ISSN:2302-8149
e-ISSN:2540-9719

Optimasi Desain Infrastruktur Big Data Menggunakan Teknologi Hadoop Berdasarkan Analisis Kinerja Aplikasi

Big Data Infrastructure Design Optimizes Using Hadoop Technologies Based on Application Performance Analysis

[1]Shafiyah*, [2]Ahmad Syauqi Ahsan, [3]Rengga Asmara
Teknik Informatika, Departemen Teknik Informatika & Komputer,
Politeknik Elektronika Negeri Surabaya Jl. Raya ITS – Kampus PENS Sukolilo
Kota Surabaya 60111, Jawa Timur
*e-mail: shafiyah4@gmail.com

(***received*:** 16 Juli 2021, ***revised*:** 29 Juli 2021, ***accepted*:** 18 September 2021)

Abstrak

Infrastruktur Big Data merupakan teknologi yang menyediakan kemampuan untuk menyimpan, memproses, menganalisis, dan memvisualisasikan data yang berukuran besar. Alat dan aplikasi yang digunakan menjadi salah satu tantangan ketika membangun infrastruktur big data. Pada penelitian ini kami menawarkan sebuah strategi baru guna mengoptimasi desain infrastruktur big data yang merupakan bagian penting dalam pengolahan big data dengan cara melakukan analisis kinerja aplikasi yang digunakan pada setiap tahap pemrosesan big data. Proses diawali dari mengumpulkan data yang bersumber dari berita online dengan menggunakan metode *web crawler* dengan menerapkan aplikasi *Scrapy* dan *Apache Nutch.* Selanjutnya menerapkan teknologi *Hadoop* untuk mempermudah dalam penyimpanan dan komputasi big data secara terdistribusi. *No-SQL database Mongo DB* dan *HBase* untuk mempermudah melakukan *query data*, Setelah itu membangun mesin pencari menggunakan *Elasticsear*ch dan *Apache Solr*. Data yang tersimpan kemudian di analisis menggunakan *Apache Hive, Pig dan Spark*. Data yang telah dianalisis kemudian ditampilkan dalam website menggunakan aplikasi *Zeppelin, Metabase, Kibana dan Tableau*. Skenario pengujian terdiri dari jumlah server dan file yang digunakan. Parameter pengujian mulai dari kecepatan proses, penggunaan memori, penggunaan CPU, *throughput* dll. Hasil pengujian kinerja setiap aplikasi kemudian dibandingkan dan dianalisis untuk melihat kelebihan dan kekurangan aplikasi sebagai referensi dalam membangun desain infrastruktur yang optimal sesuai dengan kebutuhan penggunanya. Penelitian ini telah menghasilkan dua alternatif desain infrastruktur big data. Infrastruktur yang disarankan sudah diimplementasikan pada *node-node* komputer di Lab Big Data Pens untuk mengolah big data dari media online dan terbukti dapat berjalan dengan baik.

Kata kunci: Infrastruktur Big Data, Optimasi, Analisis Kineja, *Hadoop*

Abstract

Big data's infrastructure is a technology that provides the ability to store, process, analyze, and visualize large data. The tools and applications used are one of the challenges when building big data's infrastructure. In the study, we offered a new strategy to optimize big data infrastructure design that was an essential part of big data processing by performing performance analysis applications used at each stage of big data processing. The process started from collecting data sourcing online news using web crawler methods using Scrapyand Apache Nutch. Next, implement Hadoop technologies to facilitate the distribution of big data storage and computing. No-sql databases Mongo DB and HBase made it easier to query data, after which they built search engines using Elasticsearch and Apache Solr. Data saved later in analysis using hive apache, pig, and spark. The data has been analyzed was shown on the website using Zeppelins, Metabolase, Kibana, and Tableau. The test scenario consisted of the number of servers and files used. Testing parameters started from process speed, memory usage, CPU usage, throughput, etc. The performance testing results of each application were compared to and analyzed to see the merits and defaults of the application as a reference to building optimal infrastructure design to meet the needs of the user. This research has

produced two big data infrastructure design alternatives. The suggested infrastructure has been implemented on computer nodes in the big data pens for processing big data from online media and proving to be running well.

Keywords: *Big data Infrastructure, Optimization, Analysis Performance, Hadoop*

1 Pendahuluan

Infrastruktur Big Data merupakan dasar dari ekosistem teknologi big data yang dapat melakukan penyimpanan, analitik, dan visualisasi data [1]. Alat dan aplikasi yang digunakan pada infrastruktur big data akan mengubah pusat data secara signifikan pada dekade mendatang[2]. Hal pertama yang perlu diperhatikan dalam membangun infrastruktur big data adalah desain dan rancangan sistem[3]. Desain yang di terapkan tanpa memahami beberapa aspek penting seperti kebutuhan teknologi, kompleksitas teknologi, masalah ketersediaan data, biaya yang tinggi, privasi, dan integritas teknologi. Menjadi solusi yang kurang tepat dalam proses analisis big data sebagai komponen penting dalam pengambilan keputusan [4].

Big data adalah kondisi dimanah model penyimpanan basis data konvensional tidak dapat lagi menanggulangi data dengan jumlah yang besar. Big data memiliki karakteristik lima V, yaitu Volume, Velocity, Variety, Value, dan Veracity. Lima V tersebut yang menjadi tantangan dalam mengelola big data [5]. Data yang kompleks tersebut perlu diolah khusus dengan suatu infrastruktur yang dapat mengelola data dalam volume besar. Namun dalam implementasinya banyak sekali tantangan yang di hadapi saat membangun desain infrastruktur big data. Hal tersebut terkait dengan penggunaan alat dan aplikasi yang digunakan saat pembangunan desain infrastruktur big data. Beberapa tahun ini sudah terdapat banyak sekali perangkat lunak yang dapat digunakan untuk memproses big data. Namun perangkat lunak tersebut belum tentu optimal dalam membangun rancangan infrastruktur big data yang sesuai dengan kebutuhan penggunanya[6]. Desain infrastruktur big data berpengaruh dengan efisiensi pengolahan big data menjadi informasi baru yang di butuhkah di berbagai aspek kehidupan.

Penelitian ini menjadikan Big Data sebagai objek penelitian karena pentingnya big data dalam membantu Organisasi dalam mengelola data berukuran besar untuk mendapatkan peluang baru, yang mengarah pada peluang bisnis cerdas, operasi yang lebih efisien, peningkatan laba dan meningkatkan tingkat kebahagiaan pelanggan. Penelitian ini mengajukan suatu pendekatan baru mengenai pembuatan desain infrastruktur big data yang optimal dengan cara melakukan evaluasi dan analisa kinerja aplikasi pada setiap tahap pemrosesan big data[10]. Analisis kinerja aplikasi yang digunakan berpengaruh dengan optimasi desain infrastruktur big data yang dibangun. Dengan demikian, bentuk penyelesaian dari penelitian ini adalah hasil analisa kinerja aplikasi pada setiap tahap pemrosesan big data dan sebuah desain infrastruktur big data yang di buat berdasarkan acuan dari hasil analisa kinerja aplikasi big data yang telah dihasilkan sebelumnya, dengan desain topologi jaringan big data yang telah di tentukan.

2 Tinjauan Literatur

Kebutuhan akan infrastruktur analitik terintegrasi untuk membantu manajer dalam meningkatkan kinerja operasi dan membongkar silo informasi yang dihasilkan dari analitik data besar[3]. Menggunakan infrastruktur Big Data tanpa memahami masalah biaya tinggi, kompleksitas teknologi, ketersediaan data, privasi, dan integritas mungkin belum tentu menjadi cara yang tepat untuk organisasi mana pun karena Big Data adalah komponen penting dari pengambilan keputusan manajemen yang membutuhkan kapabilitas baru, serta perubahan organisasi dan budaya[4]. Infrastruktur big data yang handal mampu beradaptasi dengan kondisi yang ada yang memberikan kinerja dan kemampuan yang memadai untuk melakukan penambangan informasi untuk membantu dalam pengambilan keputusan[7]. Merancang algoritme dan infrastruktur untuk mengoptimalkan pergerakan data untuk data yang kurang terstruktur dalam analisis grafik skala besar akan mendukung semua kelas beban kerja yang penting[8].

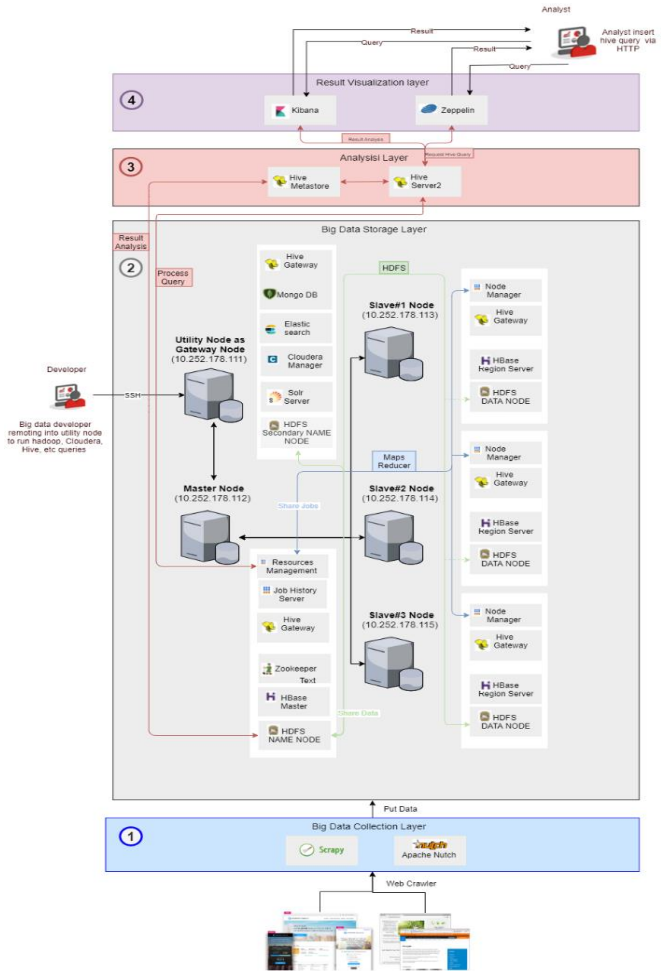
Tidak banyak makalah akademis yang terkait dengan Big Data, kebanyakan kasus difokuskan pada beberapa teknologi komponen (misalnya Analisis Data atau Pembelajaran Mesin) atau solusi itu mencerminkan hanya sebagian kecil dari keseluruhan area masalah. Sama berkaitan dengan definisi Big Data yang akan memberikan konseptual dasar untuk pengembangan teknologi[4]. Terdapat

ISSN:2302-8149
e-ISSN:2540-9719

tantangan – tantangan yang akan dihadapi peneliti saat memproses big data di setiap tahapan pemrosesan data. Pada layer pengambilan data tantangan mengenai ketersediaan data, privasi dan keamanan data. Pada layer penyimpanan data tantangan mengenai skalabilitas, ketersediaan, dan integritas data. Pada layer pre-prosesing data mengenai kualitas data, layer pemrosesan berkaitan dengan permasalahan aliran data dan real-time prosesing, layer analisa data tantangan mengenai jenis data yang heterogen. Layer visualisasi mengenai integrasi dengan sumber data. Peneliti harus siap dengan tantangan – tantangan yang akan dihadapi dalam mengelola data berukuran besar. pipa pemrosesan big Data diajukan dalam upaya untuk merangkum prosedur pemrosesan terkait analitik media sosial[9].

3 Metode Penelitian

Penelitian ini mengimplantasikan teknologi Hadoop yang di terapkan pada Laboratorium Big Data, Program Studi Teknik Informatika, Politeknik Elektronika Negeri Surabaya. Hadoop diperlukan untuk mempermudah penyimpanan dan pemrosesan secara tersitribusi, Hadoop mengelola data secara parallel ke beberapa node yang dapat memanipulasi data pada node sehingga akses data akan lebih cepet dibandingkan menggunakan jaringan konvensional. Penelitian ini menggunakan lima buah komputer yang terdiri dari satu komputer berperan sebangai name node. Name node sendiri adalah komputer yang berperan sebagai master dari HDFS (Hadoop Distributed File System), Data node, Mengatur bagaimana file dibagi dalam blok dan penyimpanannya, Membagi job ke data node. Satu komputer berperan sebagai utility node yang memiliki tugas jika name node mati dan diganti dengan name node baru maka name node baru bisa langsung bekerja dengan mengambil data dari secondary name node. Serta tiga buah komputer berperan sebagai data node atau yang biasa di sebut sebagai worker yang memiliki peran sebagai tempat penyimpanan blok, menerima instruksi dari name node. Berikut ini detail desain infrastruktur big data berserta teknologi yang di gunakan.



Gambar 1. Infrastruktur Big Data Beserta Aplikasi yang di Gunakan

Gambar 1 merupakan infrastruktur big data beserta aplikasi yang di gunakan yang di terapkan dalam penelitian ini. Berikut ini penjelasan dari setiap bagian tahapan.

1. Layer Pengumpulan Data

Pada lapisan ini ditujukan untuk mengumpulkan semua jenis data. web crawler adalah program yang secara otomatis melintasi struktur hyperlink Web dan mengunduh setiap halaman yang terhubung ke penyimpanan lokal. Sumber data yang digunakan pada penelitian ini berasal dari situs berita online lokal yang di ambil dengan metode web crawler. Penelitian ini hanya berfokus pada aplikasi pengambilan data menggunakan metode web crawler aplikasi tersebut antara lain Apache Nutch dan Scrapy. Pada tahap ini akan dilakukan pengujian antara Aplikasi Nutch dan Scrapy pengujian dilakukan dengan berbagai parameter yang memiliki nilai output acuhan. Pada parameter throughput kriteria output penilaian yaitu nilai throughput yang lebih besar dalam satuan (operasi / detik). Selanjutnya pada parameter waktu eksekusi kriteria output penilaian yaitu waktu input data yang lebih cepat dalam satuan detik. Kemudian parameter penggunaan CPU kriteria output penilaian yaitu Performa penggunaan CPU yang lebih Kecil dalam satuan persen. Terakhir pada parameter penggunaan memori kriteria output peilaian yaitu Performa penggunaan memori yang lebih kecil dalam satuan persen.

2. Layer Penyimpanan Data

Pemrosesan paralel big data di implementasikan menggunakan Apache Hadoop. Apache Hadoop menyediakan layanan penyimpanan data secara terdistribusi ke beberapa komputer (HDFS). Sekaligus juga menyediakan layanan pemrosesan big data terdistribusi (Maps Reducer) yang mempermuda kompilasi data. No-SQL database di perlukan untuk mempermudah melakukan query data di HDFS. NoSQL database tersebut antara lain HBase dan Mongo DB. Pada bagian ini akan dilakukan pengujian kepada dua aplikasi tersebut pengujian dilakukan dengan berbagai parameter yang memiliki nilai output acuhan dalam penilaian hasil pengujian pada setiap parameter. Pada parameter throughput kriteria output penilaian yaitu nilai throughput yang lebih besar dalam satuan (operasi / detik). Selanjutnya pada parameter waktu eksekusi kriteria output penilaian yaitu waktu input data yang lebih cepat dalam satuan detik. Kemudian parameter penggunaan CPU kriteria output penilaian yaitu Performa penggunaan CPU yang lebih Kecil dalam satuan persen. Terakhir pada parameter penggunaan memori kriteria output penilaian yaitu Performa penggunaan memori yang lebih kecil dalam satuan persen.

Aplikasi search engine juga di perlukan untuk mempermudah pencarian data dan melakukan indexing data. Aplikasi tersebut antara lain Apache Solr dan Elasticsearch. Pada bagian ini akan dilakukan pengujian kepada dua aplikasi tersebut pengujian dilakukan dengan berbagai parameter yang memiliki nilai output acuhan dalam penilaian hasil pengujian pada setiap parameter. Pada parameter waktu pembagian data kriteria output penilaian yaitu Waktu pembagian data yang lebih cepat dalam milisecond. Selanjutnya pada parameter waktu query data kriteria output penilaian yaitu waktu melakukan query data yang lebih cepat dalam satuan millisecond.

3. Layer Analisa Data

Pemrosesan data di Big Data untuk data besar yang terdapat di HDFS menghasilkan analisa data yang lebih detail sesuai dengan kebutuhan user. Hasil dari pemrosesan ini dimasukkan ke dalam data store untuk kemudian bisa di lihat di level aplikasi. Pada penelitian ini data berbentuk bentuk text akan dianalisis menggunakan metode sentimen analisis based on lexicon based.

Aplikasi Analisa Big data yang di gunakan pada penelitian ini antara lain Apache Hive, Apache Spark dan Apache Pig. Kemudian penjelasan mengenai kriteria output yang disajikan sebagai acuhan dalam penilaian hasil pengujian pada setiap parameter. Pada parameter waktu input data kriteria output penilaian yaitu waktu input data yang lebih cepat dalam milisecond. Selanjutnya pada parameter waktu pemotongan kalimat menjadi kata data kriteria output penilaian yaitu waktu melakukan pemotongan data yang lebih cepat dalam satuan millisecond.

4. Layer Visualisasi Data

Pada umumnya aplikasi di sini hanyalah untuk melakukan visualisasi dari data yang sudah dianalisis sebelumnya. aplikasi yang berinteraksi langsung dengan user. Aplikasi di sini mengakses

 ISSN:2302-8149
e-ISSN:2540-9719

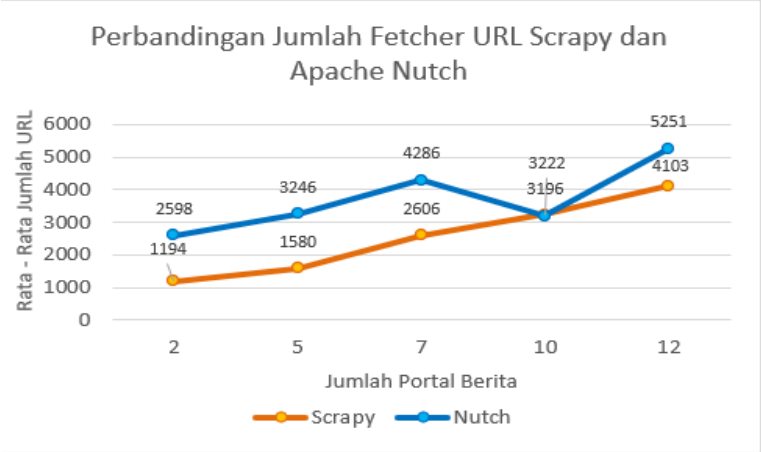
data yang berada di data store untuk kemudian disajikan kepada user yang di sesuaikan dengan kebutuhan user. Aplikasi Visualisasi data yang diujikan antara lain diujikan adalah Apceh Zeppline, Kibana, Meabase dan Tableau. Kemudian penjelasan mengenai kriteria output yang disajikan sebagai acuhan dalam penilaian hasil pengujian pada setiap parameter. Pada parameter proses instalasi data kriteria output penilaian yaitu proses instalasi yang lebih muda dan lebih cepat. Selanjutnya pada parameter fitur – fitur data kriteria output penilaian yaitu fitur - fitur yang lengkap dalam mendukung penampilan data. Kemudian pada parameter perangkat pendukung aplikasi kriteria output penilaian yaitu kemampuan aplikasi dieksekusi tanpa bantuan aplikasi alternatif pendukung. Terakhir pada parameter Tampilan antar muka kriteria output penilaian yaitu bentuk, warna, struktur grafik dalam menampilkan data.

4 Hasil dan Pembahasan

Pada bagian ini dijelaskan hasil pengujian aplikasi serta analisa hasil pengujian dalam grafik disertai penjelasan hasil analisa pada setiap layer big data.

1. Hasil Perbandingan Pengujian Aplikasi pada Layer Pengambilan Data

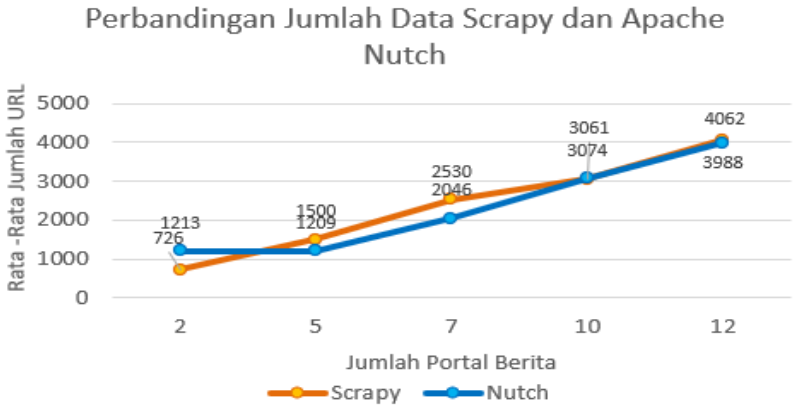
Perbandingan pengujian pada layer pengambilan data dilakukan pada aplikasi web crawler yaitu Scrapy dan Apache Nutch. Scrapy yang berbasis bahasa pemrograman Python [11] dan Apache Nuch yang berbasis bahasa pemrograman Java[12]. Jumlah hyperling portal berita sebanyak 2,5,7,10 dan 12 menjadi variable terkenali. Data yang dihasilkan dari proses perayapan berformat Json. Parameter pengujian berupa Jumlah halaman yang berhasil di rayap, jumlah data yang di dapat, penggunaan CPU dan memori, dan waktu eksekusi.



Gambar 2. Grafik Perbandingan Jumlah Fetche URL Scrapy dan Apache Nutch

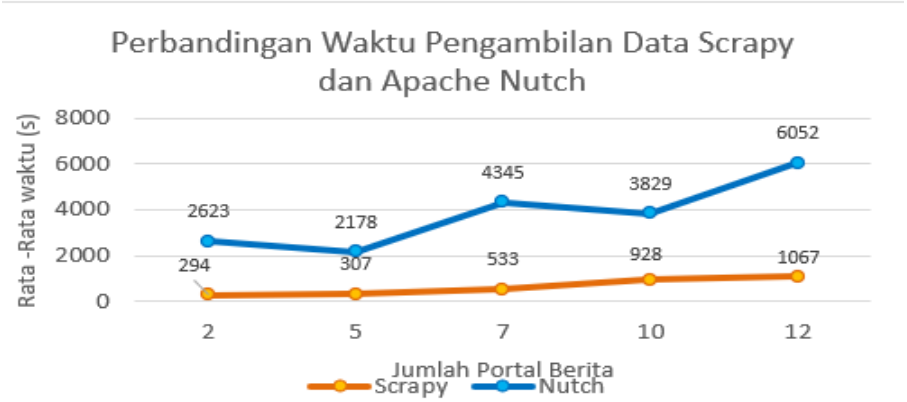
Gambar 2 merupakan grafik hasil percobaan perbandingan jumlah fetch URL antara aplikasi Apache Nutch dan Scrapy. Pengujian menggunakan dua portal berita Apache Nutch dapat melakukan fetching URL rata – rata sebanyak 2598 URL sedangkan Scrapy dapat melakukan fetching URL rata – rata sebanyak 1194 URL dari lima belas kali percobaan. Kemudian percobaan menggunakan dua belas portal berita Apache Nutch dapat melakukan fetching URL rata – rata sebanyak 5251 URL, sedangkan Scrapy dapat melakukan fetching URL rata – rata sebanyak 4103 URL dari lima belas kali percobaan.Sehingga dapat disimpulkan dua aplikasi tersebut mampu melakukan fetching URL dengan hasil yang besar. Dari hasil tersebut juga terlihat Apache Nutch mampu melakukan fetching URL dengan hasil yang lebih besar dibandingkan Scrapy. Jumlah fetch url di Apache Nutch memiliki nilai yang lebih besar di pengaruhi oleh nilai Top-N generate dan deep yang nilainya dapat diatur oleh user. Sehingga user dapat bebas mengubah kedalaman perayapan data.

ISSN:2302-8149
e-ISSN:2540-9719



Gambar 3. Grafik Perbandingan Jumlah Data Scrapy dan Apache Nutch

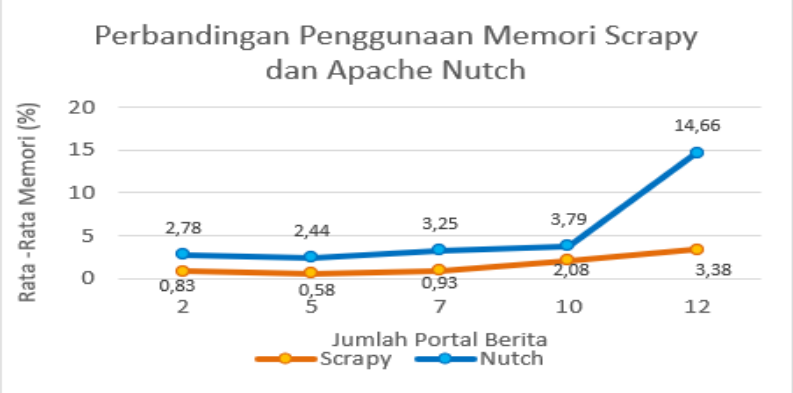
Gambar 3 merupakan grafik hasil percobaan perbandingan jumlah data yang di dapatkan antara aplikasi Apache Nutch dan Scrapy. Pergujian menggunakan dua portal berita Scrapy mendapatkan hasil rata – rata 726 objek data berbentuk JSON, sedangkan Apache Nutch mendapatkan rata – rata 1213 objek data berbentuk JSON. Pengujian kembali menggunakan dua belas portal berita Scrapy mendapatkan rata – rata 4062 objek data sedangkan Apache Nutch mendapatkan rata – rata 1917 objek data. Hasil perolehan data menggunakan Scrapy meningkat seiring dengan penambahan jumlah portal berita yang diujikan. hal tersebut berkaitan degan keandalan Scrapy saat melakukan perayapan data secara terfokus. Sedangkan Apache Nutch terlihat stabil dan terukur seiring dengan meningkatnya jumlah portal yang diujikan.



Gambar 4. Grafik Perbandingan Waktu Pengambilan Data Scrapy dan Apache Nutch

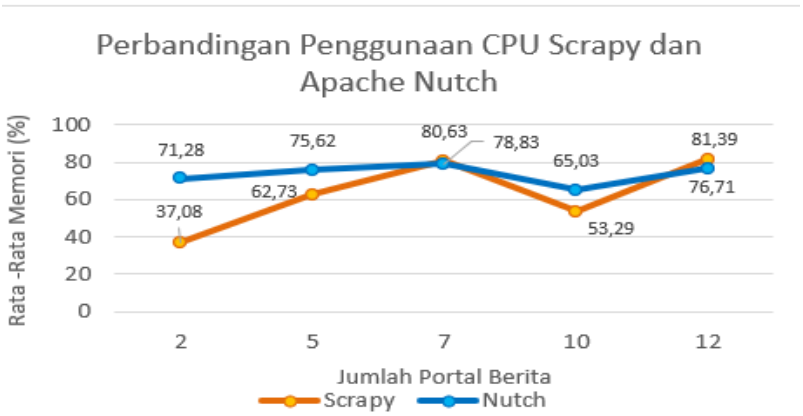
Gambar 4 merupakan grafik hasil percobaan perbandingan waktu pengambilan data antara aplikasi Apache Nutch dan Scrapy. Pengujian menggunakan dua portal berita Scrapy mendapatkan hasil rata – rata waktu sebesar 294 detik, sedangkan Apache Nutch mendapatkan rata – rata waktu sebesar 2623 detik. Pengujian kembali menggunakan dua belas portal berita Scrapy mendapatkan mendapatkan rata – rata waktu sebesar 1067 detik sedangkan Apache Nutch mendapatkan rata – rata waktu sebesar 6052 detik. Scrapy lebih cepat dibandingkan Apache Nuch hal tersebut dibuktikan dengan percobaan menggunakan varian jumlah portal. Kecepatan kedua aplikasi memiliki perbandingan lurus dengan banyaknya portal berita yang digunakan. Kecepatan Scrapy dalam mengambil data dipengaruhi oleh metode Xpath yang digunakan.

ISSN:2302-8149
e-ISSN:2540-9719



Gambar 5. Grafik Perbandingan Penggunaan Memori Scrapy dan Apache Nutch

Gambar 5 merupakan grafik hasil percobaan perbandingan penggunaan memori antara aplikasi Apache Nutch dan Scrapy. Pengujian menggunakan dua portal berita Scrapy mendapatkan hasil rata – rata penggunaan memori sebesar 0,83 %, sedangkan Apache Nutch mendapatkan rata – rata pemakaian memori sebesar 2,78. Pengujian kembali menggunakan dua belas portal berita Scrapy mendapatkan mendapatkan rata – rata memori sebesar 3,38 % sedangkan Apache Nutch mendapatkan rata – rata memori sebesar 14,66 %. Scrapy memerlukan memori yang lebih sedikit dibandingkan dengan Apache Nutch. Perbandingan penggunaan memori dengan parameter jumlah url dan sedikit menunjukkan Scrapy menggunakan memori yang lebih rendah. Apache Nutch menunjukkan nilai penggunaan memori yang terukur saat bekerja di atas Hadoop.



Gambar 6. Grafik Perbandingan Penggunaan CPU Scrapy dan Apache Nutch

Gambar 6 merupakan grafik hasil percobaan perbandingan penggunaan CPU antara aplikasi Apache Nutch dan Scrapy. Pengujian menggunakan dua portal berita Scrapy mendapatkan hasil rata – rata penggunaan CPU sebesar 37,08 %, sedangkan Apache Nutch mendapatkan rata – rata pemakaian CPU sebesar 71,28%. Pengujian kembali menggunakan dua belas portal berita Scrapy mendapatkan mendapatkan rata – rata CPU sebesar 76,71 % sedangkan Apache Nutch mendapatkan rata – rata CPU sebesar 81,39 %. Scrapy lebih sedikit menggunakan CPU dibandingkan dengan Apache Nutch. Apache Nutch sendiri memiliki nilai penggunaan CPU yang stabil saat memproses data berukuran kecil sampai ke data berukuran besar.

2. Hasil Perbandingan Pengujian Aplikasi pada Layer Penyimpanan Data

Perbandingan pengujian pada layer Penyimpanan data dilakukan pada dua tahan. Tahap pertama pengujian pada No-SQL database yaitu Hbase dan Mongo DB. Database Mongo DB merupakan Tahap kedua pengujian pada aplikasi search engine Elasticsearch dan Solr. HBase yang merupakan

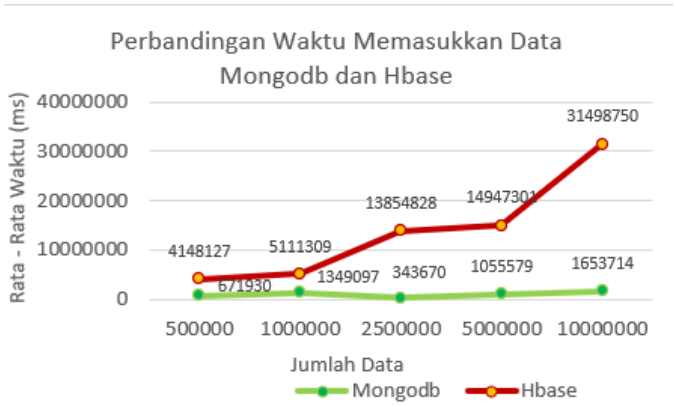
 ISSN:2302-8149
e-ISSN:2540-9719

database open-source non-relasional terdistribusi yang dimodelkan setelah Google Big table dan ditulis dalam Java. Selain itu Mongodb merupakan sistem basis data berorientasi dokumen lintas platform.database ini mendukung file berformat JSON[13]. Apache Solr merupakan projek dari Apache Lucene yang memiliki fungsi utama untuk pencarian teks lengkap, dan pengindeksan real-time[14]. Elastic search merupakan open sources untuk melakukan pencarian serta analisa terdistribusi. dikembangkan menggunakan perpustakaan Lucene[15].

Tabel 1. Data Uji pada Pengujian Database

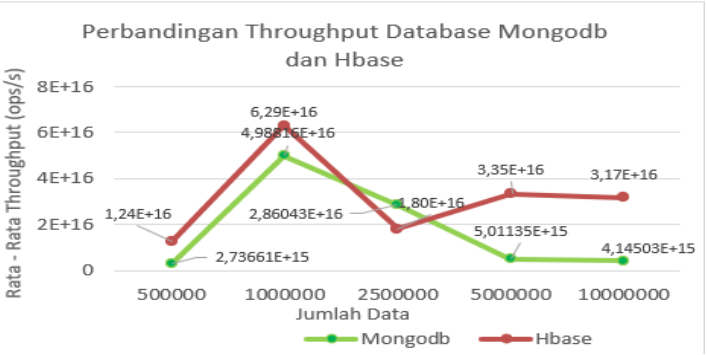
No	Jumlah Baris Data	Ukuran dalam Gigabyte
1	500000 baris	500 Megabyte
2	100000 baris	1 Gigabyte
3	2500000 baris	2 Gigabyte
4	5000000 baris	4 Gigabyte
5	10000000 baris	8 Gigabyte

Tabel 1 merupakan detail jumlah data pengujian No-SQL database menggunakan YCSB (Yahoo! Cloud Serving Benchmark). Data berupa random data dari beberapa karakter, A -Z, a-z dan 0-9. Junlah data yang digunakan antara lain 500000 baris atau setara dengan 500 mb, sampai dengan 10000000 baris atau setara dengan 8 GB. Adapun parameter uji yang digunakan antara lain throughput, waktu eksekusi, penggunaan memori dan CPU.



Gambar 7. Grafik Perbandingan Waktu Input Data Hbase dan Mongo DB

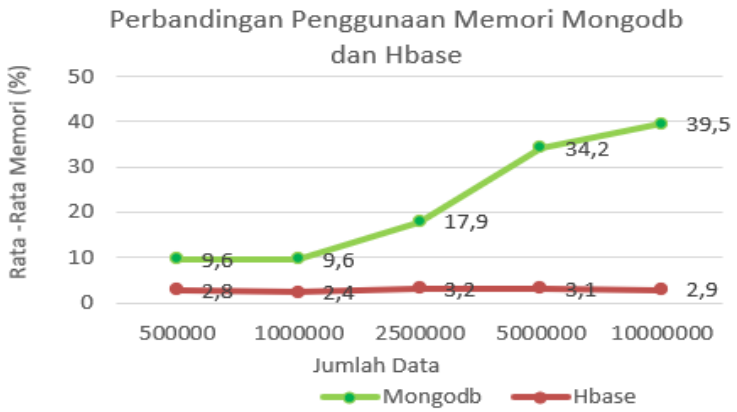
Gambar 7 merupakan grafik hasil perbandingan pengujian waktu memasukkan data antara No-SQL database HBase dan MongoDB. Pengujian menggunakan 500Mb data HBase mendapatkan hasil rata – rata waktu sebesar 4148127 ms, sedangkan MongoDB mendapatkan rata – rata waktu sebesar 671930 ms. Pengujian kembali menggunakan 10 GB data Hbase mendapatkan mendapatkan waktu sebesar 31498750 ms sedangkan MongoDB mendapatkan rata – rata waktu sebesar 1653714 ms. Database MongoDB bekerja lebih dibandingkan database HBase. Lamanya waktu eksekusi database HBase di pengaruhi oleh jenis struktur database yang terinstal secara terdistribusi. Sehingga saat proses memasukkan data memerlukan waktu untuk melakukan pembagian atau pemecahan ke beberapa node HBase.



Gambar 8. Perbandingan Throughput Hbase dan Mongo DB

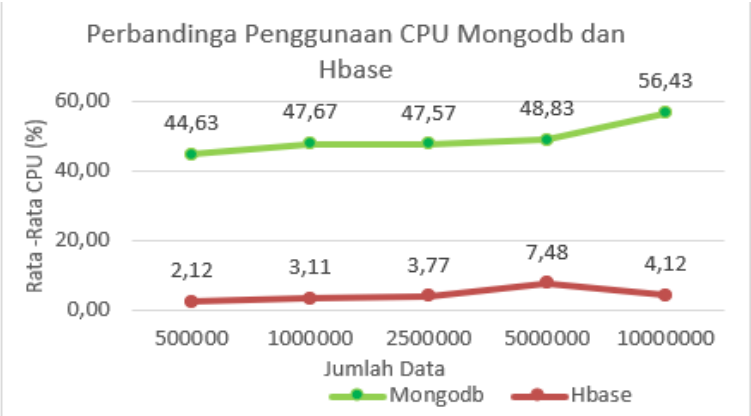
ISSN:2302-8149
e-ISSN:2540-9719

Gambar 8 merupakan grafik hasil perbandingan pengujian throughput memasukkan data antara No-SQL database HBase dan MongoDB. Pergujian menggunakan 1 GB data HBase mendapatkan hasil rata – rata throughput sebesar 12.439.764.615.225.800 ops/s, sedangkan MongoDB mendapatkan rata – rata throughput sebesar 2736613826085190 ops/s. Pengujian kembali menggunakan 8 GB data HBase mendapatkan mendapatkan throughput sebesar 31.747.283.496.066.700 ops/s, sedangkan MongoDB mendapatkan rata – rata throughput sebesar 4145031130633430 ops/s. Throughput MongoDB lebih kecil dibandingkan dengan database HBase pada pengujian dengan data berukuran besar maupun kecil. Ukuran throughput yang di hasilkan oleh database berpengaruh dengan kecepatan database dalam mengeksekusi data, semakin cepat maka tharoughput yang di hasilkan akan semakin kecil.



Gambar 9. Grafik Perbandingan Penggunaan Memori Hbase dan Mongo DB

Gambar 9 merupakan grafik hasil perbandingan pengujian penggunaan memori memasukkan data antara No-SQL database HBase dan MongoDB. Pengujian menggunakan 1 GB data HBase mendapatkan hasil rata – rata memori sebesar 2,8 %, sedangkan MongoDB mendapatkan rata – rata memori sebesar 9,6 %. Pengujian kembali menggunakan 8 GB data HBase mendapatkan mendapatkan memori sebesar 2,9 %, sedangkan MongoDB mendapatkan rata – rata memori sebesar 39,5 %. Mongo DB menggunakan memori lebih besar di bandingkan dengan HBase untuk pegujian data berukuran besar maupun kecil. MongoDB menggunakan memori yang lebih banyak dipengaruhi oleh desain struktur databasenya yang hanya terinstal di satu komputer yang membuat beban memori lebih banyak. Dibandingkan dengan database HBase yang terinstal secara terdistribusi di atas Hadoop.

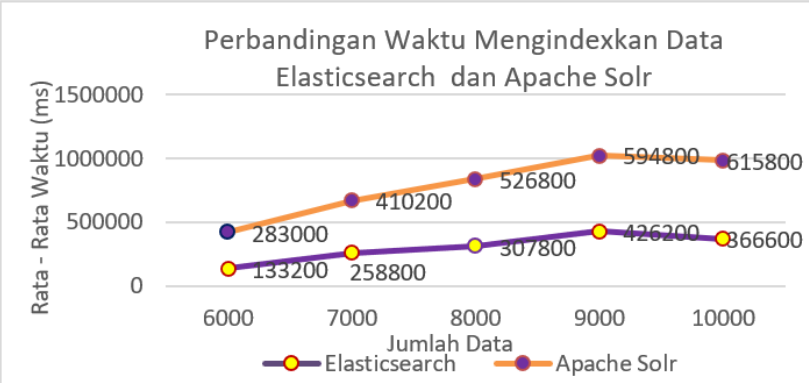


Gambar 10. Grafik Perbandingan Penggunaan CPU Hbase dan Mongo DB

Gambar 10 merupakan grafik hasil perbandingan pengujian penggunaan CPU memasukkan data antara No-SQL database HBase dan MongoDB. Pergujian menggunakan 1 GB data HBase mendapatkan hasil rata – rata CPU sebesar 2,12 %, sedangkan MongoDB mendapatkan rata – rata CPU sebesar 44,63 %. Pengujian kembali menggunakan 8 GB data HBase mendapatkan hasil

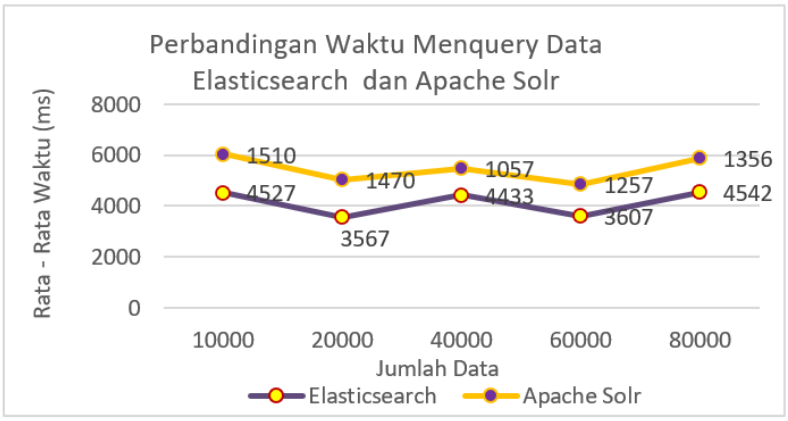
 ISSN:2302-8149
e-ISSN:2540-9719

sebesar 4,12 %, sedangkan MongoDB mendapatkan rata – rata CPU sebesar 56,43 %. penggunaan CPU database Mongodb lebih besar di bandingkan dtabase HBase Pemakaian CPU yang sedikit pada database HBase dipengaruhi oleh instalasi HBase yang terdistribusi membuat beban kerja terbagi ke beberapa komputer.



Gambar 11. Grafik Perbandingan Waktu Mengindexkan Data Solr dan Elasticsearch

Gambar 11 merupakan hasil pengujian perbandingan waktu mengindekan data aplikasi search engine Apache Solr dan Elasticsearch. Percobaan pengujian menggunakan 6000 objek data Elasticsearch membutuhkan waktu sebesar 133200 ms, sedangkan Apaceh Solr membutuhkan waktu 283000 ms. Kemudian percobaan menggunakan 10000 objek data Elasticsearch membutuhkan waktu sebesar 866600 ms, sedangkan Apache Solr membutuhkan waktu 615800 ms. Elastisearch lebih cepat di bandingkan Apache Solr. Hasil pengujian kedua aplikasi juga menunjukkan jumlah data mempengaruhi waktu pengindeksan data. Semakin banyak data maka semakin bayak waktu yang di perlukan untuk mengindekkan data. Keduanya memiliki struktur search engine yang sama yaitu secara terdistribusi ke beberapa node melalui Zookeeper.



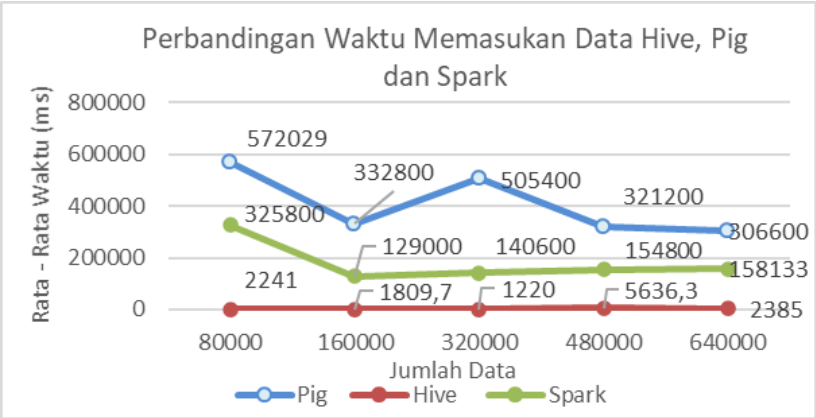
Gambar 12. Grafik Perbandingan Waktu Mengquery Data Solr dan Elasticsearch

Gambar 12 merupakan hasil pengujian perbandingan waktu mengquery data aplikasi search engine Apache Solr dan Elasticsearch. Percobaan pengujian menggunakan 10000 objek data Elasticsearch membuahkan waktu sebesar 4527 ms, sedangkan Apaceh Solr membutuhkan waktu 1510 ms. Kemudian percobaan menggunakan 80000 objek data Elasticsearch membutuhkan waktu sebesar 1356 ms, sedangkan Apache Solr membutuhkan waktu 4542 ms. Waktu mengquery data Apache Solr lebih cepat di bandingkan Elastisearch. Hasil pengujian kedua aplikasi juga menunjukkan jumlah data tidak mempengaruhi waktu pengindeksan data. Penambahan jumlah data menunjukkan waktu pencarian yang cenderung stabil.

 ISSN:2302-8149
e-ISSN:2540-9719

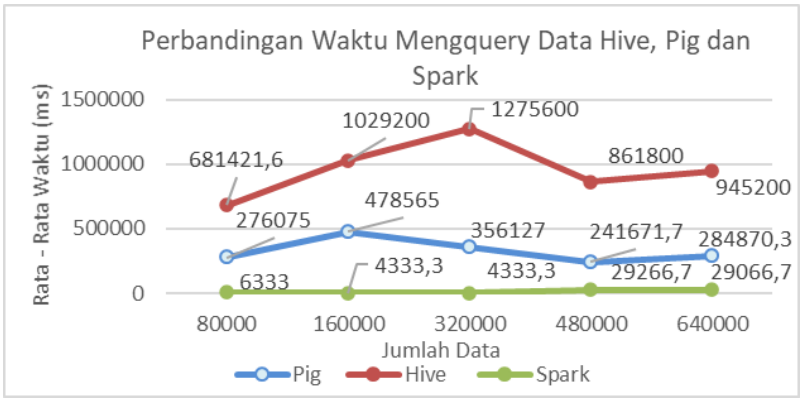
3. Hasil Perbandingan Pengujian Aplikasi pada Layer Analisa Data

Perbandingan pengujian pada layer analisa data yang dilakukan pada tiga aplikasi analisis big data. Aplikasi yang diujikan antara lain Apache Hive, Apace Pig dan Spark. Pengujian dilakukan untuk mengukur performa waktu aplikasi saat memasukan data dan melakukan query data. pengujian dilakukan dengan menggunakan data berita berbentuk objek yang telah tersimpan pada HDFS dengan varian jumlah data.



Gambar 13. Grafik Analisa Hive, Pig dan Spark Input Data Waktu Input Data

Gambar 13 merupakan hasil pengujian perbandingan waktu memasukan data antara Apache Hive, Apche Pig dan Spark. Pengujian menggunakan 80000 object data Apache Hive memerlukan rata – rata waktu sebesar 2241 detik, Apache Pig 572029 detik dan Spark 325800 detik. Kemudian pengujian dengan 640000 objek data Apache Hive memerlukan waktu 2385 detik, Apche Pig806600 detik dan Spark 158133 detik. Apache hive memiliki waktu input data yang lebih cepat dibandingkan dengan Apache Pig dan Spark. Hive mempermudah menulis pekerjaan di mapreducer karena semua kueri Hive dikonversi menjadi pekerjaan MapReduce yang sesuai di bawah bagian yang berjalan di cluster. Pig memungkinkan untuk tidak menulis program MapReduce dan melakukan operasi melalui kueri Pig langsung. Sedangkan Spark bukan database, melainkan kerangka kerja yang dapat mengakses kumpulan data terdistribusi eksternal menggunakan metodologi RDD (Resilient Distributed Data) dari penyimpanan data seperti Hive, Hadoop, dan HBase.



Gambar 14. Grafik Analisa Hive, Pig dan Spark Input Data Waktu Mengquery Data

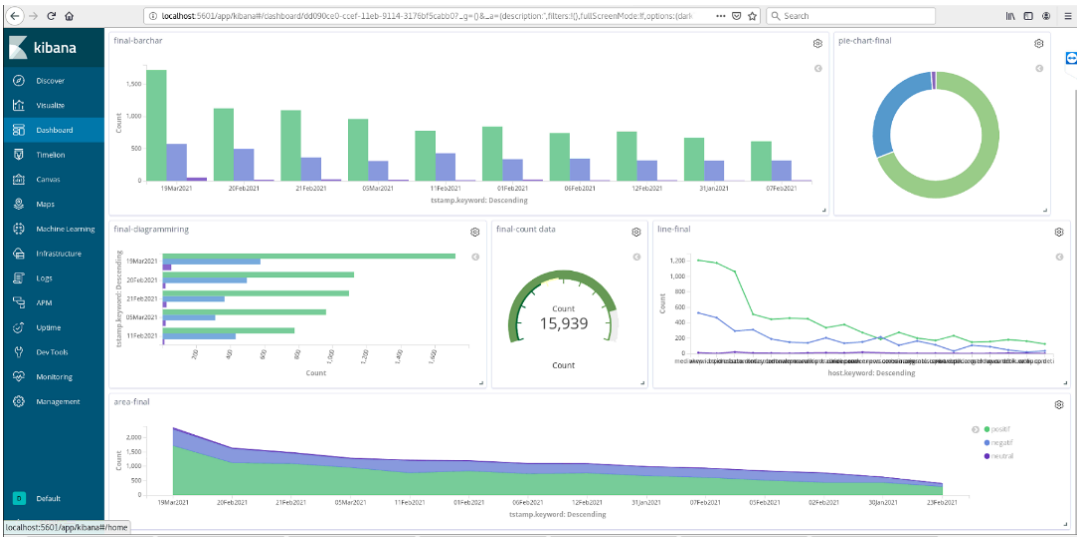
Gambar 14 merupakan hasil pengujian perbandingan waktu mengquery data antara Apache Hive, Apche Pig dan Spark. Pengujian menggunakan 80000 object data Apache Hive memerlukan rata – rata waktu sebesar 681421,6 detik, Apache Pig 276075 detik dan Spark 6333 detik. Kemudian pengujian dengan 640000 objek data Apache Hive memerlukan waktu 945200 detik, Apche Pig 284870,3 detik dan Spark 29066,7 detik. Spark memiliki waktu query data yang lebih cepat dibandingkan dengan Apache Hive dan Apache Pig. Kekuatan inti Spark adalah kemampuannya

ISSN:2302-8149
e-ISSN:2540-9719

untuk melakukan analitik dalam memori yang kompleks dan mengalirkan data berukuran hingga petabyte, membuatnya lebih efisien dan lebih cepat daripada MapReduce. Spark dapat menarik data dari penyimpanan data apa pun yang berjalan di Hadoop dan melakukan analitik kompleks dalam memori dan secara paralel. Kemampuan ini mengurangi Disk I/O dan pertentangan jaringan, membuatnya sepuluh kali atau bahkan seratus kali lebih cepat. Selain itu, kerangka kerja analitik data di Spark dapat dibangun menggunakan Java, Scala, Python, R, atau bahkan SQL.

4. Perbandingan Aplikasi pada Layer Visualisasi Data

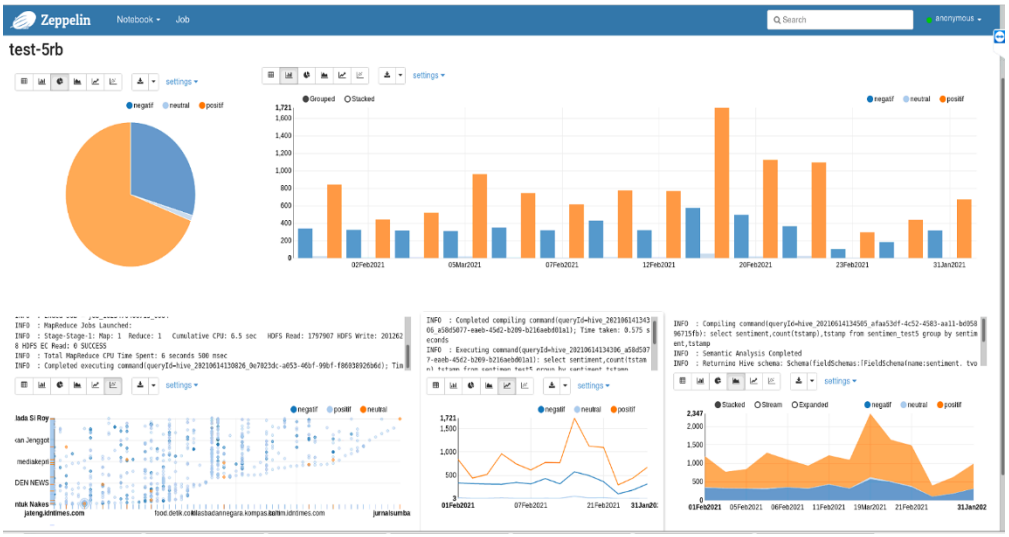
Perbandingan pengujian pada layer visualisasi data yang dilakukan pada empat aplikasi visualisasi big data. Aplikasi yang diujikan antara lain Kibana, Apache Zeppelin, Metabase dan Tableau. Pengujian dilakukan untuk dengan beberapa parameter antara lain integrasi ke sumber data, fitur fisualisasi, kelebihan, kekurangan dan lisensi. Pengujian dilakukan dengan menggunakan datahasil analisis dari layer sebelumnya.



Gambar 15. Tampilan Fitur – Fitur Tampilan Antar Muka Kibana

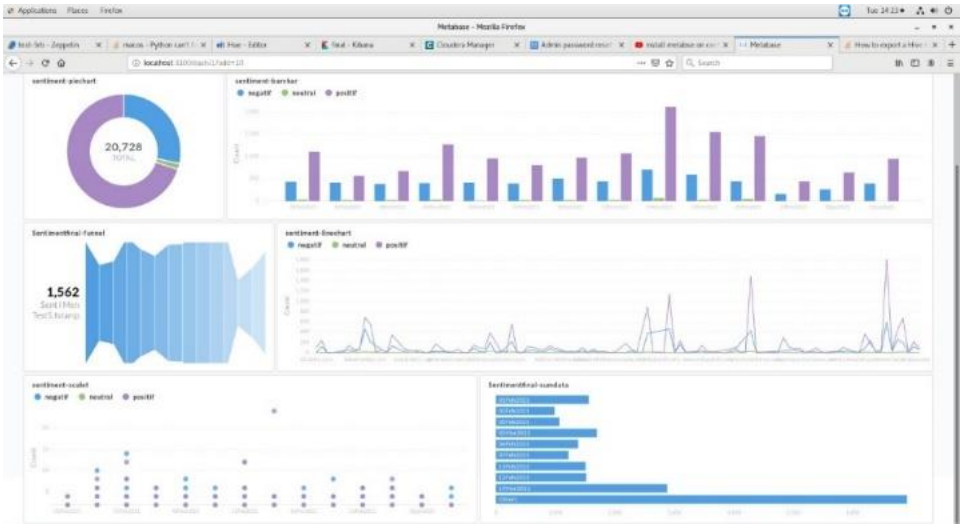
Gambar 15 merupakan hasil visualisasi data menggunakan aplikasi kibana. Aplikasi ini merupakakan solusi open source yang memungkinkan visualisasi data Elasticsearch. Visualisasi yang ditawarkan sangan interatif menampilkan fitur bagan yang beragam mulai dari bagan garis, area, dan batang yang digunakan untuk memplotkan data pada sumbu X/Y. serta diagram lingkaran yang berbentuk seperti lingkaran, serta ada fitur matrik untuk menampilkan progress. Kibana menyediakan fitur dashboard yang dapat merubah data satuan menjadi kumpulan panel. Terdapat fitur pencarian yang dapat mencari nilai dalam suatu tabel, namun fitur pencarian yang dimiliki sedikit rumut karena pencarian harus mengetahui nama kolom pada bagian yang ingin dicari. Perlu adanya usaha lebih bagi pengguna awal karena tampilan untuk memakai sebuah fitur kurang mudah dipahami.

ISSN:2302-8149
e-ISSN:2540-9719



Gambar 16. Tampilan Fitur – Fitur Tampilan Antar Muka Zeppelin

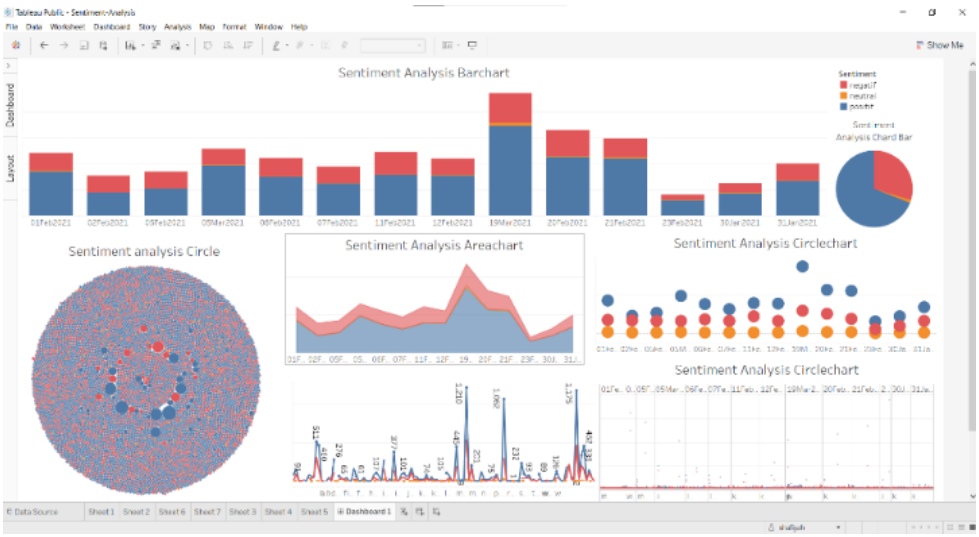
Gambar 16 merupakan hasil visualisasi data menggunakan aplikasi Apache Zeppelin, Aplikasi ini merupakan notebook berbasis web yang memiliki fitur eksplorasi data, visualisasi, berbagi, dan berkolaborasi dengan beberapa aplikasi big data serta mendukung beberpa bahasa pemograman seperti Python, Scala, Hive, SparkSQL, Shell dan Markdown. Fokus utama aplikasi ini adalah menjalankan perintah untuk menampikan data ynag kemudian hasil output dapat di tampillkan dalam bentuk Bagan batang, area, garis, area dan titik. Serta dapat menampilkan data dalam bentuk diagram lingkaran. Jenis bagan terbatas dan opsi penyesuaian terbatas, tampilan sederhana, output kikuk karena mencetak nilai variable di setiap proses.



Gambar 17. Tampilan Fitur – Fitur Tampilan Antar Muka Metabase

Gambar 17 merupakan hasil visualisasi data menggunakan aplikasi Metabase. Aplikasi ini merupakan aplikasi Visualisasi data yang sering digunakan aplikasi ini bersifat (open source), Metabase menggunakan konsep ask questation and display answers dengan konsep mengajukan pertanyaan kemudian hasil pertanyaan disajikan dalam berbagai bentuk antara lain tabel, diagram batang, diagram baris, diagram garis, diagram area, diagram lingkaran, progress bar, angka, tren, scatterplot, combo, funnel, dan peta. Aplikasi Metabase menyediakan fitur dashboard. Metabase dapat berintegrasi dengan beberpa database MySQL, MongoDB, PostgreSQL, Microsoft SQL Server, dan Amazon. Untuk membuat suatu query yang kompleks metabase sangat bergantung dengan MySQL.

ISSN:2302-8149
e-ISSN:2540-9719



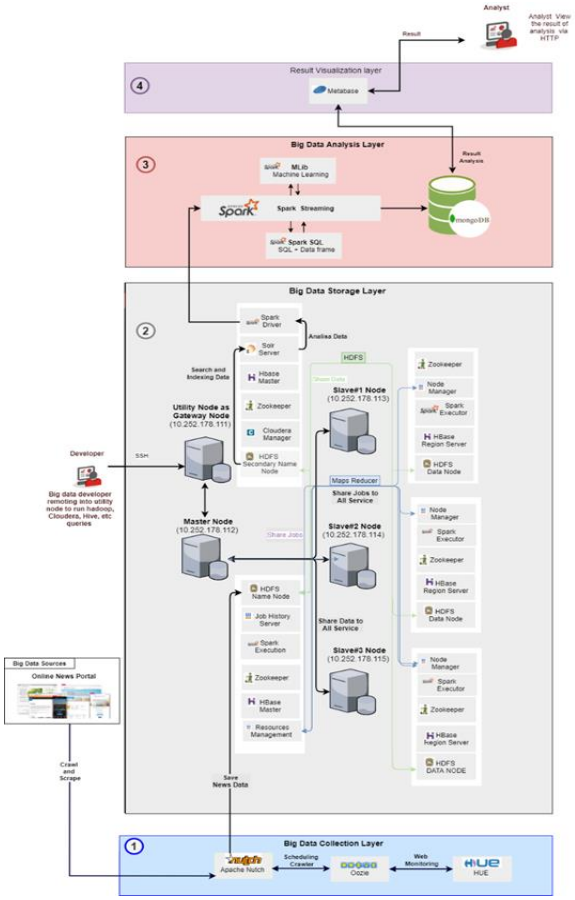
Gambar 18. Tampilan Fitur – Fitur Tampilan Antar Muka Metabase

Gambar 18 merupakan hasil visualisasi data menggunakan aplikasi Tableau. Aplikasi ini merupakan buisiness intelligence sofware yang powerfull untuk visualisasi data. Tableau menyediakan banyak fitur diantaranya dashboard dan scorecards, ad hoc analysis and queries, pemrosesan analitik online, penemuan data, pencarian BI, integrasi spreadsheet, dan lainnya. Tampilan visual sangat menarik dan interatif banyak pilihan grafik yang dapat bergerak. Cara kerjanya sangat mudah tinnggal drag-and-drop field yang akan ditampilkan dalam grafik. Tableau merupakan aplikasi berbayar dengan buaya perbualan mulai dar $35 dan menyediakan versi free trial aplikasi selama 14 hari. Tableau biasanya dipakai oleh perusahaan besar.

4.1 Desain Infrastruktur Big Data Hasil Analisa Kinerja Aplikasi

Hasil analisis kinerja aplikasi yang telah dilakukan tersebut kemudian dijadikan acuhan dalam melakuakan optimasi infrastruktur yang telah ada sebelumnya. Gambar 19 dan 20 merupakan desain infrastruktur big data berdasarkan analisis kinerja aplikasi. Terdapat dua desain nfrastruktur yang direkomendasikan. Masing masing desan menyertakan kelebihan dan kekurangan dari aplikasi yang digunakan sehingga nantinya dapat disesuaikan kembali dengan kebutuhan penggunanya. Untuk lebih detail ada pada gambar di bawah ini:

ISSN:2302-8149
e-ISSN:2540-9719

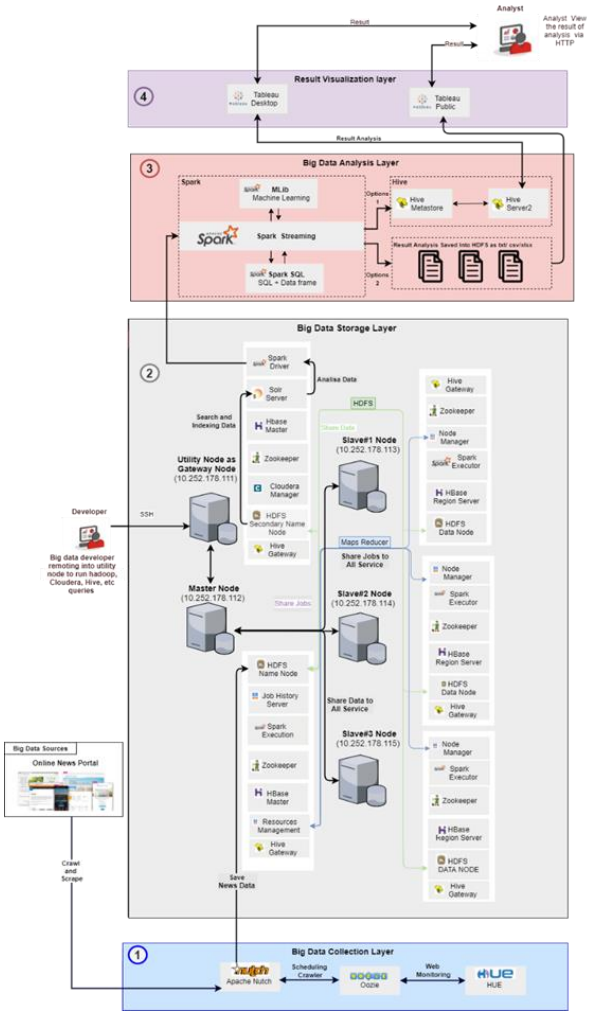


Gambar 19. Rancangan 1 Infrastruktur Big Data Hasil Analisa Aplikasi

Gambar 19 menunjukkan hasil rancangan pertama infrastruktur big data berdasarkan analisa kinerja aplikasi yang telah dilakukan sebelumnya. Proses diawali dari pengambilan data dari portal berita online lokal yang diambil dengan metode web carwling menggunakan Apache Nutch. Nutch lebih cocok dan stabil saat bekerja dengan Hadoop. dibandingkan dengan Scrapy yang lebih cocok untuk fokus crawling. Data berita hasil dari proses crawling disimpan dalam HDFS (Hadoop Distribution File System). Sebelum disimpan data terlebih dahulu diindekan untuk mempermudah dalam pencarian data menggunakan Apache Solr. Apache Solr dipilih karena menurut penelitian yang telah dilakukan Apache Solr lebih unggul dalam melakukan pencarian dan sangat mudah di integrasikan dengan HDFS. Setelah data tersimpan dalam HDFS, data di integrasikan dengan No-SQL database HBase untuk mempermudah melakukan query pada data yang tersimpan dalam HDFS.

Data kemudian dianalisis menggunakan aplikasi Spark. Aplikasi Spark yang digunakan menggunakan konsep master claster. Master claster merupakan proses dimanah job dikendalikan oleh Map reducer, job nantinya akan disebarkan ke semua node manager sehingga pekerjaan akan lebih ringan saat mengelola data berukuran besar. Saat proses Analisa berjalan menggunakan Apache Spark data akan ditampung dalam Spark data frame. Memanfaatkan perpustakaan yang dimiliki oleh Spark MLib yang menyediakan package analisa big data. Data hasil analisa data akan disimpan dalam database Mongo DB. Mongo DB memilik waktu input data yang lebih cepat dibanding database Hbase. Data hasil analisa yang telah tersimpan dalam database akan ditampilkan dalam bentuk grafik menggunakan aplikasi metabase. Metabase merupakan aplikasi open sources memiliki tampilan antar muka yang beragam, terdapat fitur dashboard, serta dapat terintegrasi dengan banyak sumber data seperti database Mongo DB.

ISSN:2302-8149
e-ISSN:2540-9719



Gambar 20. Rancangan 2 Infrastruktur Big Data Hasil Analisa Aplikasi

Gambar 20 menunjukkan hasil rancangan pertama infrastruktur big data berdasarkan analisa kinerja aplikasi yang telah dilakukan sebelumnya. Proses diawali dari pengambilan data dari portal berita online lokal yang diambil dengan metode web carwling menggunakan Apache Nutch. Nutch lebih cocok dan stabil saat bekerja dengan Hadoop. dibandingkan dengan Scrapy yang lebih cocok untuk fokus crawling. Data berita hasil dari proses crawling disimpan dalam HDFS (Hadoop Distribution File System). Sebelum disimpan data terlebih dahulu di indekan untuk mempermudah dalam pencarian data menggunakan Apache Solr. Apache Solr dipilih karena menurut penelitian yang telah dilakukan Apache Solr lebih unggul dalam melakukan pencarian dan sangat mudah di integrasikan dengan HDFS. Setelah data tersimpan dalam HDFS, data di integrasikan dengan No-SQL database HBase untuk mempermudah melakukan query pada data yang tersimpan dalam HDFS.

Data kemudian dianalisis menggunakan aplikasi Spark. Aplikasi Spark yang digunakan menggunakan konsep master claster. Master claster merupakan proses dimanah job dikendalikan oleh Map Reducer, job nantinya akan disebarkan ke semua node manager sehingga pekerjaan akan lebih ringan saat mengelola data berukuran besar. Saat proses Analisa berjalan menggunakan Apache Spark data akan ditampung dalam Spark data frame. Data hasil analisa diberikan dua opsi pilihan penyimpanan dan penampilan data. Opsi pertama dimanah data hasil analisa yang berukuran besar disimpan dalam Aplikasi Hive yang merupakan gudang data (database) yang berada di atas cluster Hadoop. Kemudian data ditampilkan dalam bentuk grafik yang beragam yang di kumpulkan dalam dashboard menggunakan aplikasi Tableau Desktop. Opsi ini direkomendasikan bagi pengolahan big data untuk perusahaan besar, karena data visualisasi di simpan dalam server pribadi dan berkaitan dengan biaya berlangganan aplikasi. Opsi kedua menawarkan hasil analisa dari Apache Spark di simpat dalam HDFS (Hadoop Distribution File System) dalam bentuk file. File kemudian ditampilkan

dalam bentuk grafik yang beragam menggunakan aplikasi Tableau Public. Opsi ini di rekomendasikan bagi seorang data analis pemula atau bagi Penggemar BI yang ingin belajar lebih mengenai aplikasi Tableau Public. Aplikasi ini dapat menampilkan sampai satu juta baris data, open sorces dan menggunakan server Tableau.

5 Kesimpulan

Dari hasil pengujian dan analisa yang telah dilakukan selama pengerjaan tugas akhir ini dapat diambil beberapa kesimpulan sebagai berikut: (1) Scrapy cocok untuk fokus crawling, waktu crawling data sangat cepat namun tidak terukur dibandingkan Apache Nutch yang bekerja secara terukur dan bekerja dinamis melalui Hadoop. (2) Mongo DB memiliki waktu input data yang lebih cepat dengan HBase dan memiliki nilai Throughput yang lebih kecil dibandingkan namun nilai penggunaan memori dan CPU yang lebih besar dibandingkan dengan HBase Database. (3) Elasticsearch memiliki waktu input data yang lebih cepat di banding Apache Solr namun Solr lebih cepat dalam melakukan query data di bandingkan dengan Elasticsearch. Solr juga mudah diintegrasikan dengan HDFS. (4) Hive memiliki waktu input data lebih cepat dibandingkan dengan Apache Pig dan Spark, sedangkan Spark memiliki waktu eksekusi pemotongan data lebih cepat dibandingkan Pig dan Hive. (5) Metabase dapat berintegrasi dengan banyak sumber database, serta memiliki tampilan yang menarik. Kibana bisa digunakan jika hanya menggunakan search engine Elasticsearch. Apache Zepplin seperti notebook, dan tampilannya sederhana. Tableau yang memiliki fitur visualisasi yang beragam namun aplikasi tersebut kurang responsif dan berbayar. (6) Penelitian ini telah menghasilkan dua alternatif desain infrastruktur big data. Lengkap dengan menunjukkan kelemahan dan kelebihan dari setiap aplikasi yang digunakan, sehingga dapat mengoptimal desain Infrastruktur big data sesuai dengan kebutuhan penggunanya. (7) Infrastruktur yang disarankan tersebut sudah diimplementasikan pada node-node komputer di Lab Komputer Big Data Pens untuk mengolah big data dari media online terbukti telah berjalan dengan baik. Penelitian ini membuktikan penerapan analisis kinerja aplikasi big data dapat digunakan sebagai sarana optimasi desain infrastruktur big data. Namun penelitian ini hanya berfokus pada performa aplikasi saja belum melihat faktor – faktor lain yang menunjang optimasi desain infrastruktur big data. Pemilihan aplikasi dan variabel pengujian yang masih terbatas bila dibandingkan dengan banyaknya aplikasi big data yang ada. Infrastruktur yang dihasilkan harus disesuaikan dengan jumlah claster dan node yang digunakan pada penelitian. Perlu adanya penelitian lebih lanjut untuk mendapatkan hasil yang lebih baik pada pengujian aplikasi big data yang lain dengan parameter yang lebih variatif. Serta perlu dilakukan pengujian lebih lanjut pada infrastruktur Big Data yang dihasilkan dengan menggunakan node claster yang lebih banyak dan beragam sehingga dapat disesuaikan dengan kebutuhan dan kepentingan pengguna dalam menerapkan infrastruktur big data yang telah ada.

Referensi

[1] Gronwald, K.D., Integrated business information systems: A holistic view of the linked business process chain ERP-SCM-CRM-BI-Big Data, hal. 1–200,2017.

[2] Brown, K. *dkk.*, Accelerating Big Data Infrastructure and Applications (Ongoing Collaboration), *Proceedings - IEEE 37th International Conference on Distributed Computing Systems Workshops, ICDCSW 2017*, hal. 343–347, 2017.

[3] Chunpir, H.I., Rathmann, T. dan Zaina, L.M., An empirical evidence of barriers in a big data infrastructure, *Interacting with Computers*, vol. 30, no. 6, hal. 507–523, 2018.

[4] Venkatraman, R. dan Venkatraman, S., Big data infrastructure, data visualization and challenges, *ACM International Conference Proceeding Series*, hal. 13–17, 2019.

[5] TAMA, C.G.N., Sistem Operasi untuk Pemrosesan Big Data dengan berbasis Centos 7, 2017.

[6] Gorodov, E.Y.E. dan Gubarev, V.V.E., Analytical review of data visualization methods in application to big data, *Journal of Electrical and Computer Engineering*, vol. 2013, 2013.

[7] Series, C., Educational big data infrastructure : opportunities, design and challenges, 2021.

[8] Demchenko, Y., De Laat, C. dan Membrey, P., Defining architecture components of the Big Data Ecosystem, *2014 International Conference on Collaboration Technologies and Systems, CTS 2014*, no. March 2015, hal. 104–112, 2014.

[9] Sebei, H., Hadj Taieb, M.A. dan Ben Aouicha, M., Review of social media analytics process and Big Data pipeline, *Social Network Analysis and Mining*, vol. 8, no. 1, 2018.

[10] S.Widy, Teknologi Big Data Dengan Hadoop, [Daring]. Tersedia pada: https://medium.com/skyshidigital/teknologi-big-data-dengan-Hadoop-d8a2e93791a8 (Diakses tanggal 29 Maret, 2021).

[11] Rohman, M.S., Santoso, H.A. dan Saraswati, G.W., Pemanfaatan Topic-Focused Crawler untuk Pembangunan Corpus Berita Bencana menggunakan Teknik Scrapy CSS Selector, *Seminar Nasional APTIKOM (SEMNASTIK) 2019*, hal. 250–258, 2019.

[12] Suh, J., Vujin, V., Barac, D., Bogdanovic, Z. dan Radenkovic, B., Designing Cloud Infrastructure for Big Data in E-Government, *RUO. Revija za Univerzalno Odlicnost*, vol. 4, no. 1, hal. A26–A38, 2015.

[13] Hammood, A.H., A Comparison Of NoSQL Database Systems : A Study On MongoDB , Apache Hbase , And Apache Cassandra, no. October, hal. 20–23, 2016.

[14] Aydoğan, T., İlkuçar, M. dan AKCA, M.A., An Analysis on the Comparison of the Performance and Configuration Features of Big Data Tools Solr and Elasticsearch, *International Journal of Intelligent Systems and Applications in Engineering*, vol. 4, no. Special Issue-1, hal. 8–12, 2016.

[15] Sartika, E.P. dan Cahyono, A.B., Implementasi Elasticsearch Logstash Kibana Stack pada Sistem Portal Pengembangan dan Pembinaan Sumber Daya Manusia, vol. 1, no. 1, 2019.