Volume 4 Nomor 1 Januari 2022 JJEEE e-ISSN : 2715-0887
p-ISSN : 2654-7813

Jambura Journal of Electrical and Electronics Engineering

Prototipe Big data *Cluster* Berbasis Mikrokontroller Untuk Edukasi

Riska Kurniyanto Abdullah
Jurusan Matematika dan Teknologi Informasi
Institut Teknologi Kalimantan
Balikpapan, Indonesia
riska.abdullah@lecturer.itk.ac.id

Bowo Nugroho
Jurusan Matematika dan Teknologi Informasi
Institut Teknologi Kalimantan
Balikpapan, Indonesia
bowo.nugroho@lecturer.itk.ac.id

Ramadhan Paninggalih
Jurusan Matematika dan Teknologi Informasi
Institut Teknologi Kalimantan
Balikpapan, Indonesia
ramadhan.paninggalih@lecturer.itk.ac.id

Diterima : November 2021
Disetujui : Desember 2021
Dipublikasi : Januari 2022

***Abstrak*—Pembangunan *cluster* untuk *big data* tidaklah murah, butuh standar tertentu untuk menghasilkan *cluster* big data yang baik. Pada Penelitian ini dibuat suatu Prototipe *big data cluster* berbasis *mikrokontroller* yang bertujuan untuk merumuskan langkah-langkah penting dalam pembuatan *cluster big data* dan secara khusus. Hal ini dapat digunakan untuk keperluan edukasi untuk proses pembelajaran konsep – konsep dari *big data*. Sarana untuk belajar dan latihan implementasi konsep *big data* yang ingin kami tekankan ini yaitu sebagai *tools* (alat bantu) untuk mendukung dari perwujudan ekosistem pendidikan yang lebih baik khususnya untuk *mikrokontroller* dan *big data processing*. Metode yang digunakan pada penelitian ini yaitu dengan cara melaksanakan eksperimen prototipe untuk mendapatkan langkah yang tepat dalam membangun lingkungan Big data yang sesuai. Selain itu terlibat juga metode perancangan sistem untuk merancang Prototipenya terlebih dahulu. Tahapan dari penelitian yaitu di mulai dari proses studi literatur, kemudian proses rancangan, lalu evaluasi proses rancangan kemudian setelah itu dilakukan proses implementasi *hardware* dan *software*. Setelah selesai implementasi *hardware* dan *software* dua langkah terakhir yaitu proses analisis performa. Hasil dari penelitian menunjukkan performa *big data cluster* yang dibuat dari 5 perangkat *raspberry pi* dibandingkan dengan satu buah server cloud dengan *software big data* yang sama yaitu *hadoop* dan *spark.* Dari hasil tersebut performa yang didapat terdapat perbandingan yang jauh di mana jika di ketika *task* eksekusi di cloud lebih cepat hingga 8,5 kali di performa HDFS, sedangkan untuk waktu eksekusi CPU lebih singkat hingga 7 kali jika dibandingkan dengan *Protoype* ini. Hal ini menunjukkan untuk mendekati performa sesuai dengan standar lab *big data,* maka dibutuhkan lebih banyak lagi perangkat *raspberry pi* dan penyesuaian konfigurasi lainnya.**

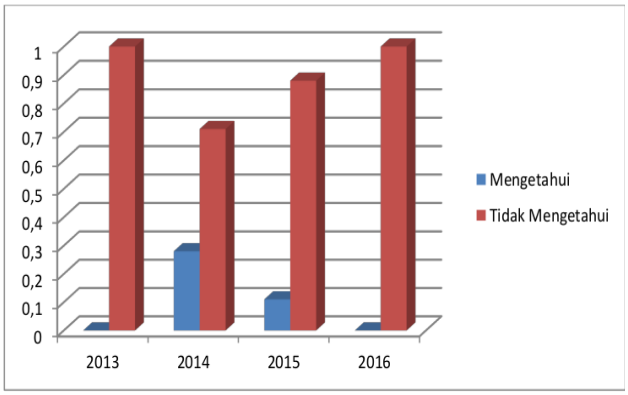
Kata Kunci—big data, raspberry pi, hadoop, spark.

***Abstract* — Cluster development for Big data is not cheap, it takes certain standards to produce a good Big data cluster. In this research, a microcontroller-based Big data cluster prototype was created which aims to formulate the important steps in making Big data clusters and specifically. This can be used for educational purposes for the process of learning the concepts of Big data. The means for learning and practicing the implementation of the Big data concept that we want to emphasize are as tools to support the realization of a better educational ecosystem, especially for microcontrollers and big data processing. The method used in this research is to carry out prototype experiments to get the right steps in building an appropriate Big data environment. In addition, the system design method is also involved in designing the prototype first. The stages of the research are starting from the literature study process, then the design process, then evaluating the design process then after that the hardware and software implementation process is carried out. After completing the hardware and software implementation, the last two steps are the performance analysis process. The results of the study show the performance of a big data cluster made of 5 raspberry pi devices compared to one cluod unit computing with the same big data software, namely hadoop and spark performance, that is, there is a very far comparison where when the task execution on the cloud system is faster up to 8.5 times in HDFS performance, while for CPU execution time up to 7 times shorter compared to this Protoype. This shows that to get closer to performance according to big data lab standards, more raspberry pi devices are needed.**

Term— big data, raspberry pi, hadoop, spark.

I. PENDAHULUAN

Teknologi IoT saat ini sudah sangat berkembang pesat bahkan sudah pada beberapa penerapan di pelayanan kesehatan lokal di Indonesia[1]. Pemrosesan data untuk menjadi informasi, kemudian menjadi pengetahuan dapat dilakukan dengan cepat. salah satunya cara dan metode yang tersedia yaitu dengan menggunakan konsep *Big data*. Pada konsep *Big data,* data yang diperoleh akan disimpan pada suatu media kemudian diproses dengan metode khusus untuk pengelolaan data yang sangat besar. Proses tersebut memungkinkan akses dan pemrosesan data dilakukan dalam rentang waktu yang masih dianggap wajar. Sering kali pendekatan pada metode transformasi data tradisional dengan data yang sudah sangat besar bisa terpengaruh dengan faktor *Big (O)*. Faktor *Big(O)* di mana waktu proses meningkat



Gambar 1. Pemahaman Mahasiswa Mengenai Big data

sumber: (Maula, 2016)

seiring dengan jumlah dan besar data yang jadi masukan dari sistem[2].

Pada Gambar 1 dapat dilihat yaitu data yang disajikan oleh[3] yang mana responden yang dimaksud adalah mahasiswa UPI (Universitas Pendidikan Indonesia) yang rutin ikut dalam berbagai pelatihan Big data. Data tersebut merepresentasikan antara tahun angkatan mahasiswa dan persentase dari pengetahuannya tentang *Big data*. Dari data tersebut kita bisa melihat bahwa untuk memahami konsep *Big data* sendiri tidaklah mudah, meskipun institusi yang sudah memiliki lab TIK saja hasilnya masih kurang maksimal. Artinya mahasiswa yang dimaksud memelajari *Big data* hanya dari teorinya saja, tanpa terlibat langsung cara mengimplementasikannya di perangkat yang memadai. Oleh karena itu sangat penting untuk institusi mulai menggarap Infrastruktur terkait bidang Big data ini. Sebuah lab yang khusus untuk menangani Pekerjaan Big data, bukan hanya Lab TIK saja.

Kesenjangan dari ketersediaannya infrastruktur *big data* menyebabkan kegiatan edukasi yaitu pelatihan dan pembelajaran tentang konsep *big data* menjadi masalah baru[4]. Sebelum mengimplementasikan berbagai macam *tools* untuk pengolahan *Big data*, hal yang pertama dilakukan adalah membuat *cluster* komputer. Kadang kala pengadaan *cluster* ini yang menjadi hambatan[5]. Tidak semua lembaga pendidikan mempunyai infrastruktur *cluster big data*. Sementara untuk menyewa di *server cloud* membutuhkan biaya yang tidak sedikit. Dengan adanya masalah ini maka tentunya diperlukan *low cost cluster* yang siap untuk diimplementasikan *tools big data.* Dengan adanya sarana tersebut maka keperluan edukasi dan latihan untuk penerapan konsep-konsep *Big data* bisa lebih mudah dan murah untuk dilakukan[6].

Informasi yang sangat besar perlu untuk diolah menjadi suatu pengetahuan yang berguna. Untuk mewujudkan hal tersebut maka data yang diperoleh perlu diproses pada suatu *cluster* yang bernama *big data cluster*. Masalahnya pembangunan *big data cluster* yang sesuai standar butuh biaya yang sangat mahal[7]. Karena hal tersebut Guru, Dosen, pendidik ataupun penyelenggara pendidikan kadang hanya menyampaikan teorinya saja tanpa melibatkan peserta didik terjun langsung dalam praktikum di *cluster big data* itu sendiri.

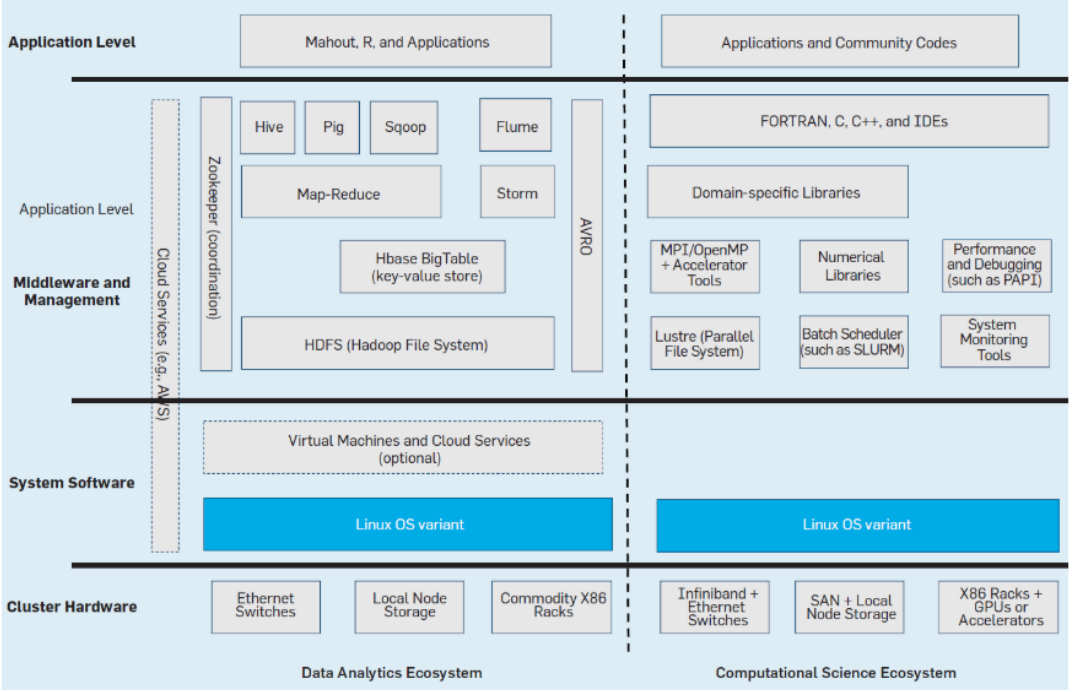
Seperti yang telah dilakukan oleh[8], yang mana salah satu cara mengatasi kesenjangan tersebut di atas, mereka menggunakan perangkat *Iridis-pi* untuk kepentingan edukasi dalam penerapan konsep – konsep *big data* bagi para mahasiswa. Namun sayangnya perangkat yang digunakan sudah tidak relevan lagi untuk saat ini.

Pada artikelnya menyebutkan *resource* komputasi untuk big data disarankan menggunakan teknologi *cloud computing*[9]. Latar belakang dari penelitiannya yaitu semakin berkembangnya teknologi big data dan cloud maka kedua teknologi ini seharusnya dapat disatukan menjadi sistem yang sepenuhnya utuh. Detail perangkat yang dimaksud diantaranya menggunakan teknologi virtual untuk mengakses dan mengelola *cpu* setelah itu membuat arsitektur khusus, sehingga *resource* tersebut dapat diakses dalam layanan *cloud*. Sementara itu penggunaan *storage* pun ditekankan menggunakan *storage* dengan tipe NAS (*Network Attached Storage*). NAS ini merupakan suatu metode yang mampu memisahkan antara sistem komputasi dan *storage*. Secara spesifik *storage* yang dimaksud dapat diakses dalam jaringan tertentu.

Dalam jangka panjang format penyatuan teknologi big data dan *cloud computing* ini akan memudahkan *maintenance*, namun di lain sisi ternyata membutuhkan *fixed cost* untuk memanfaatkan / menyewa *cloud* yang dimaksud agar tetap tersedia. Pada dasarnya artikel yang disampaikan mengenai kampanye penggunaan big data untuk UKM agar dapat melakukan eksplorasi dengan bisnisnya ke level yang lebih lanjut.

Melihat potensi data yang berkembang di Industri dan ingin mencoba mengolahnya dengan teknologi *hadoop*. Terdapat masalah dalam memanfaatkan data mentah secara langsung ke dalam *cluster* big data sehingga diperlukan metode khusus untuk menyelesaikannya[10]. Dalam paper-nya disebutkan metode baru diusulkan untuk menjaga agar performa *hadoop* tetap efisien. Hasil dari penelitiannya yaitu berhasil melakukan analisis big data dengan *core* dari *hadoop* namun hanya dilakukan pada satu *node* saja. Selain itu pemrograman paralel dilakukan untuk memaksimalkan kemampuan *processor* yang digunakan. *Dataset* yang digunakan tergolong standar untuk big data yaitu sebesar 1 TB. Sayang sekali pada penelitiannya tidak disertakan detail perangkat keras apa yang digunakan dan seperti apa spesifikasi yang digunakannya.

Kebutuhan industri yang mulai lumrah karena data yang dihasilkan sudah sangat banyak[11]. Dari hal itu maka perlu untuk mengorganisasikan dan melakukan analisis terhadap data tersebut, sehingga bisa memaksimalkan proses bisnis maupun penggunaan resource secara tidak langsung dapat diminimalisir. Masalah yang diangkat pada penelitian ini yaitu ketidakmampuan komputer produk *stock* (*general*) dalam melakukan komputasi skala besar baik berupa *storage*, kemampuan *processor* begitu pun dengan eksekusi program. Pada **Error! Reference source not found.**: Perbandingan ekosistem data analitik dengan komputasi. Secara sederhana merupakan perbandingan antara komputer dengan daya komputasi yang memanfaatkan *hadoop* dan map *reduce*

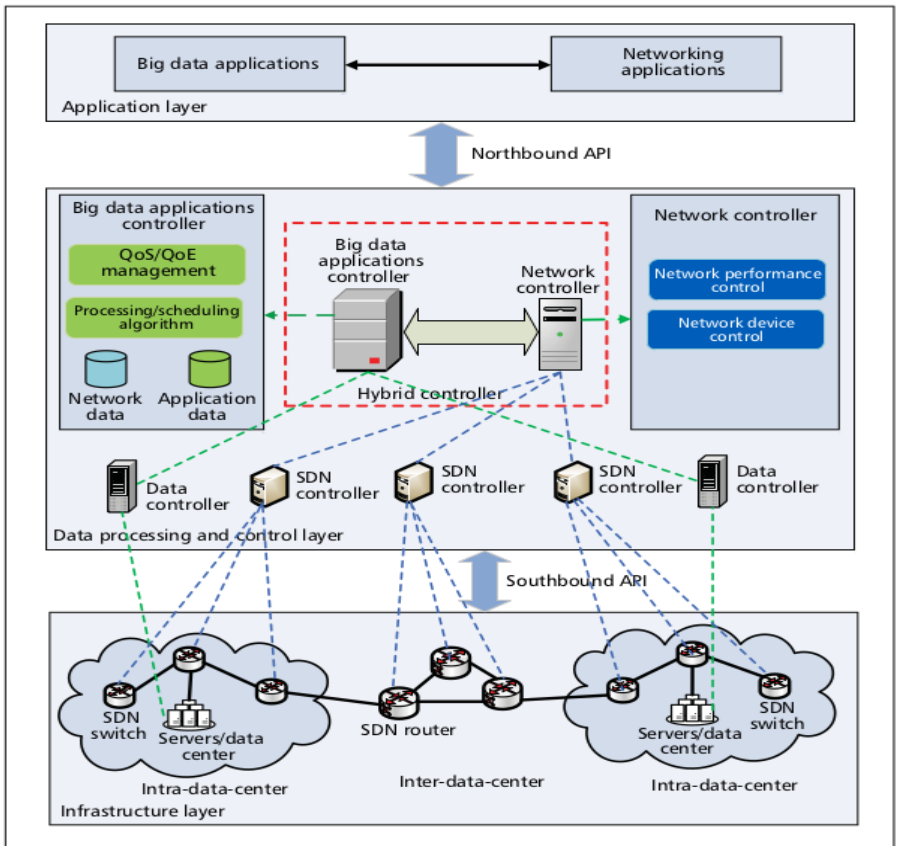


Gambar 2. Perbandingan ekosistem data analitik dengan komputasi. (Bramasto dan Sunarto,2016)

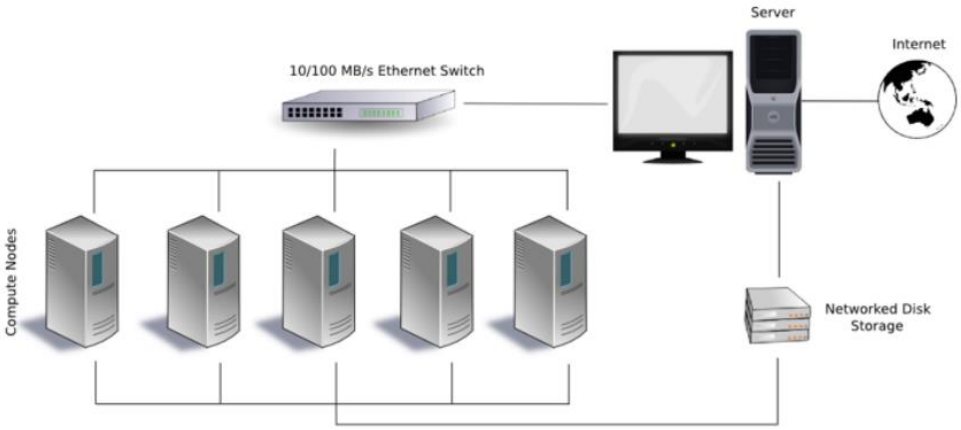
untuk analitik dan di sebelah kanan merupakan suatu perangkat komputer umum yang biasanya kita gunakan[11].

Pada artikel ini[11] menekankan perangkat keras yang digunakan harus ramah energi dengan tingkat kompleksitas yang komputasi yang mumpuni. Diantaranya terkait teknologi nano yang diterapkan pada berbagai macam *chip* komputer *saat* ini dan terus berkembang. Sementara itu pada perangkat lunak desain I/O *buffer* dan pemrograman paralel otomatis harus dimanfaatkan untuk mendapatkan hasil yang maksimal.

Keterkaitan dan hubungan antara *big data* dan SDN (*Software Define Networking*) menjadi topik yang dibahas pada penelitiannya[12]. Secara umum SDN dapat dimanfaatkan sebagai perangkat penunjang dari infrastruktur



Gambar 3. Intra dan inter data center network big data berbasis SDN (Cui dkk., 2016).



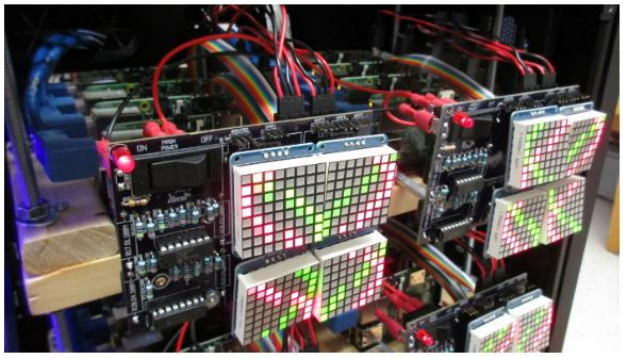
Gambar 4. Konfigurasi komputer cluster

untuk *cluster big data*. Pada **Error! Reference source not found.** dapat dilihat merupakan strategi penggunaan SDN untuk memaksimalkan akses dan *trafict* data antara *intra* dan *inter* pada data *center*. Untuk mewujudkan hal ini dibutuhkan perangkat keras, perangkat lunak berupa SDN dan operator yang berpengalaman. Secara teknis untuk arsitektur yang demikian dibutuhkan penanganan yang intensif atau juga bersifat berkala karena di bagian *controller* aplikasi tersebut merupakan *node* yang sangat sibuk karena mengharuskan beroperasi dan melayani hampir seluruh *request* dari sistem (semua *traffict* lewat *controller* ini). Meskipun demikian bukan hal yang mustahil hal ini bisa dilakukan. Perencanaan dan penggunaan alokasi sumber daya yang tepat sangat dibutuhkan untuk mewujudkan arsitektur yang ada pada sistem infrastruktur big data.

Pada penelitiannya[13] menerapkan perangkat *big data cluster* yang ramah energi. Perangkat tersebut diwujudkan berupa perangkat keras *micro* komputer *cluster* dari perangkat *raspberry pi* sejumlah 25 Buah. *Raspberry pi* yang digunakan yaitu model 2B yang dihubungkan dengan sistem jaringan 100 MB. *Cluster* ini dapat menghasilkan *floating point operation* sejumlah 15.4 GFLOPS dengan hanya menggunakan daya sebanyak 93 Watt saja. Namun memang jika dibandingkan dengan komputer *high-end* Intel x86 performa ini tetap saja ini masih jauh di bawahnya. Yang menarik yaitu bisa mengoperasikan banyak *node* layaknya *supercomputer* dengan daya yang sangat rendah. Hasil dari penelitian ini yaitu sebuah *cluster raspberry-pi* yang terdapat pada Gambar .

Pada penelitiannya[14] percobaan *streaming* data analisis menggunakan *apache spark*. Disebutkan dalam artikelnya *resource* perangkat keras yang digunakan yaitu 3 buah PC dengan RAM 4GB, HDD 500 GB, dan CPU Intel *Core i3*. Sementara perangkat lunak yang digunakan yaitu *apache spark* yang berjalan di atas infrastruktur *hadoop*.

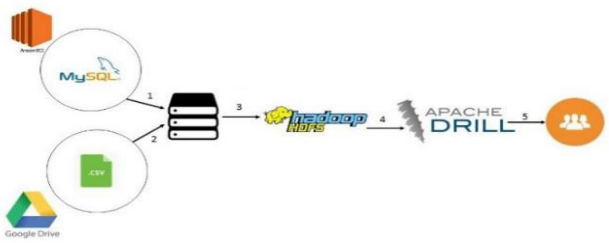
Ekstraksi pengetahuan dari sumber data yang sangat banyak baik jenis dan ragamnya disebut dengan big data. Dalam penelitian[15] penggunaan *cluster* dengan platform *MapReduce* diupayakan menggunakan komputasi paralel dengan metode SMOTE. Metode SMOTE merupakan salah satu metode *oversampling* yang populer di mana berdasarkan aturan *nearest neighbor rule*. SMOTE-GPU mengupayakan proses yang efisien dalam melakukan *handling dataset* bahkan pada beberapa juta *instance*. Metode ini sangat efektif meski berjalan pada perangkat komputer komoditas bahkan pada perangkat laptop sekalipun.



Gambar 5.. Raspberry-pi cluster (Cloutier dkk., 2016).

Dalam artikelnya[5], mereka mencoba membuat *cluster general* untuk mewujudkan *computer cluster* yang ramah energi. Sistem yang berhasil dibuat pada proyeknya yaitu sistem *cluster* dengan jumlah *node raspberry-pi* sebanyak lima buah dan perangkat penunjang lainnya. Pada Gambar 5 merupakan sistem yang diimplementasikannya, adapun model perangkat yang digunakan pada *projectnya* yaitu *raspberry-pi* model 3B. Pada perangkat lunak yang digunakan untuk analisis menggunakan program paralel dengan bantuan *library* MPI. Penggunaan *library* ini sangat minim karena para developer yang merasa familiar akibat kurang tersedianya tutorial dan referensi dari *library* ini. Mungkin saat ini teknologi terkait dengan hal tersebut yang bisa mengeksekusi baik yaitu sekelas teknologi yang digunakan pada *library python ray*. *Library ray* dapat mengoptimalkan *python* saat dieksekusi dalam sistem terdistribusi.

Terdapat kasus unik pada sebuah penelitian[16], di mana hal itdi implementasikan dengan sistem yang bernama *ecommerce* SIRCLO. Pada *ecommerce* tersebut terdapat data mentah yang akan dijadikan sebagai bahan eksperimen ekstraksi pengetahuan. Untuk mewujudkan hal tersebut mereka melakukan perancangan infrastruktur pemrosesan *big data* dengan menggunakan apache drill. Selain itu HDFS digunakan juga sebagai sistem *file* terdistribusi agar data yang di-ekstrak dan dijalankan secara paralel.

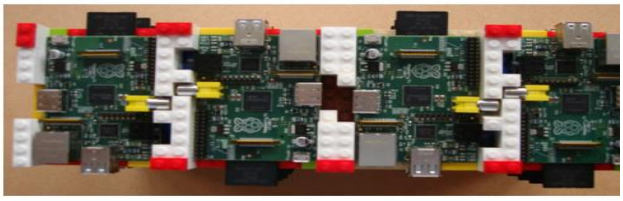


Gambar 2. Ilustrasi rancangan studi kasus SIRCLO

Dalam *paper*[17] dijelaskan pemrosesan data yang dilakukan pada *website* tidak perlu seutuhnya dilakukan pada satu proses *website* tersebut. Solusi yang ditawarkan oleh [17]yaitu melakukan pemrosesan data yang besar terpisah dengan program utama *website*. Sebenarnya dalam pengembangan aplikasi berbasis *website* dan rekayasa perangkat lunak metode ini disebut dengan nama lain yaitu *async request*. *Async request* membuat program utama *website* dapat berjalan terpisah dari thread sehingga jika ada proses yang memerlukan komputasi berat maka akan di skip. Setelah komputasi berat tersebut selesai barulah kemudian data yang ada di halaman *website* diperbaharui.

Paralel *programimng* dapat memudahkan pemrosesan data yang ada pada sistem. Pemrosesan yang biasanya di eksekusi sekuensial kini dapat dilakukan dengan mode *concurrent* (eksekusi pada waktu yang bersamaan). [18]Dalam penelitiannya yaitu membuat suatu *cluster* rendah energi dengan perangkat *raspberry-pi* 3 model A. *Cluster* dibangun dengan perangkat sebanyak 20 buah *raspberry-pi*. Selain itu perangkat lunak yang digunakan pada *cluster* yaitu menggunakan Open-MPI (*Message Parsing Interface*) sebagai *interface cluster*. Berbeda dengan implementasi di penelitian yang lain[18] sama sekali tidak melibatkan *hadoop* ataupun *spark*. Secara singkat bisa disimpulkan perangkat yang dibangun ini merupakan perangkat universal yang dibuat untuk tujuan edukasi. Bertujuan edukasi karena dengan perangkat kecil seperti itu sangat tidak mungkin untuk menghasilkan performa yang mendekati super komputer yang dibentuk dari komputer komoditas (perangkat komputer murah yang digunakan sehari-hari, bukan standar server). Penggunaan *cluster raspberry-pi* spesifik dengan MPI juga dilakukan oleh[19] yang juga sama tetap memanfaatkan perangkat *raspberry-pi* sebagai perangkat *cluster* yang rendah biaya. Perbedaan mendasar dari proyek lainnya[19] yaitu kali ini menambahkan VPN sebagai akses jaringan untuk memfasilitasi perangkat yang lokasinya terpisah jauh.

Indonesian Road Data Center Operation (IRODCO) merupakan suatu konsep yang disusun berdasarkan suatu masalah di mana ada beberapa operator sulit untuk mendapatkan informasi data jalan. Pada artikelnya[20], mengemukakan ide ini. Pada Gambar 6: Arsitektur IRODCO dapat dilihat dirancang untuk dapat melakukan proses analitik pada berbagai data yang tidak seragam dari berbagai instansi dalam hal ini Kementerian PU dan Pemerintah daerah, BUMN dan Swasta. Karena sifat data yang sangat beragam tersebut maka tentu ini diperlukan suatu metode khusus agar data yang diolah bisa menyesuaikan bentuk satu sama lain hingga memberikan informasi yang sesuai yang diinginkan.

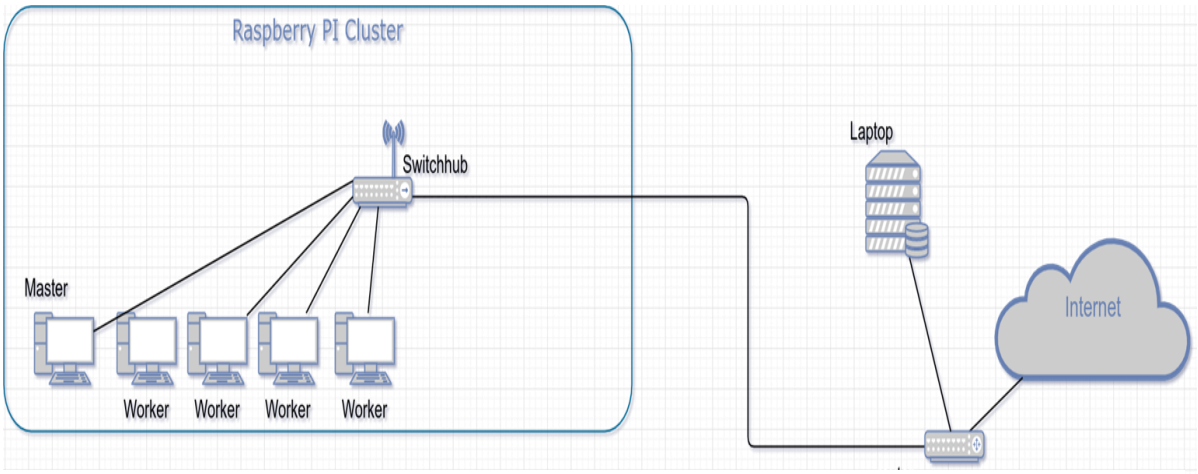


Gambar 3. Layout layer 4 Raspberry Pi Computer (Cox dkk., 2014)

Arsitektur Cluster Big data

Dalam penelitian sering kali ada pekerjaan untuk memroses *dataset* yang bisa saja untuk mengerjakannya membutuhkan resource yang besar. Untuk mengatasi hal tersebut dalam penelitiannya [21] membuat suatu *cluster* dari *raspberry-pi* yang dapat menunjang kegiatan penelitian mereka. Dalam artikelnya disebutkan performa dari *cluster* yang dibuat masih bisa memungkinkan untuk digunakan, mengingat dari pada harus membayar biaya lebih untuk menyewa *cloud*.

Data yang ada di internet terus berkembang, ibaratnya sedang tumbuh menjadi lebih besar dan masif secara eksponensial dari waktu ke waktu. Data tersebut atau sejenisnya memerlukan infrastruktur khusus untuk dianalisis sehingga bisa menjadi pengetahuan. Pada penelitiannya [22] yang berjudul *Data, Distribution, Deployment: Software*



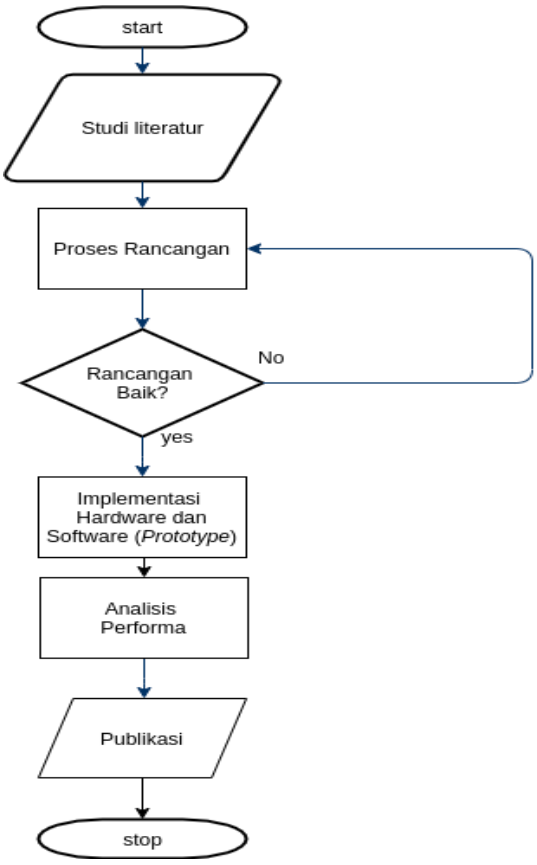
Gambar 8. Rancangan Arsitektur Sistem

Architecture Convergence in Big data Systems mengemukakan tentang konsep arsitektur *big data*. Dikemukakan bahwa antara *requirement* dan struktur dari rancangan harus saling berharmonisasi satu sama lain. Adapun konsep yang dimaksud dapat dilihat pada **Error! Reference source not found.** secara detail dijelaskan rancangan sistem *big data* ini dibuat menggunakan teknologi *database nosql* (*mongodb*). Keunggulan dari *mongodb* yaitu dapat melakukan baca tulis pada *storage* dengan latensi yang rendah. Selain itu *mongodb* juga mudah sekali dalam melakukan *scaling*, sehingga load dari sistem bisa dibagi pada beberapa *instance*.

Terdapat sebuah *project* yang dinamai iridis-pi. Iridis-pi[8] adalah sebuah *cluster* superkomputer yang dibuat dari 64 buah *raspberry pi* 2 model B. Tujuan dari *cluster* ini dibentuk yaitu untuk mendemonstrasikan *cluster* kecil dengan energi rendah. Sangat detail dalam laporan di artikelnya di mana *cluster* dibangun dengan biaya sebesar 3400 GBP, yang jika di konversikan ke nilai rupiah saat ini yaitu sebesar Rp. 68.654.768,-. Adapun *layout* dari *raspberry-pi* yang dibangun dapat dilihat pada Gambar 7.

II. METODE

Sebelum memulai melakukan formatting, pertama-tama tuliskan isi teks makalah anda pada file yang berbeda. Pastikan urutan paragaraf, sub judul atau heading dan persamaan telah berada pada urutan yang tepat.



Gambar 9. Metode Penelitian

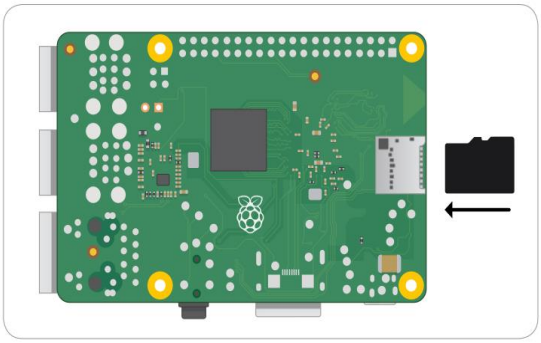
Pada tahap studi literatur indikator Pencapaian yaitu didapatkan signifikansi penelitian yang tercermin dari artikel jurnal yang ditemukan pada rentang kurang lebih 10 tahun sebelumnya. Setelah itu selanjutnya tahap proses rancangan yaitu merincikan alat, bahan, dan metode perancangan. Indikator pencapaian pada proses ini yaitu Gambar blok modul dari rancangan sistem.

Tahap berikutnya yaitu Implementasi desain yang dilakukan untuk membangun infrastruktur yang dimaksud. Indikator dari tahap ini yaitu *Cluster Big data* sudah bisa digunakan dengan terminal. Selanjutnya tahap akhir yaitu Analisis performa, tahap ini dilakukan untuk mengetahui seberapa baik dan seberapa pantas infrastruktur ini digunakan pada konsep-konsep *big data* dan studi kasus tertentu. Indikator tahap ini yaitu terciptanya grafik eksekusi *cluster* terhadap suatu metode *big data*.

III. HASIL DAN PEMBAHASAN

A. Rancangan

Seperti terlihat pada Gambar 9 desain sistem akan dimulai terlebih dahulu dari merakit perangkat, instalasi sistem operasi hingga konfigurasi dan percobaan sistem. Secara detail untuk tahap perakitan dan instalasi akan di bahas sebagai berikut.

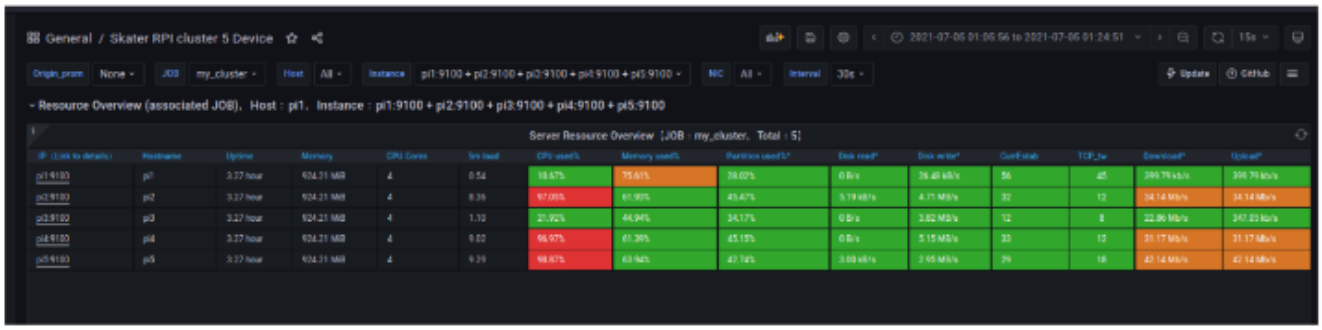


Gambar 10. Memasukkan microsd ke board raspberry pi 3

1) Perakitan Perangkat Keras

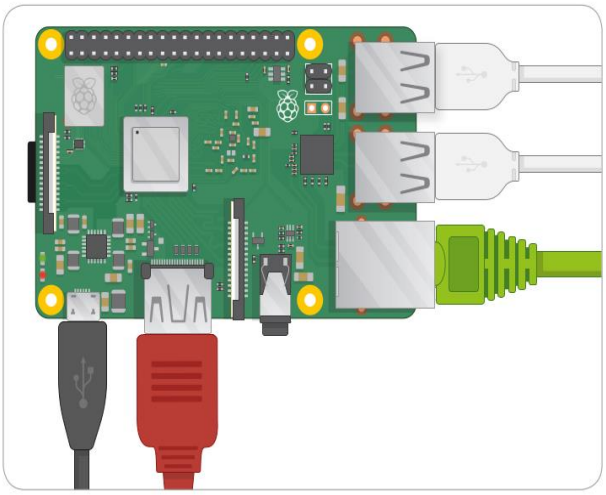
Langkah pertama bisa dilakukan dengna memasukkan microsd ke board raspberry pi 3 seperti pada ilustrasi 9. Pastikan bagian kuningan *micro sd card* harus mengarah ke atas sehingga *micro sd card* dapat di baca dengan benar oleh *board raspberry pi* 3. Setelah memasangkan *micro sd card*, maka langkah selanjutnya yaitu memasang *keyboard* USB dan *mouse* USB pada *interface* USB yang ada pada *board raspberry pi* 3. Sama seperti interface USB pada umumnya *board raspberry pi* 3 mendukung USB 2.0 dan USB 3.0.

Untuk keperluan *output* pada monitor maka *board raspberry pi* 3 perlu dihubungkan dengan kabel HDMI yang menuju ke monitor. Biasanya jika yang tersedia hanya monitor dengan *interface* VGA, maka diperlukan konverter HDMI to VGA agar monitor bisa dipergunakan sebagaimana mestinya. Kemungkinan besar penggunaan *port* HDMI hanya digunakan satu kali saja pada saat inisiasi konfigurasi / konfigurasi awal. Langkah selanjutnya yaitu memasangkan kabel jaringan LAN pada *port* yang disediakan pada *raspberry pi 3* kemudian langkah terakhir yaitu menyambungkan pada sumber tegangan dari adaptor sebagai sumber tegangan dengan *interface micro usb* di *board*.



Gambar 11. Load task pada HDFS dan SPARK System

Berikut pada gambar 12: *Board* lengkap seluruh *interface* terpasang merupakan ilustrasi ketika semua *interface* yang dibutuhkan untuk konfigurasi awal terpasang. Setelah ini bisa melanjutkan untuk menghidupkan *board* dengan mengalirkan listrik 5 V dari *port micro usb* yang tersedia.



Gambar 12. Board lengkap seluruh interface terpasang

2) Instalasi Sistem Operasi

Instalasi sistem operasi *raspberry pi* 3 dilakukan di atas sistem *linux*. Oleh karena itu sebelum melakukan instalasi program baru maka dianjurkan untuk melakukan *update*. Perangkat lunak yang dibutuhkan untuk *install* sistem operasi *raspberry pi* ke dalam *micro sd card* yaitu rpi-imager. Perangkat lunak ini dapat diinstall melalui snap. Cara lain selain menggunakan snap bisa juga dengan cara melakukan unduh *rpi-imager* dari *link github*. Setelah berkas deb berhasil diunduh lakukan lalu gunakan progam tersebut untuk membantu instalasi *os raspberry pi* ke *microsd* yaitu *rpi-imager*. Program ini yang akan digunakan untuk menginstall sistem operasi *raspberry pi* di *micro sc card*.

3) Desain Jaringan

Jaringan komputer yang digunakan yaitu jaringan komputer dengan kabel melalui *interface ethernet*. Pemilihan teknologi ini terkait performa yang harus dicapai oleh *cluster*. Tentunya sangat memungkinkan untuk menggunakan jaringan nirkabel namun akan sangat mempengaruhi *load data* pada setiap perangkat *raspberry pi* yang berkomunikasi di dalam *cluster* (Halfacree, 2018). Lakukan konfigurasi perangkat *rapsberry pi* dengan mengubah IP jaringan *ethernet*. Langkah tersebut ditunjukkan pada Gambar 1: Pilih *wireless & wired networks settings* dan Gambar 2: Atur IP untuk *ethernet* pada eth0. Alokasi yang diterapkan yaitu ip 192.168.1.151 – 192.168.1.155. Pada perangkat dengan IP 192.168.1.151 merupakan perangkat dengan peran master, sedangkan sisanya merupakan perangkat *worker*.

4) Cloning MicroSD

Cloning microsd card dibutuhkan sebagai opsi agar proses konfigurasi dari masing-masing perangkat raspberry pi dapat dilakukan lebih efisien. Proses modifikasi yang perlu dilakukan jika menggunakan hasil cloning dari micro sd card dari perangkat master yaitu hanya perlu melakukan perubahan hostname dan ip address. Proses cloning dapat dilakukan di sistem linux dengan perintah sebagai berikut.

```
sudo dd if=/dev/sdc | gzip -c >
/media/storage1/master/iso/big-data-cluster-1.1.img.gz
```

Sementara itu untuk menyalin image ke dalam microsd card yaitu dengan perintah berikut.

```
sudo gzip -dc /media/storage1/master/iso/big-data-cluster-
1.1.img.gz | sudo dd of=/dev/sdc bs=1M status=progress
```

5) Instalasi Hadoop dan Spark

Hadoop memerlukan java jdk sebagai syarat utamanya untuk berjalan di sistem terdistribusi. Oleh karena itu sangat penting untuk memastikan jika Java JDK sudah ter-install sebelumnya. Versi Java JDK yang dianjurkan yaitu Java JDK 8 (Watson, 2019). lakukan instalasi hadoop dengan command berikut.

```
sudo tar -xvf hadoop3.2.0.tar.gz -C /opt/
rm hadoop3.2.0.tar.gz && cd /opt
sudo mv hadoop-3.2.0 hadoop
sudo mv /opt/hadoop-3.2.0/ /opt/hadoop/
sudo chown pi:pi -R /opt/hadoop
```

Untuk instalasi spark dilalukan setelah diunduh, extract spark dengan perintah berikut seperti

```
$ sudo tar -xvf spark-2.4.3-bin-hadoop2.7.tgz -C /opt/
```

```
$ rm spark-2.4.3-bin-hadoop2.7.tgz && cd /opt
$ sudo mv spark-2.4.3-bin-hadoop2.7 spark
$ sudo chown pi:pi -R /opt/spark
```

B. Hasil dan Pembahasan

Proses ETL dilakukan dengan mengeksekusi berkas *python* di atas *spark*. Berikut *source code* dari program :

```
from pyspark import SparkConf, SparkContext
import collections
import time
start = time.time()
conf =
SparkConf().setMaster("yarn").setAppName("RatingsHistogramInRPI")
sc = SparkContext(conf = conf)
# turn off warn and info
sc.setLogLevel("ERROR")
lines = sc.textFile("/SparkCourse/ml-100k/u.data")
ratings = lines.map(lambda x: x.split()[2])
result = ratings.countByValue()
sortedResults = collections.OrderedDict(sorted(result.items()))
for key, value in sortedResults.items():
print("%s %i" % (key, value))
end = time.time()
print(f"total waktu yang dibutuhkan {end - start}")
```

hasil output dari program :

```
2021-06-30 22:07:40,142 INFO cluster.YarnClientSchedulerBackend: Add
WebUI
Filter. org.apache.hadoop.yarn.server.webproxy.amfilter.AmIpFilter,
Map(PROXY_HOSTS -> pi1, PROXY_URI_BASES ->
http://pi1:8088/proxy/application_1625061645646_0004),
/proxy/application_1625061645646_0004
1 6110
2 11370
3 27145
4 34174
5 21201
total waktu yang dibutuhkan 179.4764699935913
```

Hasil pengujian menunjukkan dari eksekusi TESTDFIO dan SPARKBENCH yang dimonitor dengan *grafana*. *Metric* terdapat pada bagian akhir dokumentasi pengujian, didapat sistem *cluster raspberry pi* ternyata hanya bisa melakukan proses data kurang dari 1GB. Selain itu akan mengalami *error Java Heap Memory*.

TABEL 1. SPESIFIKASI RPI CLUSTER

Spesifikasi Item	*Jumlah*
Core CPU	40
RAM	9 GB
HDD	116 GB
Network	100 Mbps

Pada tabel 1 spesidikasi raspberrry pi cluster yaitu Jumlah core CPU sebanak 40 core, dengan RAM 9 GB, hardisk untuk HDFS sebanyak 116 GB dan konfigurasi antar jaringan lokal sebesar 100 Mbps. Perlu diketahui saat melakukan benchmark pada network hasil yang kami peroleh network dari masing-masing node tidak dapat bekerja maksimal hingga 100 Mbps dikarenakan bottleneck dari desain raspberry pi sendiri. Hal ini sebenarnya sudah diatasi pada perangkat raspberry pi di genenrasi selanjutnya yaitu raspberry pi 4.

TABEL 2. SPESIFIKASI CLOUD SERVER

Spesifikasi Item	*Jumlah*
Core CPU	48
RAM	32 GB
HDD	1 TB
Network	10 Mbps

Pada tabel 2. Spesifikasi cloud server yang kami gunakan yaitu vps cloud dengan cpu core sejumlah 48 core, ram 32 GB, HDD sebesar 1TB, serta dengan jalur internet IX sebesar 10Mbps. Spesifikasi ini adalah spesifikasi maksimal yang ada jika melihat jumlah CPU yang digunakan dengan referensi pada penyedia cloud server yang kami gunakan.

TABEL 3. HASIL PENGUJIAN 1 TASK HDFS

Komponen Uji	*Satuan*	*Rpi Cluster*	*Cloud*	*Toleransi Error*
Waktu Eksekusi CPU	Detik	179,56	24,13	5%
Total Penggunaan RAM	GB	8.32	23,20	5%
Waktu bootstrap	Detik	93,2	3,01	5%
Waktu total eksekusi	Detik	272.76	27,14	5%

Pengukuran kami lakukan sebanyak lima kali percobaan dan hasil rata-rata pengukuran tersebut kami tuangkan pada tabel 3. Hasil pada tabel 3 dan selanjutnya pada tabel 4 disertakan toleransi error 5% sebagai ambang batas dari kemungkinan kesalahan pengukuran akibat perbedaan konfigurasi yang kami lakukan pada masing-masing perangkat. Dapat dilihat pada tabel 3 kurang lebih ada perbedaan sekitar 700 persen untuk waktu eksekusi CPU.

Sementara itu pada tabel 4 konfigurasi khusus menggunakan tambahan aplikasi spark yang mana terjadi kenaikan performa dari masing-masing sistem kurang lebih sebesar 75%. Hal ini menunjukkan penggunaan kombinasi antara HDFS dan Spark sangat direkomendasikan untuk mendapatkan performa yang maksimal.

TABEL 4. HASIL PENGUJIAN 1 TASK HDFS DAN SPARK

Komponen Uji	*Satuan*	*Rpi Cluster*	*Cloud*	*Toleransi Error*
Waktu Eksekusi CPU	Detik	132,26	18,23	5%
Jumlah Penggunaan Memory	GB	8,12	20,21	5%
Waktu bootstrap	Detik	92,33	2,50	5%
Waktu total eksekusi	Detik	224,59	20,73	5%

IV. KESIMPULAN

Prototipe *big data cluster* dengan perangkat raspberry pi dengan 1 perangkat master dan 4 worker belum cukup untuk menyaingi kemampuan *single node cloud server* yang

terpasang *hadoop* dan *spark*. Namun hal ini sebenarnya tidak terlalu signifikan pada penelitian ini karena dengan modul ini diharapkan mahasiswa dapat melatih kemampuannya untuk membuat *cluster big data* dari awal dengan tantangan dan kondisi yang mendekati dengan kondisi data center untuk sistem *big data* umumnya. Sementara itu untuk memperbaiki performa *cluster raspberry pi* agar mendekati performa sesuai dengan *cloud server*, maka dibutuhkan lebih banyak perangkat *worker* dari *raspberry pi* dengan kemungkinan sebanyak 28 *node worker*. 28 *node* tersebut diperoleh dari 7 dikali dengan 4 *worker existing*. Sesuai dengan hasil yang kami peroleh ada perbedaan sekitar kurang lebih 700 persen di antara cluster raspberry pi dalam penelitian ini dan *cloud server* yang kami gunakan.

UCAPAN TERIMA KASIH

Seluruh tim peneliti mengucapkan Terima kasih banyak untuk LPPM Institut Teknologi Kalimantan atas dukungan dana yang telah diberikan pada penelitian ini. Proses monitoring dan progress yang telah dilakukan sangat menunjang perkembangan penelitian ini.

Referensi

[1] G. Priyandoko, "Rancang Bangun Sistem Portable Monitoring Infus Berbasis Internet of Things," *Jambura J. Electr. Electron. Eng.*, vol. 3, no. 2, hal. 56–61, 2021, [Daring]. Tersedia pada: https://ejurnal.ung.ac.id/index.php/jjeee/article/view/10508/3092.

[2] I. Cholissodin dan E. Riyandani, *Analisis Big data (Teori & Aplikasi)*. Fakultas Ilmu Komputer - Universitas Brawijaya, 2016.

[3] R. N. Maula, "Penggunaan Big data Dalam Intansi Dibawah Naungan," in *Forum Keuangan dan Bisnis V, Th. 2016*, 2016, hal. 405–414, [Daring]. Tersedia pada: http://fkbi.akuntansi.upi.edu/wp-content/uploads/2017/10/FKBI-V_ITFC_03_Rita-Nikamatul-Maula_Universitas-Pendidikan-Indonesia.pdf.

[4] F. B. Universitas Gajah Mada, "Biotalks#8: 'Tantangan dan Peluang Penggunaan Big data dalam Pengembangan Riset Biologi,'" 2020. https://biologi.ugm.ac.id/2020/10/07/biotalks8-tantangan-dan-peluang-penggunaan-big-data-dalam-pengembangan-riset-biologi/ (diakses Apr 05, 2021).

[5] K. Doucet dan J. Zhang, "Learning cluster computing by creating a raspberry Pi cluster," *Proc. SouthEast Conf. ACMSE 2017*, hal. 191–194, 2017, doi: 10.1145/3077286.3077324.

[6] B. Williamson, "The hidden architecture of higher education: building a big data infrastructure for the 'smarter university,'" *Int. J. Educ. Technol. High. Educ.*, vol. 15, no. 1, hal. 1–26, 2018, doi: 10.1186/s41239-018-0094-1.

[7] A. Adrian, "Big data challenges," *Control Eng.*, vol. 4, no. 3, hal. 31–40, 2013, doi: 10.4172/2324-9307.1000133.

[8] S. J. Cox, J. T. Cox, R. P. Boardman, S. J. Johnston, M. Scott, dan N. S. O'Brien, "Iridis-pi: A low-cost, compact demonstration cluster," *Cluster Comput.*, vol. 17, no. 2, hal. 349–358, 2014, doi: 10.1007/s10586-013-0282-7.

[9] B. M. Purcell, "Big data using cloud computing," *J. Technol. Res.*, no. October, hal. 8, 2013.

[10] A. Saldhi, D. Yadav, D. Saksena, A. Goel, A. Saldhi, dan S. Indu, "Big data analysis using Hadoop cluster," in *2014 IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2014*, 2015, hal. 37–46, doi: 10.1109/ICCIC.2014.7238418.

[11] S. Bramasto dan S. Sunarto, "Big data dan Komputasi Skala Besar," *J. IPTEK*, vol. 1, no. 1, hal. 12–23, 2016, doi: 10.31543/jii.v1i1.92.

[12] L. Cui, F. R. Yu, dan Q. Yan, "When big data meets software-defined networking: SDN for big data and big data for SDN," *IEEE Netw.*, vol. 30, no. 1, hal. 58–65, 2016, doi: 10.1109/MNET.2016.7389832.

[13] M. F. Cloutier, C. Paradis, dan V. M. Weaver, "A raspberry Pi cluster instrumented for fine-grained power measurement," *Electron.*, vol. 5, no. 4, 2016, doi: 10.3390/electronics5040061.

[14] H. Sitepu, C. Z. Tumbel, dan M. Hutagalung, "Analisis Big data Berbasis Stream Processing Menggunakan Apache Spark," *J. Telemat.*, vol. 11, no. 1, hal. 6, 2017, [Daring]. Tersedia pada: http://journal.ithb.ac.id/telematika/article/view/145.

[15] P. D. Gutiérrez, M. Lastra, J. M. Benítez, dan F. Herrera, "SMOTE-GPU: Big data preprocessing on commodity hardware for imbalanced classification," *Prog. Artif. Intell.*, vol. 6, no. 4, hal. 347–354, 2017, doi: 10.1007/s13748-017-0128-2.

[16] Y. H. Partogi, A. Bhawiyuga, dan A. Basuki, "Rancang Bangun Infrastruktur Pemrosesan Big data Menggunakan Apache Drill (Studi Kasus : SIRCLO)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 3, hal. 951–957, 2018.

[17] H. Sulastri, A. Rahmatulloh, dan D. K. Hidayat, "Server-Side Processing Techniques for Optimizing the Speed of Presenting Big data," *J. Pilar Nusa Mandiri*, vol. 15, no. 1, hal. 47–52, 2019, doi: 10.33480/pilar.v15i1.62.

[18] K. Doucet dan J. Zhang, "The creation of a low-cost raspberry Pi cluster for teaching," *Proc. 24th West. Can. Conf. Comput. Educ. WCCCE 2019*, no. Figure 1, 2019, doi: 10.1145/3314994.3325088.

[19] D. V. Diwedi dan S. J. Sharma, "Development of a Low Cost Cluster Computer Using Raspberry Pi," in *Proceedings - 2018 IEEE Global Conference on Wireless Computing and Networking, GCWCN 2018*, 2019, hal. 11–15, doi: 10.1109/GCWCN.2018.8668647.

[20] D. S. Dewandaru, “Perancangan Big data Jalan Dan Jembatan,” *J. HPJI (Himpunan Pengemb. Jalan Indones.*, vol. 6, no. 2, hal. 83–92, 2020.

[21] P. A. Pankov, I. V Nikiforov, dan D. F. Drobintsev, “Hardware and software data processing system for research and scientific purposes based on Raspberry Pi 3 microcomputer,” 2021, vol. 32, no. 3, hal. 57–69, doi: 10.15514/ISPRAS.

[22] I. Gorton dan J. Klein, “Distribution, Data, Deployment: Software Architecture Convergence in Big data Systems,” *J. Chem. Inf. Model.*, vol. 53, no. 9, hal. 1689–1699, 2013.