

Perbandingan Kinerja Hbase dan MongoDB Sebagai *Backend IoT Data Storage*

Niki Yuniar Wicaksono¹, Eko Sakti Pramukantoro², Widhi Yahya³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹ nikiyw05@gmail.com, ²ekosakti@ub.ac.id, ³ widhi.yahya@ub.ac.id

Abstrak

IoT (*Internet Of Things*) saat ini sedang berkembang dibuktikan dengan banyaknya penelitian yang membahas IoT. IoT terdiri dari dua komponen utama yaitu Internet dan *Things* yang saling terhubung, untuk mengumpulkan data dan informasi kedalam media penyimpanan data yang bisa diolah sesuai kepentingan. Namun terdapat tantangan dalam membangun media penyimpanan data IoT, yaitu volume data yang besar, bentuk dan format data yang beragam. Pada penelitian sebelumnya telah dikembangkan sebuah *Framework* media penyimpanan data IoT dari node sensor. *Framework* tersebut menggunakan *NoSQL MongoDB* sebagai media penyimpanan datanya. Pada penelitian ini, diusulkan sebuah *IoT data storage NoSQL Hbase* dengan menggunakan *Framework MongoDB* karena *less scema* dan *writing speed* yang bagus dalam pengujian *performa random writing test*. Penelitian ini menguji Hbase dan MongoDB dengan pengujian penyimpanan data dan pengujian kinerja. Untuk pengujian kinerja MongoDB unggul atas uji *text* pada parameter runtime sebesar 33s, throughput 1559 ops/s, CPU Usage 48%, Memory Usage 67%, dan Disk I/O 4354kb. Saat uji gambar hasil perbedaan berbeda sedikit keunggulan MongoDB pada parameter Runtime 64s, Throughput 90 ops/s, CPU Usage 46%, Memory Usage 84%, sedangkan saat uji gambar Hbase unggul pada parameter *Disk I/O* sebesar 109.462kb.

Kata kunci: IoT (*Internet Of Things*), Hbase, MongoDB, Data Storage

Abstract

IoT (*Internet Of Things*) is currently evolving as evidenced with many studies that discuss the IoT. IoT consists of two main components of the Internet and interconnected Things, to collect data and information into data storage media that can be processed as needed. However, there are challenges in building IoT data storage media, like a large data volumes, diverse forms and data formats. In previous research has developed a *Framework of IoT data storage media from the sensor node*. The *Framework* uses *NoSQL MongoDB* as its data storage. In this study, proposed an *IoT NoSQL Hbase data storage using the MongoDB Framework* because it free scema and good writing speed in the performance testing of random writing tests. This study, tested Hbase and MongoDB by testing ability to storage data in data storage. The result of this test is Hbase and MongoDB is able to receive and insert text data and image data. While for testing the *Framework of data storage media with Hbase using storage testing* For testing the performance of MongoDB, better result test on runtime parameters of 33s, throughput of 1559 ops/s, 48% Usage CPU, 67% Memory Usage, and 4354kb Disk I/O. When testing images the results of differences in slightly different better of MongoDB on Runtime parameters 64s, throughput 90 ops/s, CPU Usage 46%, Memory Usage 84%, while when testing Hbase images superior in Disk I/O parameters of 109.462kb.

Keywords: IoT (*Internet Of Things*), Hbase, MongoDB, Data Storage

1. PENDAHULUAN

IoT (*internet Of Things*) saat ini sedang berkembang dibuktikan dengan banyaknya penelitian yang membahas IoT. IoT pada dasarnya terdiri dari dua komponen utama yaitu internet dan *things* yang saling terhubung. IoT berfungsi mengumpulkan data dan informasi dari

lingkungan fisik(*environment*), yang selanjutnya data tersebut akan diproses sesuai dengan kepentingan dan kebutuhan di setiap bidangnya. Konsep IoT mengacu pada tiga elemen utama yaitu barang fisik yang dilengkapi modul IoT, perangkat koneksi internet seperti *router* dan modem, dan *data center* berbasis *cloud* tempat untuk menyimpan aplikasi beserta *data storage*

(Pramukantoro. Dkk.,2017.)

Terdapat tantangan dalam membangun media penyimpanan data IoT, yaitu volume data yang besar, bentuk dan format data yang beragam. Pada penelitian sebelumnya (Pramukantoro.Dkk., 2017) telah dikembangkan sebuah framework media penyimpanan data IoT dari node sensor. Framework tersebut menggunakan *NoSQL MongoDB* sebagai media penyimpanan datanya. Skenario pengujian yang digunakan untuk menguji framework tersebut meliputi *functional testing*, *scalability testing* dan *respon time testing* dengan IoT Apps. Hasil yang didapat dari *functional testing* adalah framework yang dikembangkan dapat mengirim dan menyimpan data bervolume besar dan beragam. Sedangkan dari *non-functional testing*, framework yang dikembangkan dapat menerima 443 data /detik dari IGD(*Internet Gateway Device*), dapat mengirimkan 173 data/detik ke IoT Apps dan respon time yang didapat dibawah 1 detik.

Saat ini telah terdapat 225 jenis *NoSQL Database* yang telah dikembangkan. Setiap tipe *Database* pada *NoSQL* memiliki mekanisme implementasi, karakteristik penyimpanan, konfigurasi dan optimasi yang berbeda-beda. Ini yang menyebabkan semakin terbukanya tantangan dalam pemilihan *NoSQL* yang akan digunakan (EnqiqTang & Yushun Fan 2016). Oleh sebab itu, *NoSQL MongoDB* yang digunakan pada penelitian sebelumnya (Pramukantoro. Dkk.,2017) perlu dibandingkan dengan *NoSQL Database* yang lain.

Diantara 225 jenis *NoSQL Database*, Hbase juga cukup populer digunakan untuk menangani data dalam jumlah besar yang setiap harinya dihasilkan dari bermacam – macam sumber. Hbase didesain untuk dapat mengelola data berukuran besar dalam satu sistem terdistribusi karena berjalan diatas Hadoop, lebih tepatnya *Hadoop Distributed File System(HDFS)* dan memiliki fungsi *sharding original* bawaan yang dapat bekerja secara otomatis maupun manual. Hbase memiliki karakteristik '*fault tolerance*' atau mampu menjamin keutuhan data meskipun terjadi kegagalan pada beberapa komputer yang dikerjakannya (Wijaya, Wayan, M, 2015).

Dengan demikian dilakukan perbandingan kinerja yang akan buat pada penelitian saat ini yaitu Hbase dengan MongoDB sebagai media penyimpanan data IoT dan menggunakan lingkungan (*environment / framework*) yang telah dikembangkan oleh peneliti sebelumnya. Hasil pengujian nantinya akan didapat sebagai

penilaian kinerja kedua database. Pengujian *framework media* penyimpanan data dengan Hbase menggunakan pengujian penyimpanan *database*. Dari penelitian ini diharapkan memberikan pertimbangan pengguna *database sql* maupun *nosql* dalam membangun sistem pada penyimpanan data IoT.

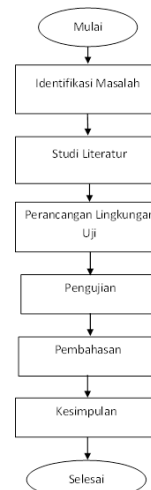
2. LANDASAN KEPUSTAKAAN

Pada penelitan sebelumnya yang terkait dengan *NoSQL database* adalah penelitian Naheman W, Wei J(2013) yang berjudul "*Review of NoSQL database and performance testing on Hbase*". Dalam penelitian tesebut menguji kinerja dari *NoSQL Hbase* dalam lingkungan server yang terdistribusi. Pengujian database tersebut yaitu menguji *column family test*, *sorting test*, *random read and write test*, dan *query test*. Pengujian tersebut masing-masing ditambahnya jumlah beban atau operasi tergantung pada proses pengujiannya.

Pada penelitian terkait kedua yaitu penelitian yang dilakukan oleh EnqingTang & Yushu Fan(2016) dengan judul "*Performance Comparison Between Five NoSQL Databases*". Melakukan perbandingan kinerja lima *NoSQL* dengan meningkatkan jumlah baris data. Pengujian ini memakai parameter *Runtime* dan *Throughput* pada operasi insert, operasi read dan operasi update. Penelitian tersebut menggunakan *tools* YCSB untuk menguji ke-lima *NoSQL database* tersebut sesuai dengan parameternya.

3. METODOLOGI

Dalam bab ini akan dijelaskan tahapan dan metode yang digunakan untuk mengerjakan penelitian perbandingan kinerja *Database NoSQL Hbase* dan *MongoDB* pada IoT data Storege. Diagram alur metedologi penelitian yang ditunjukan pada Gambar 1.



Gambar 1. Gambar Diagram Alir Metodologi Penelitian

3.1 Identifikasi Masalah

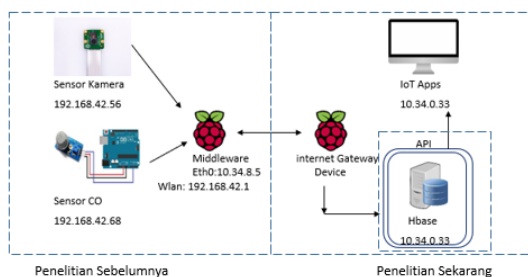
Identifikasi masalah dilakukan untuk menentukan penelitian apa yang dilakukan. Dengan melihat kondisi saat ini serta melihat masalah apa yang akan dan mungkin terjadi di lingkungan. Dengan adanya identifikasi masalah penelitian akan menjadi fokus dan memiliki tujuan serta manfaat yang jelas.

3.2. Studi Literatur

Studi literature bertujuan untuk mencari dasar-dasar teori dan kajian pustaka yang digunakan untuk menunjang penulisan skripsi ini. Teori-teori pendukung tersebut didapat dari jurnal, e-book, artikel, website dan dokumentasi project pada penelitian sebelumnya.

3.3 Perancangan Lingkungan Uji

Pembangunan lingkungan uji merupakan tahap yang digunakan penulis untuk menyiapkan berbagai kebutuhan yang dibagi menjadi kebutuhan perangkat lunak dan kebutuhan perangkat keras. Kebutuhan perangkat keras berisikan daftar perangkat yang akan digunakan dalam penelitian. Kebutuhan perangkat lunak berisikan daftar program yang digunakan dalam penelitian. Sebelum melakukan lingkungan uji, diperlukan instalasi Hbase pada server yang digunakan kemudian dilanjutkan tahap konfigurasi Hbase. Proses konfigurasi bertujuan agar Hbase yang akan digunakan dapat berjalan dengan baik. Adapun topologi lingkungan yang akan dibangun sebagai berikut :



Gambar 2. Topologi Lingkungan System

Gambar 2. Menjelaskan tentang penelitian sebelumnya dan penelitian yang akan dilakukan. Penelitian yang akan dilakukan akan menggunakan lingkungan dari penelitian sebelumnya, dan hanya mengganti tempat penyimpanannya saja menjadi Hbase sesuai dengan garis kotak merah putus – putus.

3.4. Pengujian

Pengujian dilakukan untuk mengetahui apakah Hbase yang diusulkan dapat bekerja sesuai fungsionalitas. Pengujian yang dilakukan yaitu Hbase dapat menerima dan menyimpan data. Sesuai dengan tabel 1.

Tabel 1. Pengujian Penyimpanan

No	Deskripsi Pengujian	Skenario
UF_001	API Webservice dapat mengirim data sensor dht11 dari <i>Internet Gateway Device</i> ke database Hbase	1.Middleware sudah berjalan 2.Pengguna menjalankan kode <i>subscriber</i> 3.Pengguna menjalankan kode service 4.kode menampilkan pesan data sensor dht11 berhasil di POST
UF_002	API Webservice dapat mengirim data sensor kamera dari <i>Internet Gateway Device</i> ke database Hbase	1.Middleware sudah berjalan 2.Pengguna menjalankan kode <i>subscriber</i> 3.Pengguna menjalankan kode service 4.kode menampilkan pesan data sensor kamera berhasil di POST
UF_003	Database Hbase Dapat menyimpan data dari sensor dht11	1.Pengguna melihat data pada tabel penyimpanan data
UF_004	Database Hbase dapat menyimpan data dari sensor kamera	1.Pengguna melihat data pada tabel penyimpanan data

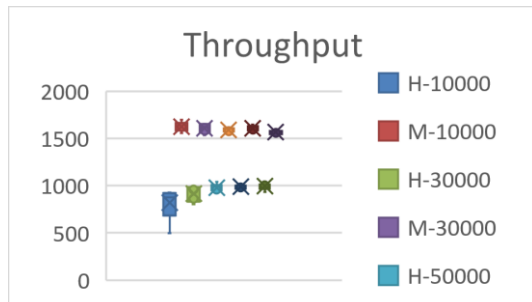
3.5 Pembahasan

Setelah proses pengujian fungsionalitas dan pengujian kinerja *Database*, dilakukan pembahasan pada hasil pengujian tersebut. Pembahasan dilakukan untuk mengetahui fungsi Hbase dapat menjalankan fungsinya.

3.6 Kesimpulan

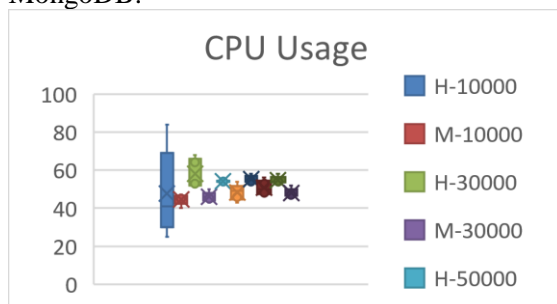
Kesimpulan dan saran merupakan tahapan penutup dari proses penelitian setelah melalui

dengan parameter *Runtime* dengan banyak data 100.000 menghasilkan 100s Hbase, 64s MongoDB.



Gambar 8. *Insert data text dengan parameter Throughput*

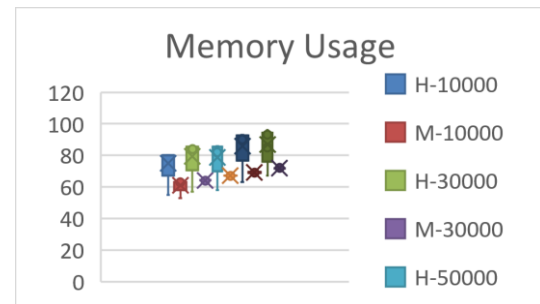
Terlihat pada gambar 8, pengujian data *text* menggunakan parameter *throughput* dengan banyak data 10.000 menghasilkan 818 ops/s Hbase, 1.626 ops/s MongoDB. *Text* dengan parameter *throughput* dengan banyak data 30.000 menghasilkan 917 ops/s Hbase, 1.608 ops/s MongoDB. *Text* dengan parameter *throughput* banyak data 50.000 menghasilkan 977 ops/s Hbase, 1.588 ops/s MongoDB. *Text* dengan parameter *throughput* dengan banyak data 70.000 menghasilkan 987 ops/s Hbase, 1.607 ops/s MongoDB. *Text* dengan parameter *throughput* dengan banyak data 100.000 menghasilkan 992 ops/s Hbase, 1.564 ops/s MongoDB.



Gambar 9. *Insert data text dengan parameter CPU Usage*

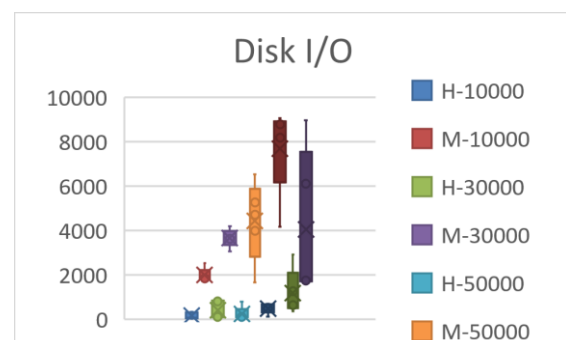
Terlihat pada gambar 9. Pengujian data *text* menggunakan parameter *CPU Usage* dengan banyak data 10.000 menghasilkan 48% Hbase, 44% MongoDB. *Text* dengan parameter *CPU Usage* dengan banyak data 30.000 menghasilkan 58% Hbase, 46% MongoDB. *Text* dengan parameter *CPU Usage* banyak data 50.000 menghasilkan 54% Hbase, 48% MongoDB. *Text* dengan parameter *CPU Usage* dengan banyak data 70.000 menghasilkan 56% Hbase, 51%

MongoDB. *Text* dengan parameter *CPU Usage* dengan banyak data 100.000 menghasilkan 55% Hbase, 48% MongoDB.



Gambar 10. *Insert data text dengan parameter Memory Usage*

Terlihat pada gambar 10, Pengujian data *text* menggunakan parameter *Memory Usage* dengan banyak data 10.000 menghasilkan 75% Hbase, 61% MongoDB. *Text* dengan parameter *Memory Usage* dengan banyak data 30.000 menghasilkan 79% Hbase, 64% MongoDB. *Text* dengan parameter *Memory Usage* banyak data 50.000 menghasilkan 79% Hbase, 67% MongoDB. *Text* dengan parameter *Memory Usage* dengan banyak data 70.000 menghasilkan 83% Hbase, 69% MongoDB. *Text* dengan parameter *Memory Usage* dengan banyak data 100.000 menghasilkan 87% Hbase, 72% MongoDB.



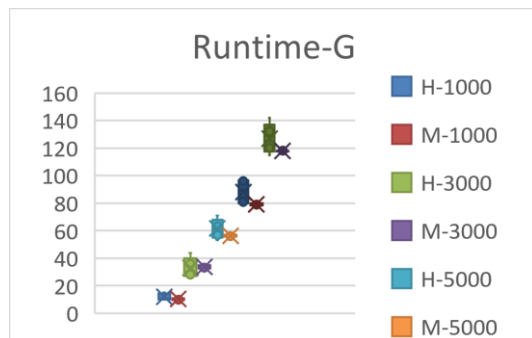
Gambar 11. *Insert data text dengan parameter Disk I/O*

Terlihat pada gambar 11, Pengujian data *text* menggunakan parameter *Disk I/O* dengan banyak data 10.000 menghasilkan 204kb Hbase, 687kb MongoDB. *Text* dengan parameter *Disk I/O* dengan banyak data 30.000 menghasilkan 387kb Hbase, 415kb MongoDB. *Text* dengan parameter *Disk I/O* banyak data 50.000 menghasilkan 231kb Hbase, 1.169kb MongoDB. *Text* dengan parameter *Disk I/O* dengan banyak data 70.000 menghasilkan 477kb Hbase, 1.289kb MongoDB. *Text* dengan

parameter *Disk I/O* dengan banyak data 100.000 menghasilkan 1164kb Hbase, 927kb MongoDB.

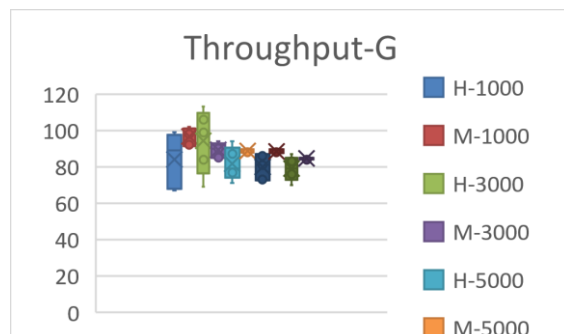
4.3.2 Pengujian Hbase dan MongoDB Insert Data Gambar

Pengujian insert data gambar pada Hbase dan MongoDB terdapat pada grafik chart dibawah dengan masing-masing parameternya :



Gambar 12. Insert data gambar dengan parameter *Runtime*

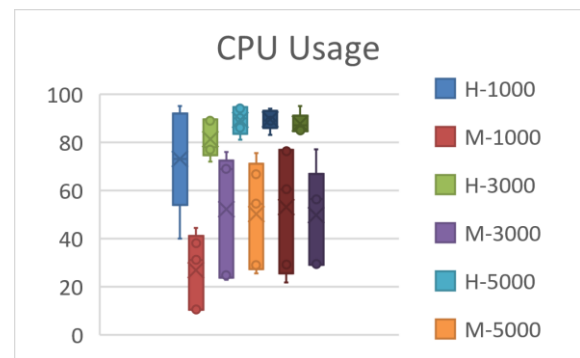
Terlihat pada gambar 12, pengujian data gambar menggunakan parameter *Runtime* dengan banyak data 1.000 menghasilkan 12s Hbase, 10s MongoDB. Gambar dengan parameter *Runtime* dengan banyak data 3.000 menghasilkan 33s Hbase, 34s MongoDB. Gambar dengan parameter *Runtime* banyak data 5.000 menghasilkan 62s Hbase, 56s MongoDB. gambar dengan parameter *Runtime* dengan banyak data 7.000 menghasilkan 88s Hbase, 79s MongoDB. gambar dengan parameter *Runtime* dengan banyak data 10.000 menghasilkan 127s Hbase, 118s MongoDB.



Gambar 13. Insert data gambar dengan parameter *Throughput*

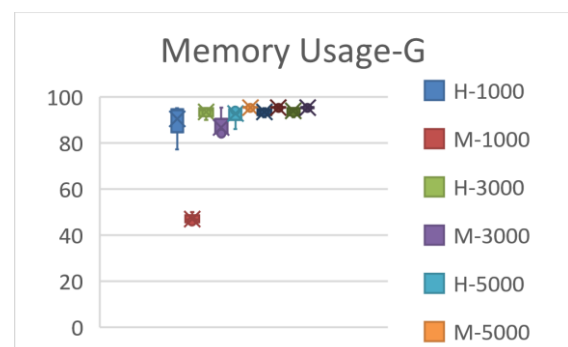
Terlihat pada gambar 13, pengujian data gambar menggunakan parameter *throughput* dengan banyak data 1.000 menghasilkan 84

ops/s Hbase, 97 ops/s MongoDB. Gambar dengan parameter *throughput* dengan banyak data 3.000 menghasilkan 94 ops/s Hbase, 89 ops/s MongoDB. Gambar dengan parameter *throughput* banyak data 5.000 menghasilkan 82 ops/s Hbase, 89 ops/s MongoDB. Gambar dengan parameter *throughput* dengan banyak data 7.000 menghasilkan 80 ops/s Hbase, 88 ops/s MongoDB. Gambar dengan parameter *throughput* dengan banyak data 10.000 menghasilkan 79 ops/s Hbase, 85 ops/s MongoDB.



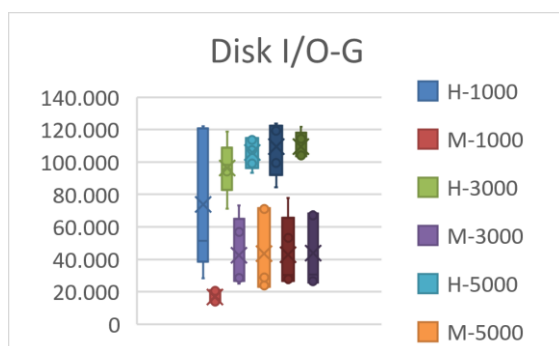
Gambar 14. Insert data gambar dengan parameter *CPU Usage*

Terlihat pada gambar 14, Pengujian data gambar menggunakan parameter *CPU Usage* dengan banyak data 1.000 menghasilkan 71% Hbase, 27% MongoDB. Gambar dengan parameter *CPU Usage* dengan banyak data 3.000 menghasilkan 80% Hbase, 52% MongoDB. Gambar dengan parameter *CPU Usage* banyak data 5.000 menghasilkan 86%, 50% MongoDB. Gambar dengan parameter *CPU Usage* dengan banyak data 7.000 menghasilkan 88% Hbase, 53% MongoDB. Gambar dengan parameter *CPU Usage* dengan banyak data 10.000 menghasilkan 87% Hbase, 50% MongoDB.



Gambar 15. Insert data gambar dengan parameter *Memory Usage*

Terlihat pada gambar 15, Pengujian data gambar menggunakan parameter *Memory Usage* dengan banyak data 1.000 menghasilkan 91% Hbase, 47% MongoDB. Gambar dengan parameter *Memory Usage* dengan banyak data 3.000 menghasilkan 94% Hbase, 87% MongoDB. Gambar dengan parameter *Memory Usage* banyak data 5.000 menghasilkan 93%, 95% MongoDB. Gambar dengan parameter *Memory Usage* dengan banyak data 7.000 menghasilkan 94% Hbase, 95% MongoDB. Gambar dengan parameter *Memory Usage* dengan banyak data 10.000 menghasilkan 94% Hbase, 95% MongoDB.



Gambar 16. Insert data gambar dengan parameter Disk I/O

Terlihat pada gambar 16, Pengujian data gambar menggunakan parameter *Disk I/O* dengan banyak data 1.000 menghasilkan 73.891kb. Hbase, 16.779kb MongoDB. Gambar dengan parameter *Disk I/O* dengan banyak data 3.000 menghasilkan 96.287 kb Hbase, 87% MongoDB. Gambar dengan parameter *Disk I/O* banyak data 5.000 menghasilkan 105.815 kb, 43.617kb MongoDB. Gambar dengan parameter *Disk I/O* dengan banyak data 7.000 menghasilkan 109.560kb Hbase, 42.901kb MongoDB. Gambar dengan parameter *Disk I/O* dengan banyak data 10.000 menghasilkan 109.461kb Hbase, 43.761kb MongoDB.

4.3.3 Analisis Pengujian

MongoDB dan Hbase memiliki kekurangan dan kelebihan masing – masing, jika dilihat dari kinerja pada hasil pengujian MongoDB tentu lebih baik. Akan tetapi Hbase pun masih layak digunakan untuk menyimpan data sensor pada *environment* yang sudah ada.

5. KESIMPULAN

Kesimpulan ditarik dari pengujian diatas yang mana menghasilkan hal-hal berikut :

1. Hbase dibangun menggunakan VPS yang berada di filkom ub. VPS tersebut memiliki spesifikasi yang sama dengan penelitian sebelumnya. Hbase dapat menyimpan dan menampilkan data yang heterogen dari node sensor dalam lingkungan (*environment*) sebelumnya yaitu dengan menggunakan lingkungan MongoDB.

2. Dari hasil pengujian Hbase dan MongoDB menggunakan dua pengujian text dan gambar dengan parameter yaitu *runtime*, *throughput*, *cpu usage*, *memory usage*, dan *disk i/o* dan lima data uji menghasilkan perbandingan data yaitu rata-rata data Text dengan parameter *runtime* sebesar 54s untuk Hbase, 33s untuk MongoDB. Parameter *throughput* sebesar 938 ops/s untuk Hbase, 1599 ops/s untuk MongoDB. Parameter CPU usage sebesar 54% untuk Hbase, 48% untuk MongoDB. Parameter Memory Usage sebesar 81% untuk Hbase, 67% untuk MongoDB. Dan Parameter Disk i/o sebesar 482kb untuk Hbase, 4354kb untuk MongoDB. Sedangkan rata-rata data gambar parameter *runtime* sebesar 64s Hbase, 60s MongoDB, parameter *Throughput* 84 ops/s Hbase, 90 ops/s MongoDB. Parameter CPU usage 83% Hbase, 46% mongo. Parameter Memory usage 93% Hbase, 84% MongoDB. Dan parameter Disk I/O 99.003kb Hbase, 37.885kb MongoDB. Dari perbandingan rata-rata nilai pengujian MongoDB lebih banyak unggulnya dibandingkan dengan Hbase pada penyimpanan data text dan data gambar. Akan tetapi jika dilihat dari menyimpan data gambar dengan parameter *Disk I/O* Hbase mempunyai kinerja baik dibanding MongoDB.

6. DAFTAR PUSTAKA

- Pramukantoro, Eko. Dkk. (2017). Topic Based IoT Data Storage Framework For Heterogeneous Sensor Data.
- EnqingTang, Fan Yushun, (2016). Performance Comparison between Five NoSQL Databases. IEEE.
- Naheman, W., Wei, J (2013). Review of NoSQL Databases and Performance Testing on Hbase. IEEE.
- Wijaya, W. M. (2015) Teknologi Big Data, Online. Available at : <https://www.kompasiana.com/wmwijaya/52bad1c6ea834f9598b4580/Hbase-hyper->

nosql-database/ (Accessed : 15 January 2018).