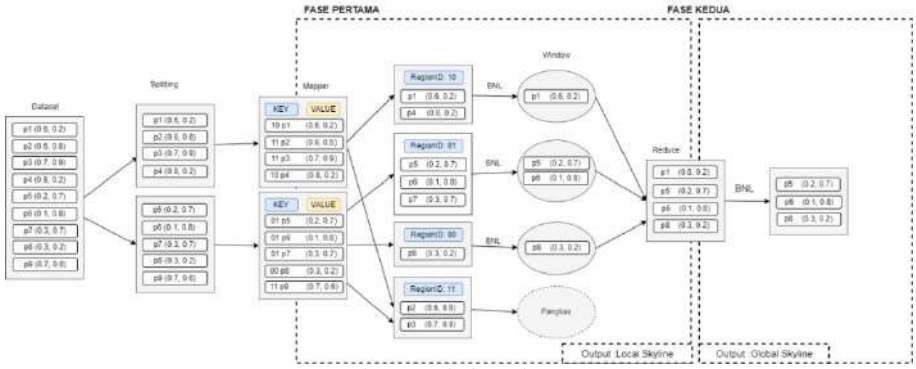
Tabel 1. Rancangan pengujian performa *klaster*

Case	***Step* Pengujian**		***Expected Result***
Komputasi *Skyline* dengan *MR-BNL*	1.	*Input* data	Sistem mampu menghasilkan *local* dan *global skyline*
	2.	*Splitting* data sama rata sebesar 2^d pada mesin *mapper*	
	3.	Berikan *flag d*-bit	
	4.	*Scanning local skyline* dengan *MR-BNL*	
	5.	*Local skyline* digabungkan menggunakan *flag*	
	6.	*Reducing* dengan *MR-BNL*	
	7.	*Global skyline* dihasilkan	
Modifikasi ukuran file	1.	Dari total ukuran *dataset* yang dimiliki, eksekusi dilakukan secara berkala dengan ukuran *file* pertama sekitar 100 MB atau jumlah data 1,5 juta.	Kecepatan *excecution time* Kecepatan waktu eksekusi semakin melambat seiring penambahan ukuran *file*
	2.	Ukuran *file* akan terus ditambah hingga data berukuran sekitar 1 *GB* dengan pecahan 200 *MB* (2,5 juta), 400 *MB* (5 juta), 800 *MB* (10 juta) hingga 1 *GB* (12 juta).	
Variasi jumlah *node*	1.	*Hadoop MapReduce* dijalankan dengan 1 *node* pada sebuah ukuran *file*.	Kecepatan *excecution time Hadoop MapReduce* meningkat seiring penambahan *node*.
	2.	*Node* kemudian ditambah mulai dari 2 hingga 7 *node*.	
Variasi ukuran *block HDFS*	1.	*Blocksize HDFS* dengan blok lebih kecil dari *default* (128 *MB*) yakni 64 *MB*	Terjadi peningkatan eksekusi *write time* seiring penambahan ukuran *block HDFS*.
	2.	Penambahan ukuran *block* menjadi 128 *MB* dan 256 *MB*, 512 *MB*	

Pengujian dilakukan untuk menilai performa klaster virtual terhadap suatu keadaan tertentu. Untuk memberikan perbandingan kinerja yang jelas dan realistis, klaster virtual akan dibandingkan dengan klaster fisik dalam *running jobs* dan diberikan skenario pengujian yang sama. Setiap pengujian akan diulang 3 kali dan hasilnya akan dirata-ratakan, kemudian direpresentasikan dalam plot grafik.

Sesuai Tabel 1, dalam menilai respon klaster, perlu disiapkan beberapa skenario pengujian yang dilakukan secara bertahap. Setiap pengujian yang dilakukan menggunakan algoritma MR-BNL dengan klaster yang diinisialisasikan untuk mencari *Skyline* pada data berukuran besar. Pada *Skyline Block*

Nested Loop (BNL) proses *MapReduce* untuk menghasilkan *local skyline* dan *global skyline* terdiri atas dua fase. Fase pertama adalah distribusi partisi data ke *mapper* dan fase kedua adalah komputasi *local skyline* pada setiap partisi dengan BNL sehingga menghasilkan *global skyline*. Komputasi *Skyline* menggunakan algoritma MR-BNL akan ditunjukkan pada Gambar 3.



Gambar 3. *Skyline* dengan MR-BNL

3. Hasil dan Pembahasan

Bagian ini menyajikan temuan yang didapatkan setelah menjalankan setiap percobaan. Ditampilkan hasil performa klaster *Hadoop MapReduce* setelah menjalankan komputasi *Skyline* MR-BNL dengan beragam kondisi di atas *private cloud* dan di atas komputer fisik. Klaster diuji dengan beragam variasi kondisi seperti perubahan ukuran file, perbedaan jumlah mesin hingga variasi ukuran blok data HDFS. Penulis telah menggunakan tiga jenis input data sintetik dan menggunakan parameter konfigurasi yang sama untuk perbandingan yang realistis. Untuk setiap skenario pengujian, waktu *running jobs* ditulis menggunakan format detik berdasarkan tiga kali percobaan pengujian. Grafik untuk setiap skenario pada klaster fisik maupun klaster virtual *private cloud* di-*plot* untuk memvisualisasikan performa komputasi.

Tabel 2. Spesifikasi Klaster

Klaster Hadoop Private Cloud					Klaster Hadoop Fisik				
Hostname	Spesifikasi				Hostname	Spesifikasi			
	Processor	Core(s)	RAM	SSD		Processor	Core(s)	RAM	SSD
Master	Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz	2	2.7 GB	35 GB	Master	Intel Core i5 CPU @ 3.00GHz	4	2.7 GB	80 GB
Slave1	Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz	1	2 GB	25 GB	Slave1	Intel Core i5 CPU @ 3.00GHz	3	2 GB	50 GB
Slave2	Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz	1	2 GB	25 GB	Slave2	Intel Core i5 CPU @ 3.00GHz	3	2 GB	50 GB
Slave3	Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz	1	2 GB	25 GB	Slave3	Intel Core i5 CPU @ 3.00GHz	3	2 GB	50 GB
Slave4	Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz	1	2 GB	25 GB	Slave4	Intel Core i5 CPU @ 3.00GHz	3	2 GB	50 GB
Slave5	Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz	1	2 GB	25 GB	Slave5	Intel Core i5 CPU @ 3.00GHz	3	2 GB	50 GB
Slave6	Intel(R) Xeon (R) E3-1225 v5 @ 3.30 GHz	1	2 GB	25 GB	Slave6	Intel Core i5 CPU @ 3.00GHz	3	2 GB	50 GB

A. Hasil Pengujian dan Analisis Hasil Klaster Virtual *Hadoop Private Cloud*

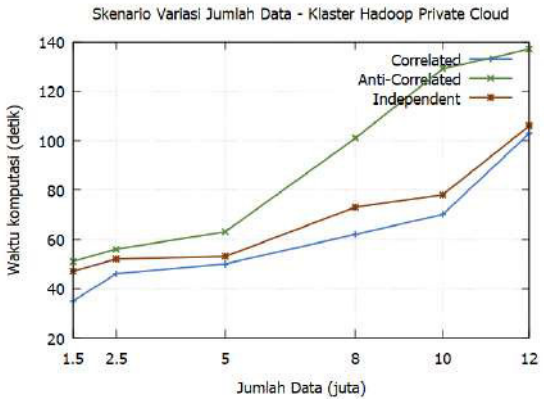
1. Skenario Variasi Jumlah Data atau Ukuran File

Menjalankan beragam variasi ukuran data dengan 4 mesin tidak membutuhkan waktu yang lama. Mesin membutuhkan waktu rata-rata yang berbeda untuk setiap jenis dan ukuran *dataset*. Dari ketiga *dataset*, data sintetis *anti-correlated* cenderung memberikan waktu komputasi yang lebih tinggi dibanding yang lainnya. Hal ini disebabkan data *anti-correlated* menghasilkan lebih banyak titik *skyline* dibandingkan kedua *dataset* lainnya, sehingga waktu pencarian *skyline global* menghasilkan waktu yang lebih lama. Hal ini dapat dilihat pada Tabel 3.

Tabel 3. Hasil komputasi skenario 1 - Klaster *Hadoop Private Cloud*

Jumlah Data (Juta)	Jenis *Dataset*		
	Correlated	*Anticorrelated*	*Independent*
1.5	35	51	47
2.5	46	56	52
5	50	63	53
8	62	101	73
10	70	129	78
12	103	137	106

Berdasarkan Gambar 4, secara garis besar ketiga *dataset* menunjukkan peningkatan waktu komputasi seiring dengan penambahan volume data. Penambahan volume data mengakibatkan semakin banyak kandidat *skyline* yang harus di bandingkan satu persatu untuk mencari *skyline global* menggunakan algoritma *Block Nested Loops*, sehingga *execution time* semakin lama. Urutan data yang menghasilkan titik *skyline* paling banyak terletak pada data *anti-correlated* diikuti oleh data *independent* kemudian data *correlated*. Hal ini menyebabkan data *anti-correlated* membutuhkan waktu komputasi paling lama dibanding kedua data lainnya. Sementara, data *correlated* menjadi data dengan komputasi tercepat karena hasil akhir *skyline global* yang sedikit.



Gambar 4. Hasil komputasi skenario 1 – Klaster *Hadoop Private Cloud*

2. Skenario Variasi Jumlah Mesin

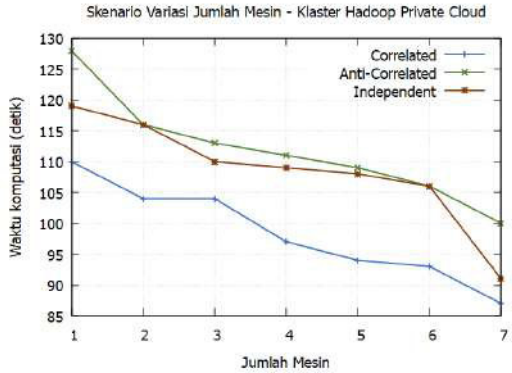
Skenario pengujian ini menampilkan tingkat signifikansi perubahan kinerja klaster secara parsial dari waktu eksekusi sebelum dan sesudah mengalami perubahan jumlah *node*. Pada skenario ini ditetapkan variabel terikat yakni jumlah data berukuran 12 juta, sementara variabel bebas yakni jumlah mesin virtual dari 1 mesin hingga 7 mesin. Pemilihan jumlah data ini dilakukan setelah mengamati peningkatan kinerja klaster pada ketiga *dataset* di skenario sebelumnya. Tabel 4 menunjukkan jumlah mesin memberi peningkatan kinerja yang secara umum cukup signifikan dalam mengeksekusi ketiga jenis *dataset* ketika klaster ditingkatkan dari 1 mesin hingga 7 mesin.

Tabel 4. Hasil Komputasi Skenario 2 - Klaster *Hadoop Private Cloud*

Jumlah Mesin	Jenis *Dataset*		
	Correlated	*Anticorrelated*	*Independent*
1	110	128	119
2	104	116	116
3	104	113	110
4	97	111	109
5	94	109	108
6	93	106	106
7	87	100	91

Pada Gambar 5, tampak bahwa waktu menyelesaikan komputasi *Hadoop MapReduce* pada aplikasi *skyline* secara umum berkurang secara konstan seiring jumlah mesin virtual yang diskalakan. Penggunaan lebih banyak mesin akan mempercepat kinerja komputasi paralel dalam memproses aplikasi *Skyline MR-BNL*. Hal ini disebabkan efisiensi pembagian tugas ke dalam bagian-bagian yang lebih kecil pada beberapa prosesor memungkinkan keseluruhan proses komputasi diselesaikan dengan lebih cepat. Efisiensi komputasi paralel pada Gambar 5 menunjukkan percepatan yang ideal, yang mana peningkatan kecepatan komputasi terjadi seiring dengan penambahan jumlah prosesor yang digunakan secara paralel. Pada ketiga jenis *dataset* yang diujikan, data *anti-correlated* membutuhkan waktu yang paling lama, kemudian menyusul *dataset*

independent dengan perbandingan waktu komputasi sedikit lebih cepat, namun tidak jauh berbeda. Sementara data *correlated* mampu diselesaikan dengan jauh lebih cepat oleh klaster dibanding kedua *dataset* lainnya.



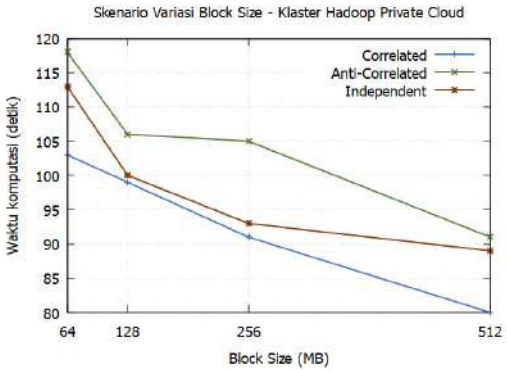
Gambar 5. Hasil komputasi skenario 2 - Klaster *Hadoop Private Cloud*

3. Skenario Ukuran Blok HDFS

Gambar 6 menunjukkan bahwa *block size* dapat mempengaruhi performa kecepatan *Hadoop MapReduce* dalam mengeksekusi aplikasi *Skyline MR-BNL* untuk setiap percobaan menggunakan file 1,06 *GB* pada *block size* yang bervariasi. Pada skenario ketiga, ketiga *dataset* dengan jumlah data 12 juta atau 1,06 *GB* akan dipotong menjadi beberapa *block* sesuai *block size* 64 *MB*, 128 *MB*, 256 *MB* dan 512 *MB*. Gambaran potongan *block* tersebut dapat dilihat pada Gambar 7. Dari gambar tersebut, dapat terlihat bahwa file berukuran 1,06 *GB* dengan *block default* (128 *MB*) dipotong menjadi 9 *block* dengan replikasi 3 yang akan tersimpan pada setiap *node*.

Tabel 5. Hasil Komputasi Skenario 3 - Klaster *Hadoop Private Cloud*

Ukuran *Block*	Jenis *Dataset*		
	Correlated	*Anticorrelated*	*Independent*
64 *MB*	103	118	113
128 *MB*	99	106	100
256 *MB*	91	105	93
512 *MB*	80	91	89



Gambar 6. Hasil Komputasi Skenario 3 - Klaster *Hadoop Private Cloud*

Jumlah *block* pada ukuran file 1,06 *GB* dengan *block size* 64 *MB* menghasilkan 17 *block*, lebih banyak dibandingkan pada *block size* 256 *MB* dan 512 *MB* yang masing-masing menghasilkan 5 *block* dan 3 *block*. Jumlah *block* yang semakin sedikit akan mengurangi ukuran *metadata* dari *namenode* sehingga mempercepat proses kerja dari *namenode*. Selain itu, jumlah *block* pada *HDFS Hadoop* menentukan jumlah *task* yang harus dikerjakan oleh *MapReduce*. Jumlah *block* yang sedikit dapat diartikan dengan jumlah *task* yang sedikit pula. Jumlah task yang sedikit dapat memudahkan

scheduler task MapReduce dalam menjadwalkan *task* yang diberikan sehingga mengurangi kerja *scheduler task MapReduce*. Jumlah *task* yang sedikit juga dapat mengurangi waktu komunikasi permintaan *task* antara *scheduler task MapReduce* dengan *Resource Manager* serta *Resource Manager* dengan *Node Manager*. Hal ini tentunya akan berdampak pada kecepatan komputasi *Hadoop MapReduce* yang dijalankan.

Gambar 6, secara garis besar memperlihatkan bahwa penambahan *block size* pada ketiga jenis *dataset* dapat mempercepat proses *MapReduce* pada *Hadoop*. Ketika menggunakan *block size* 64 *MB* dengan 17 potongan *block*, komputasi *Hadoop MapReduce* berjalan paling lambat dibanding menggunakan *block size* 128 *MB*, 256 *MB* dan 512 *MB*. Sementara itu, waktu komputasi tercepat ditunjukkan ketika *block size* 512 *MB* dengan jumlah *block* yang dihasilkan hanya 3 *block*.

Perlu diketahui bahwa pada *Hadoop*, tidak ada aturan dalam memilih *block size* HDFS, itu semua tergantung pada ukuran file yang akan dieksekusi. Maka dari itu, untuk memaksimalkan throughput, perlu disesuaikan ukuran block dengan input data. Jika data yang akan diolah adalah kumpulan data berukuran besar yang tidak bisa diolah menggunakan single machine, disarankan untuk menggunakan block size berukuran 128 MB atau 256 MB atau 512 MB. Dan jika datanya berukuran lebih kecil, maka menggunakan block size yang lebih kecil lagi adalah pilihan yang lebih baik.

Jika sebelumnya telah dijelaskan bahwa memperkecil *block size* akan memperlambat kinerja *name node* dan MapReduce, maka pada kasus tertentu memperbesar *block size* juga akan menyebabkan efisiensi paralelisme berkurang dikarenakan sedikitnya splitting data. Hal ini menyebabkan banyaknya mesin *mapper* yang kurang dimanfaatkan untuk komputasi. Ketika menggunakan block size berukuran besar, potongan block akan semakin kecil, dengan begitu block akan disebarkan pada mesin mapper sesuai jumlah block. Menjalankan komputasi dengan mesin mapper yang lebih sedikit pada beberapa kasus contohnya (Basuki dkk., 2015) akan menyebabkan komputasi melambat. Hal ini disebabkan karena ukuran potongan *block* yang harus diproses menghasilkan *block* terlalu besar dan membebani suatu node dalam mengolahnya.

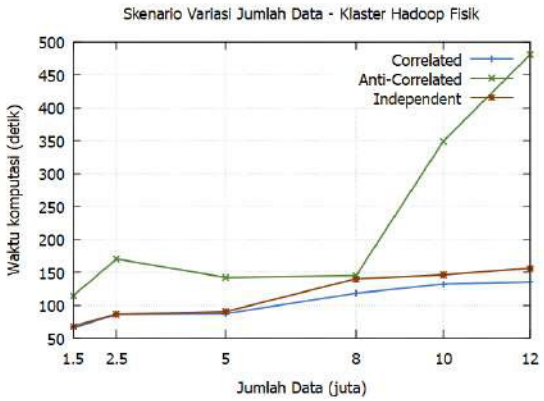
B. Hasil Pengujian dan Analisis Hasil Klaster Hadoop Fisik

1. Skenario Variasi Jumlah Data atau Ukuran File

Menjalankan komputasi *skyline MR-BNL* menggunakan klaster fisik menunjukkan bahwa perlu waktu yang lebih banyak untuk menghasilkan titik *skyline* untuk ukuran data 10 GB dan 12 GB. Menggunakan 4 *node slave*, hasil pengujian 3 jenis data sintetik menunjukkan pengaruh jumlah data terhadap waktu komputasi *Hadoop MapReduce*. Pada Tabel 6 tercantum hasil komputasi *skyline* menggunakan tiga jenis data sintetis dan dikonseptualisasikan dalam Gambar 7.

Tabel 6. Hasil Komputasi Skenario 1 – Klaster *Hadoop* Fisik

Jumlah Data (Juta)	Jenis *Dataset*		
	Correlated	*Anticorrelated*	*Independent*
1.5	65	114	68
2.5	86	170	86
5	87	142	90
8	118	145	140
10	132	349	146
12	135	481	156



Gambar 7. Hasil komputasi skenario 1 – Klaster *Hadoop* Fisik

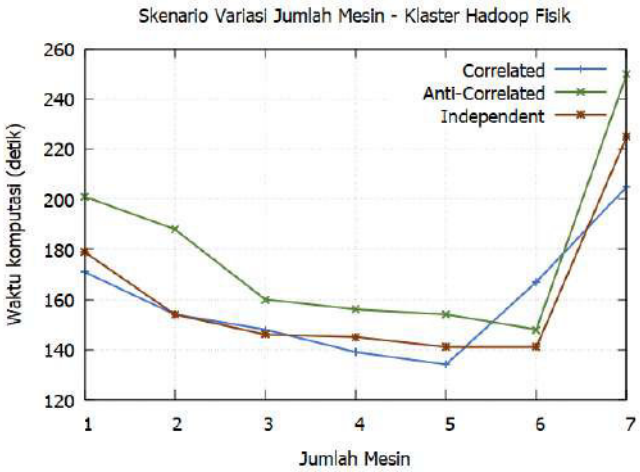
Gambar 7 menggambarkan pengaruh penambahan jumlah data yang secara garis besar mampu menurunkan kinerja klaster seiring penambahan jumlah data. Sesuai dengan jumlah titik *skyline* yang dihasilkan pada ketiga *dataset*, data *anti-correlated* dengan titik *skyline* paling banyak tentunya membutuhkan waktu komputasi paling tinggi dibanding data lainnya. Sementara data *independent* dan *correlated* membutuhkan waktu yang tidak jauh berbeda untuk memproses data. Pada data *anti-correlated*, waktu pencarian *skyline* meningkat sebesar 48% saat data dinaikkan dari 1.5 juta menjadi 2.5 juta. Namun, menambah jumlah data menjadi 5 juta dan 8 juta menyebabkan penurunan dalam waktu pencarian *skyline* sebesar 14%. Ini bisa saja disebabkan karena dampak dari *network bottleneck*, terutama komputasi *Hadoop* yang sangat dipengaruhi oleh kelancaran jaringan. Sementara saat data ditambah menjadi 10 juta dan 12 juta, waktu pencarian *skyline* meningkat kembali hingga 231%.

2. Skenario Vairiasi Jumlah Mesin

Grafik *execution time* dari komputasi *skyline query* dengan variasi jumlah data, menjadi patokan dalam menentukan jumlah data yang akan diproses pada skenario selanjutnya. Skenario pengujian ini menampilkan tingkat signifikansi perubahan kinerja *klaster* secara parsial dari waktu eksekusi sebelum dan sesudah mengalami perubahan jumlah *node*. Pada klaster *Hadoop* fisik, variabel terikat dan bebas ditetapkan sama seperti pengujian variasi jumlah mesin pada klaster *Hadoop Private Cloud*. Tabel 7 menunjukkan jumlah mesin memberikan pengaruh yang besar pada waktu eksekusi data berjumlah 12 juta.

Tabel 7. Hasil komputasi skenario 2 - Klaster *Hadoop* Fisik

Jumlah Mesin	Jenis *Dataset*		
	Correlated	*Anticorrelated*	*Independent*
1	171	201	179
2	154	188	154
3	148	160	146
4	139	156	145
5	134	154	141
6	167	148	141
7	205	250	225



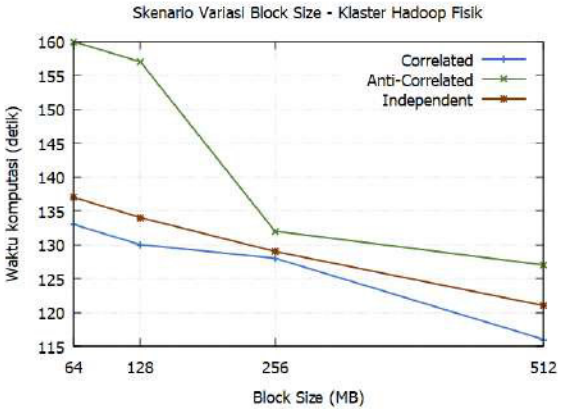
Gambar 8. Hasil komputasi skenario 2 - Klaster *Hadoop* Fisik

Menjalankan ketiga data sintetis di atas klaster fisik memberikan hasil yang berbeda. Umumnya peningkatan jumlah mesin akan mempercepat jalannya program. Namun, menjalankan banyak mesin dalam satu waktu rentan dengan kondisi *overhead* yang menyebabkan naiknya waktu komputasi. Pada Gambar 8, klaster berjalan secara optimal untuk seluruh jenis *dataset* hingga 5 *node*. Namun, ketika mesin ditambah menjadi 6 mesin, respon berbeda didapati dari masing-masing jenis *dataset*. Secara garis besar, klaster menunjukkan kinerja yang tidak optimal ketika menjalankan ketiga *dataset* menggunakan 7 *node*. Adanya peningkatan waktu komputasi ini diduga dipicu oleh kompleksitas pada proses distribusi data, sinkronisasi antar *node* dan komunikasi antar *daemon Hadoop* ketika jumlah mesin ditambah serta *network bottleneck*. *Overhead* menyebabkan waktu komputasi menjadi berlebih dan klaster bekerja tidak optimal. Terjadi peningkatan rata-rata waktu komputasi sebesar 50.3% pada ketiga *dataset*. Waktu komputasi berlebih karena overhead antar node juga terjadi pada penelitian (Made dkk., 2023; Achahbar & Abid, 2014). Pada (Achahbar & Abid, 2014), *overhead* terjadi ketika mengolah data berukuran 100 MB, 1 GB, 10 GB dan 100 GB menggunakan 7 mesin dan 8 mesin untuk *TestDFSIO- Read Performance*. Selain itu, kondisi *overhead* juga terjadi pada 8 node virtual *VMware ESXi* yang diduga disebabkan memori berlebih, tingkat latensi yang tinggi, dan kekurangan sumber daya ketika melakukan *Terasort* 30 *GB*. Menurut (Ivanov dkk., 2014), *overhead* pada mesin virtual terjadi berkisar pada 2-10% tergantung jenis aplikasinya. Namun, ada juga kasus yang mana klaster *Hadoop* yang divirtualisasikan memiliki komputasi yang lebih baik dari klaster *Hadoop* fisik karena sumber daya yang lebih baik. Salah satu kasusnya ialah penelitian ini. Klaster *Hadoop* virtual dibangun dengan spesifikasi sumber daya mesin yang lebih baik dibandingkan dengan klaster *Hadoop* fisik, sehingga memiliki kinerja yang lebih baik. Spesifikasi kedua klaster dapat dilihat pada Tabel 1.

3. Skenario Variasi Ukuran Blok HDFS

Tabel 8. Hasil komputasi skenario 3 - Klaster *Hadoop* Fisik

Ukuran *Block*	Jenis *Dataset*		
	Correlated	*Anticorrelated*	*Independent*
64 *MB*	133	160	137
128 *MB*	130	157	134
256 *MB*	128	132	129
512 *MB*	116	127	121



Gambar 9. Hasil komputasi skenario 3 - Klaster *Hadoop* Fisik

Grafik pada Gambar 8, memperlihatkan bahwa waktu eksekusi aplikasi *Skyline MR-BNL* pada file berukuran 1,06 *GB* seiring penambahan besar *block size* mengalami penurunan pada ketiga *dataset*, termasuk data *anti-correlated* yang menurun sebesar 20,6% dari 160 detik pada *block* 64 MB menjadi 127 detik pada *block* 512 MB. Penurunan waktu eksekusi juga dialami pada data *correlated* sebesar 12,7% dari 133 detik pada 64 MB menjadi 116 detik pada 512 MB. Penurunan waktu eksekusi *Hadoop MapReduce* pada ketiga *dataset* yang diujikan menunjukkan bahwa besar *block size* mempengaruhi proses komputasi dan semakin besar ukuran *block HDFS,* maka semakin rendah waktu eksekusi *Hadoop MapReduce* yang diperlukan.

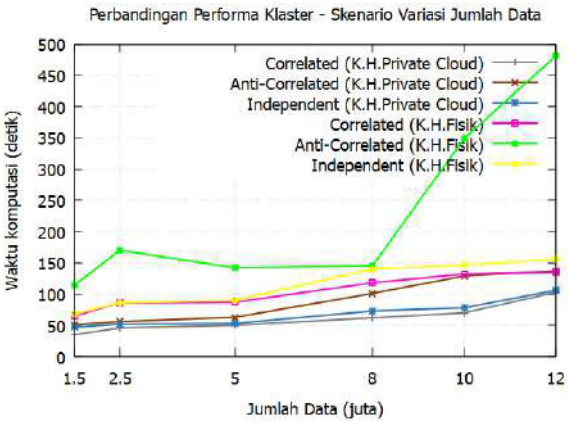
C. Perbandingan Performa Klaster Hadoop Private Cloud dan Klaster Hadoop Fisik

Secara garis besar, performa dari kedua klaster tergantung pada spesifikasi mesin, ukuran data komputasi, jenis dataset, jumlah mesin yang terlibat serta besar ukuran *block HDFS*. Untuk mengukur perbedaan yang signifikan antara kinerja klaster *Hadoop MapReduce* dengan mesin fisik (tanpa virtualisasi) dan tervirtualisasi *private cloud*, digunakan uji statistik t atau t-test. Jenis t-test yang digunakan ialah *paired sample t-test*. Uji statistik akan menunjukkan apakah rata-rata waktu komputasi *Hadoop MapReduce* akan mengalami perubahan yang bermakna ketika klaster divirtualisasikan dengan private cloud. Pada uji statistik ini, ditetapkan signifikansi (α) sebesar 5%. Kemudian, dalam mempermudah perhitungan *t-test* digunakan program Microsoft Excel.

Dalam skenario variasi jumlah data 1,5 juta pada ketiga data sintetis, klaster *private cloud* memproses data *anti-correlated* (55%), *independent* (31%) dan *correlated* (46%) lebih cepat dibandingkan klaster *Hadoop* fisik. Selanjutnya, kinerja klaster meningkat signifikan saat jumlah data pada semua jenis dataset ditambah menjadi 2,5 juta, 5 juta, 8 juta, 10 juta dan 12 juta. Dalam hal ini, klaster *Hadoop private cloud* masih jauh lebih cepat daripada klaster *Hadoop* fisik. Secara keseluruhan, klaster fisik mengeksekusi jenis data *anti-correlated*, *independent* dan *correlated* dengan rentang waktu masing-masing 114-481 detik, 68-156 detik dan 65-135 detik. Sementara klaster *Hadoop private cloud* hanya membutuhkan masing-masing 51-137 detik, 47-106 detik dan 35-103 detik pada jenis data yang sama untuk menyelesaikan aplikasi *Skyline* yang dijalankan.

Tabel 9. Perbandingan waktu komputasi pada skenario 1

Jumlah Data (Juta)	Klaster *Hadoop Private Cloud*			Klaster *Hadoop* Fisik		
	Correlated	*Anticorrelated*	*Correlated*	*Anticorrelated*	*Correlated*	*Anticorrelated*
1.5	35	51	47	65	114	68
2.5	46	56	52	86	170	86
5	50	63	53	87	142	90
8	62	101	73	118	145	140
10	70	129	78	132	349	146
12	103	137	106	135	481	156



Gambar 10. Perbandingan waktu komputasi pada skenario 1

Pada skenario perubahan ukuran atau jumlah data menggunakan 4 mesin, sebelum melakukan t-test, terlebih dahulu dirumuskan hipotesis nol (H_0) dan hipotesis alternatif (H_1). Hipotesis tersebut adalah:
H_0 = Rata-rata waktu komputasi Hadoop MapReduce dalam merespon perubahan jumlah data sebelum divirtualisasi (menggunakan mesin fisik) = Rata-rata waktu komputasi *Hadoop MapReduce* dalam merespon perubahan jumlah data setelah divirtualisasi dengan *private cloud*.
H_1 = Rata-rata waktu komputasi Hadoop MapReduce dalam merespon perubahan jumlah data sebelum divirtualisasi (menggunakan mesin fisik) ≠ Rata-rata waktu komputasi *Hadoop MapReduce* dalam merespon perubahan jumlah data setelah divirtualisasi dengan *private cloud*.

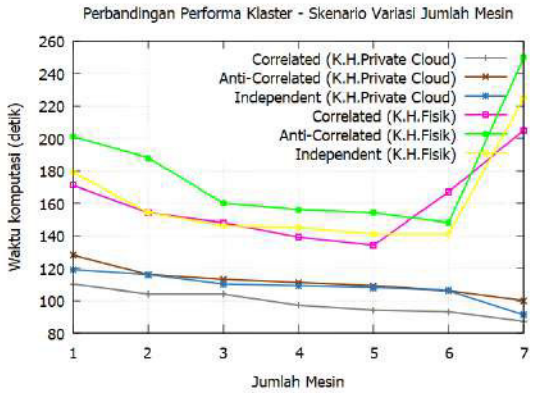
Gambar 11. Hasil Uji-t pada skenario 1

Berdasarkan hasil uji statistik pada Gambar 11, dihasilkan t hitung atau t stat (2,93) > t tabel atau *t critical two tail* (2,77) pada data *anti-correlated*, t hitung atau t stat (7,13) > t tabel atau *t critical two tail* (2,77) pada *data correlated* dan t hitung atau t stat (7.86) > t tabel atau *t critical two tail* (2,77) pada data *independent*. Hal ini berarti H_0 ditolak dan H_1 diterima. Sehingga dapat ditarik kesimpulan bahwa rata-rata waktu komputasi *Hadoop MapReduce* dalam merespon perubahan jumlah data sebelum divirtualisasi (menggunakan mesin fisik) tidak sama dengan rata-rata waktu komputasi *Hadoop MapReduce* dalam merespon perubahan jumlah data setelah divirtualisasi dengan *private cloud* atau

dengan kata lain, pada skenario perubahan jumlah data, *Hadoop MapReduce* menunjukkan performa yang jauh lebih baik saat divirtualisasi dengan *private cloud*, dibandingkan menggunakan klaster fisik.

Tabel 10. Perbandingan waktu komputasi pada skenario 2

Jumlah Mesin	Klaster *Hadoop Private Cloud*			Klaster *Hadoop* Fisik		
	Correlated	*Anticorrelated*	*Correlated*	*Anticorrelated*	*Correlated*	*Anticorrelated*
1	110	128	119	171	201	179
2	104	116	116	154	188	154
3	104	113	110	148	160	146
4	97	111	109	139	156	145
5	94	109	108	134	154	141
6	93	106	106	167	148	141
7	87	100	91	205	250	225



Gambar 12. Perbandingan performa klaster – skenario 2

Pada skenario kedua yakni variasi jumlah mesin, klaster *Hadoop* tervirtualisasi *private cloud* memiliki kinerja yang lebih baik bila dibandingkan dengan klaster fisik. Saat mengeksekusi aplikasi *Skyline MR-BNL* pada data *anti-correlated*, *correlated,* dan *independent* menggunakan 1 mesin, waktu komputasi dengan klaster *Hadoop private cloud* unggul dibanding klaster *Hadoop* fisik (Gambar 12) masing-masing sebesar 36%, 35%, dan 36%. Selanjutnya, seiring jumlah mesin yang diskalakan dari 2 mesin hingga 7 mesin, performa komputasi secara konstan meningkat, sehingga mencapai kondisi yang ideal. Sementara pada klaster *Hadoop* fisik, penambahan mesin memberikan respon yang berbeda pada ketiga dataset. Dataset *anti-correlated* dan *independent* mencapai kondisi ideal ketika mesin berjumlah 6, sehingga ketika mesin ditambah menjadi 7 mesin, terjadi *OC* (*Overhead Communication*). *OC* terjadi ketika klaster memiliki kompleksitas pada proses distribusi data, sinkronisasi antar *node* dan komunikasi antar *daemon Hadoop*. Sedikit berbeda dari kedua dataset lainnya, dataset *correlated* mencapai kondisi ideal ketika mesin berjumlah 5. Kondisi *OC* juga didapati ketika mesin diskalakan menjadi 6 mesin dan 7 mesin.

Anti-Correlated - Dengan Klaster Mesin Fisik		Anti-Correlated - Dengan Klaster Virtualisasi Private Cloud	
Mean	176		109.1666667
Variance	1508.8		31.76666667
Observations	6		6
Pearson Correlation	-0.549952051		
Hypothesized Mean Difference	0		
df	5		
t Stat	3.878752663		
P(T<=t) one-tail	0.0058283		
t Critical one-tail	2.015048373		
P(T<=t) two-tail	0.0116566		
t Critical two-tail	2.570581836		
Independent - Dengan Klaster Mesin Fisik		**Independent - Dengan Klaster Virtualisasi Private Cloud**	
Mean	158.6666667		106.6666667
Variance	1078.666667		70.26666667
Observations	6		6
Pearson Correlation	-0.848989556		
Hypothesized Mean Difference	0		
df	5		
t Stat	3.168141336		
P(T<=t) one-tail	0.012433372		
t Critical one-tail	2.015048373		
P(T<=t) two-tail	0.024866745		
t Critical two-tail	2.570581836		
Correlated - Dengan Klaster Mesin Fisik		**Correlated - Dengan Klaster Virtualisasi Private Cloud**	
Mean	157.8333333		96.5
Variance	668.5666667		44.3
Observations	6		6
Pearson Correlation	-0.618835736		
Hypothesized Mean Difference	0		
df	5		
t Stat	4.937396094		
P(T<=t) one-tail	0.002165907		
t Critical one-tail	2.015048373		
P(T<=t) two-tail	0.004331815		
t Critical two-tail	2.570581836		

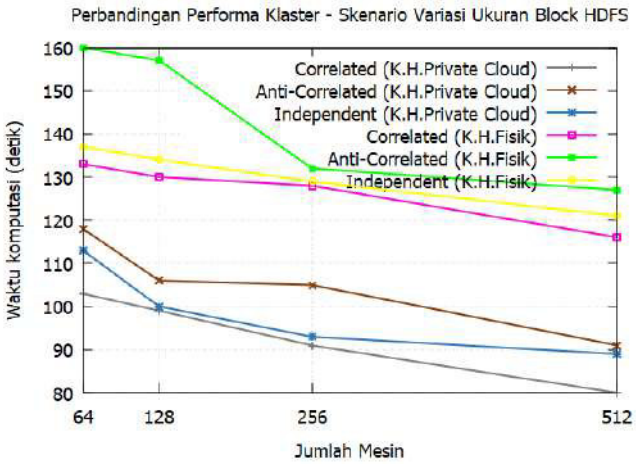
Gambar 13. Hasil Uji-t pada skenario 2

Pada Gambar 13, dikarenakan t hitung (9,11) > t tabel (2,77) pada data *anti-correlated*, t hitung (8,02) > t tabel (2,77) pada *data correlated* dan t hitung (43,8) > t tabel (2,77) pada data *independent*, maka H_1 diterima. Sehingga disimpulkan bahwa ketika menjalankan skenario 1, terdapat perbedaan antara rata-rata waktu komputasi sebelum klaster di virtualisasikan dan sesudah divirtualisasikan dengan Private Cloud. Hal ini juga menunjukkan ketika menskalakan jumlah mesin, klaster *Hadoop* yang divirtualisasikan dengan *private cloud* bekerja lebih baik menggunakan komputer fisik (tidak divirtualisasikan).

Pada skenario penambahan ukuran *block size HDFS* (Gambar 14), kedua klaster menunjukkan *trend* kinerja yang sama. Kedua klaster sama-sama menunjukkan peningkatan waktu komputasi *Hadoop MapReduce* ketika jumlah *block* yang akan dieksekusi berkurang. Dalam hal ini, secara keseluruhan, klaster *private cloud* menyelesaikan komputasi lebih cepat dibandingkan klaster *Hadoop* fisik. Misalnya, menggunakan *block size* 64 MB pada ketiga dataset berukuran 1,06 GB akan menghasilkan potongan *block* sebanyak 17 *block*. Waktu komputasi yang diperlukan oleh klaster *private cloud* pada data *anti-correlated*, *independent* dan *correlated* lebih rendah dibanding klaster *Hadoop* fisik dengan persentase masing-masing sebesar 26%, 18%, dan 23%. Kemudian, saat *block size* diperbesar menjadi 128 MB, 256 MB dan 512 MB sehingga potongan *block* menjadi lebih sedikit, kinerja klaster menurun secara signifikan pada semua jenis dataset.

Tabel 11. Perbandingan waktu komputasi pada skenario 3

Block Size	Klaster *Hadoop Private Cloud*			Klaster *Hadoop* Fisik		
	Correlated	*Anticorrelated*	*Independent*	*Correlated*	*Anticorrelated*	*Independent*
64 *MB*	103	118	113	133	160	137
128 *MB*	99	106	100	130	157	134
256 *MB*	91	105	93	128	132	129
512 *MB*	80	91	89	116	127	121



Gambar 14. Perbandingan performa klaster – skenario 3

Anti-Correlated - Dengan Klaster Mesin Fisik		Anti-Correlated - Dengan Klaster Virtualisasi Private Cloud
Mean	138.6666667	100.6666667
Variance	258.3333333	70.33333333
Observations	3	3
Pearson Correlation	0.67386737	
Hypothesized Mean Difference	0	
df	2	
t Stat	5.428571429	
P(T<=t) one-tail	0.016149255	
t Critical one-tail	2.91998558	
P(T<=t) two-tail	0.03229851	
t Critical two-tail	4.30265273	
Independent - Dengan Klaster Mesin Fisik		**Independent - Dengan Klaster Virtualisasi Private Cloud**
Mean	128	94
Variance	43	31
Observations	3	3
Pearson Correlation	0.958634312	
Hypothesized Mean Difference	0	
df	2	
t Stat	29.44486373	
P(T<=t) one-tail	0.000575705	
t Critical one-tail	2.91998558	
P(T<=t) two-tail	0.001151411	
t Critical two-tail	4.30265273	
Correlated - Dengan Klaster Mesin Fisik		**Correlated - Dengan Klaster Virtualisasi Private Cloud**
Mean	124.6666667	90
Variance	57.33333333	91
Observations	3	3
Pearson Correlation	0.955267051	
Hypothesized Mean Difference	0	
df	2	
t Stat	18.67895141	
P(T<=t) one-tail	0.00142693	
t Critical one-tail	2.91998558	
P(T<=t) two-tail	0.002853861	
t Critical two-tail	4.30265273	

Gambar 15. Hasil Uji-t pada skenario 3

Pada skenario 3, dirumuskan hipotesis uji t yakni H_0 = Rata-rata waktu komputasi *Hadoop MapReduce* dalam merespon perubahan *block size* HDFS sebelum divirtualisasi (menggunakan mesin fisik) = Rata-rata waktu komputasi *Hadoop MapReduce* dalam merespon perubahan *block size* HDFS setelah divirtualisasi dengan *private cloud*. Sementara H_1 = Rata-rata waktu komputasi *Hadoop MapReduce* dalam merespon perubahan *block size* HDFS sebelum divirtualisasi (menggunakan mesin fisik) ≠ Rata-rata waktu komputasi *Hadoop MapReduce* dalam merespon perubahan *block size* HDFS setelah divirtualisasi dengan *private cloud*. Sehingga berdasarkan Gambar 15 yang mana t hitung (5,42) > t tabel (4,30) pada data anti-correlated, t hitung (18,67) > t tabel (4,30) pada data *correlated* dan t hitung (29,44) > t tabel (4,30) pada data independent, klaster *Hadoop* yang divirtualisasikan dengan *private cloud* kembali unggul dibandingkan klaster *Hadoop* fisik pada ketiga dataset ketika memproses beragam ukuran *block* HDFS untuk data berukuran 1,06 GB.

Berdasarkan hasil uji statistik t pada Gambar 11, Gambar 13, Gambar 15, dapat dibuktikan bahwa dalam menjalankan keseluruhan pengujian mulai dari perubahan ukuran file, perubahan jumlah mesin dan modifikasi *block size* HDFS, dengan spesifikasi tertentu, klaster *Hadoop private cloud* yang dibangun bekerja lebih baik dalam menjalankan komputasi *Hadoop MapReduce*, dibandingkan klaster mesin fisik (tanpa virtualisasi).

4. Kesimpulan

- Untuk mengimplementasikan klaster *Hadoop MapReduce* di atas *private cloud computing*, dilakukan proses instalasi dan konfigurasi lingkungan tempat *daemon Hadoop* dijalankan serta parameter konfigurasi untuk *daemon Hadoop*. *Daemon Hadoop* yang dimaksud adalah *name node*, *data node*, *secondary name node*, *resource manager*, dan *node manager*.
- Penambahan volume data yang dieksekusi dari 1,5 juta hingga 12 juta akan menyebabkan kenaikan waktu komputasi dan penurunan kinerja klaster.
- Penambahan jumlah mesin dari 1 mesin menjadi 7 mesin meningkatkan kinerja klaster *Hadoop private cloud*, sementara bagi klaster *Hadoop* fisik menyebabkan overhead.
- *Block Size* menentukan jumlah potongan *block* yang akan dieksekusi dan mempengaruhi kecepatan komputasi *Hadoop MapReduce*.
- Pada seluruh skenario pengujian performa yang peneliti lakukan, klaster *Hadoop* yang divirtualisasikan di atas *Private Cloud* dengan spesifikasi mesin Intel(R) Xeon (R) E3-1225 v5 @ 3,30 GHz RAM 16 GB, bekerja jauh lebih baik dalam mengeksekusi aplikasi *Skyline* dibandingkan klaster *Hadoop* yang dibangun pada mesin fisik dengan spesifikasi mesin Intel Core i5 CPU @ 3,00GHz RAM 4 GB. Hal ini dibuktikan dari hasil perbandingan rata-rata waktu komputasi kedua klaster dengan *pairet t-test* yang mana pada skenario perubahan jumlah data dihasilkan bahwa t hitung (2,93) > t tabel (2,77) pada data *anti-correlated*, t hitung (7,13) > t tabel (2,77) pada data *correlated* dan t hitung (7,86) > t tabel (2,77) pada data *independent*. Pada skenario perubahan jumlah mesin dihasilkan t hitung (9,11) > t tabel (2,77) pada data *anti-correlated*, t hitung (8,02) > t tabel (2,77) pada data *correlated* dan t hitung (43,8) > t tabel (2,77) pada data *independent*. Sementara skenario modifikasi ukuran *block size* HDFS dihasilkan t hitung (5,42) > t tabel (4,30) pada data *anti-correlated*, t hitung (18,67) > t tabel (4,30) pada data *correlated* dan t hitung (29,44) > t tabel (4,30) pada data *independent*.

5. Daftar Pustaka

Achahbar, O., & Abid, M. R. (2014). The impact of virtualization on high performance computing clustering in the cloud. *International Journal of Distributed Systems and Technologies*, *6*(4), 65–81. https://doi.org/10.4018/IJDST.2015100104

Azizah, N., & Saptono, H. (2020). Uji Performa Dan Perbandingan Rdbms Mysql Dan Hive-Hadoop. *Jurnal Informatika Terpadu*, *6*(1), 20–28. https://journal.nurulfikri.ac.id/index.php/JIT

Basuki, K., Palit, H. N., & Dewi, L. P. (2015). Implementasi Hadoop: Studi Kasus Pengolahan Data Peminjaman Perpustakaan Universitas Kristen Petra. *Jurnal Infra*, *3*, 7. http://publication.petra.ac.id/index.php/teknik-informatika/article/view/3135

Börzsönyi, S., Kossmann, D., & Stocker, K. (2001). The skyline operator. *Proceedings - International Conference on Data Engineering*, 421–430. https://doi.org/10.1109/icde.2001.914855

Ivanov, T., Zicari, R. V., Izberovic, S., & Tolle, K. (2014). *Performance Evaluation of Virtualized Hadoop Clusters*. http://arxiv.org/abs/1411.3811

Made, I., Putra, S. W., Wijayanto, H., & Zafrullah, A. (2023). *Komputasi Paralel untuk Perhitungan Relasi Dominasi Menggunakan Skyline Query pada Lokasi Wisata di Pulau Lombok* [Universitas Mataram]. http://eprints.unram.ac.id/39679/

Nuriadin, A., Dyan Nofia Harumike, Y., Tana Sanggamu, D., Studi Ilmu Komunikasi, P., & Islam Blitar, U. (2021). Sejarah Perkembangan Dan Implikasi Internet Pada Media Massa Dan Kehidupan Masyarakat. *SELASAR KPI: Referensi Media Komunikasi Dan Dakwah*, *1*(1). https://ejournal.iainu-kebumen.ac.id/index.php/selasar/index

Prabowo, W. S., Muslim, M. H., & Iryanto, S. B. (2015). Pusat Data Privat Virtual Pemerintah Berbasis Komputasi Awan (Studi Empiris Pada Lembaga Ilmu Pengetahuan Indonesia). *Jurnal Penelitian dan Pengembangan Komunikasi, dan Informatika*, *6*(2), 1–12.

Ryanto, A. M. (2017). *Analisis Kinerja Framework Big Data Pada Klaster Tervirtualisasi: Hadoop Mapreduce Dan Apache Spark* [Universitas Hasanuddin]. http://digilib.unhas.ac.id/

Subagya, N., Wijajarto, A., & Almaarif, A. (2021). Implementasi Dan Analisis Hadoop Element Availability Berdasarkan Daemon Log Monitoring Hadoop Element Availability Implementation and Analysis Based on Daemon Log Monitoring. *E-Proceeding of Engineering*, *8*(5), 9223–9234. https://openlibrary.telkomuniversity.ac.id/pustaka/170581/implementasi-dan-analisis-hadoop-element-availability-berdasarkan-daemon-log-monitoring-menggunakan-log4j-logging.html

Subramanian, S. K., & Gouda, K. C. (2015). A Study on The Different Aspects of Virtual Private Cloud. *International Journal of Applied Engineering Research*, *10*(86), 343–347. https://www.ripublication.com/Volume/ijaerv10n86.htm

Wibawa, I. P. A. P., Giriantari, I. D., & Sudarma, M. (2018). Komputasi Paralel Menggunakan Model Message Passing Pada SIM RS (Sistem Informasi Manajemen Rumah Sakit). *Majalah Ilmiah Teknologi Elektro*, *17*(3), 439. https://doi.org/10.24843/mite.2018.v17i03.p20