

E-ISSN: 2460-8998

Jurnal Informatika Terpadu

Vol. 4 No. 2 Tahun 2018



Jurnal Informatika Terpadu

Jurnal Informatika Terpadu jurnal ilmiah yang berasal dari mahasiswa/i S-1 pada prodi Teknik Informatika dan Sistem Informasi di STT Terpadu Nurul Fikri sebagai salah satu syarat untuk meraih gelar Sarjana (S-1).

Ketua Penyunting (*Editor-in-chief*)

Sirojul Munir, S.Si., M.Kom.
Teknik Informatika
STT Terpadu Nurul Fikri

Dewan Penyunting (*Editorial Board Member*)

Hilmy Abidzar Tawakal, S.T., M.Kom.
Teknik Informatika
STT Terpadu Nurul Fikri

Tifanny Nabarian, S.Kom., M.T.I.
Teknik Informatika
STT Terpadu Nurul Fikri

Suhendi, S.T., M.MSI.
Sistem Informasi
STT Terpadu Nurul Fikri

Zaki Imaduddin, S.T., M.Kom.
Teknik Informatika
STT Terpadu Nurul Fikri

Ahmad Rio Ardiansyah, S.Si., M.Si.
Teknik Informatika
STT Terpadu Nurul Fikri

Mitra Bestari (*Reviewer*)

Drs. Rusmanto, M.M.
Sistem Informasi
STT Terpadu Nurul Fikri

Yekti Wirani, S.T., M.T.I.
Sistem Informasi
STT Terpadu Nurul Fikri

Henry Saptono, S.Si, M.Kom.
Teknik Informatika
STT Terpadu Nurul Fikri

Dr. Lukman Rosyidi, S.T., M.M., M.T.
Teknik Informatika
STT Terpadu Nurul Fikri

Amalia Rahmah, S.T., MT.
Sistem Informasi
STT Terpadu Nurul Fikri

Misna Asqia, M.Kom.
Sistem Informasi
STT Terpadu Nurul Fikri

Nurul Janah, S.IIP, M.Hum.
Sistem Informasi
STT Terpadu Nurul Fikri

Penyunting Pelaksana (*Assistant Editors*)

Muh Syaiful Romadhon, S.Kom.
Sistem Informasi
STT Terpadu Nurul Fikri

Fasyikhatun Maidah, S.Kom.
Sistem Informasi
STT Terpadu Nurul Fikri

Jurnal Informatika Terpadu (*Journal of Intregated Informathics*) telah terindeks oleh Google Scholar. Tanggungjawab isi artikel berada di penulis bukan pada penerbit atau editor.

Diterbitkan oleh:

LPPM STT Terpadu Nurul Fikri

Alamat Redaksi dan Distribusi:

Kampus B STT Terpadu Nurul Fikri lantai 3
Jl. Lenteng Agung Raya 20, Jakarta Selatan, DKI Jakarta, 12640
Telp. 021 – 786 3191

Email: lppm@nurulfikri.ac.id

Website: <https://journal.nurulfikri.ac.id/index.php/jit/> dan lppm.nurulfikri.ac.id

Daftar Isi

| | |
|---|-----------|
| Rancang Bangun Aplikasi <i>Mobile Client</i> CRM berbasis Android Studi Kasus CV Esindo Multi Tata | 34 |
| Riyan Wahyudi, Sirojul Munir | |
| Rancang Bangun Aplikasi <i>E-Commerce</i> menggunakan Framework Ruby on Rail Studi Kasus PT Hujan Rahmat Nusantara | 40 |
| Muhamad Ikhsan, Hilmy Abidzar Tawakal | |
| Penerapan Efektifitas Firewall pada <i>Software Defind Network</i> berbasis Openflow | 46 |
| Fathul Muiin, Henry Saptono | |
| Rancang Bangun Aplikasi Sistem Informasi Sekolah Go To School (GOS) berbasis Web menggunakan PHP | 58 |
| Chatur Nugroho, Suhendi | |
| Evaluasi Kinerja HDFS sebagai Infrastruktur Pembangunan Big Data | 63 |
| Yunita Surahman, Henry Saptono | |



RANCANG BANGUN APLIKASI MOBILE CLIENT CRM BERBASIS ANDROID STUDI KASUS CV ESINDO MULTI TATA

Riyan Wahyudi¹, Sirojul Munir²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
riaanwhy@gmail.com , rojulman@nurulfikri.ac.id

Abstract

CV Esindo Multi Tata is a company engaged in taxation services focusing on taxation, accounting, and related application development. In 2014, to support customer needs, CV Esindo Multi Tata built a Web-based Customer Relationship Management (CRM) Application System called Esi-CRM. Along with the development of business models and technology, as well as the needs of various users of the Esi-CRM application, several problems were found, namely the sales transaction process, invoicing, management of training products, and product ordering which was less efficient, mainly when accessed by users using a desktop computer. Alternatively, a laptop, the user wishes that the application can be appropriately accessed using a mobile smartphone device. This study aims to build a mobile-based Customer Relationship Management application at CV Esindo Multi Tata using a native android that accesses the MySQL database via a web service. The development method used in the application design process is waterfall and UML diagrams. Application functional testing uses black-box testing followed by the User Acceptance Test and user questionnaires. From this research, it was found that 100% of the functionality of the Esi-CRM mobile application was running well. The questionnaire results for the appropriateness of the application using the Likert measurement scale obtained a percentage of calibration of 93.75%, which means that the application is very suitable for use and can facilitate and streamline the user's work.

Keywords: Mobile Application, CRM, CV Esindo Multi Tata

Abstrak

CV Esindo Multi Tata adalah perusahaan yang bergerak di bidang jasa perpajakan yang berfokus pada bidang pelatihan perpajakan, akuntansi, dan pengembangan aplikasi terkait. Pada tahun 2014, untuk menunjang kebutuhan transaksi dengan pelanggan, CV Esindo Multi Tata membangun Sistem Aplikasi Customer Relationship Management (CRM) bernama Esi-Crm berbasis web. Seiring dengan perkembangan model bisnis dan teknologi, serta kebutuhan dari berbagai pihak pengguna aplikasi Esi-CRM, ditemukan beberapa permasalahan yaitu pada proses transaksi penjualan, pembuatan tagihan, pengelolaan produk pelatihan, dan pemesanan produk didapat kurang efisien, terutama ketika diakses oleh pengguna menggunakan komputer desktop atau laptop, pengguna berkeinginan aplikasi dapat di akses dengan baik menggunakan perangkat mobile *smartphone*. Penelitian ini bertujuan untuk membangun aplikasi Customer Relationship Management Berbasis mobile pada CV Esindo Multi Tata dengan menggunakan android native yang mengakses database mysql melalui web service. Metode pengembangan yang digunakan dalam proses perancangan aplikasi ini adalah dengan waterfall dan diagram UML. Pengujian fungsional aplikasi menggunakan pengujian *black box* yang dilanjutkan dengan User Acceptance Test dan kuesioner pengguna. Dari penelitian ini didapat 100% fungsionalitas aplikasi mobile Esi-CRM berjalan dengan baik, dan hasil kuesioner untuk kelayakan aplikasi menggunakan skala ukur likert didapat persentase kelayakan sebesar 93.75%, yang artinya aplikasi sangat layak digunakan serta dapat memudahkan dan mengefisiensikan pekerjaan pengguna.

Kata kunci: Aplikasi *mobile*, CRM, CV Esindo Multi Tata

1. PENDAHULUAN

Persaingan dalam dunia bisnis semakin cepat, yang mendorong masing-masing perusahaan untuk selalu berkembang dan beradaptasi dengan perubahan untuk dapat memenangkan persaingan. Perkembangan teknologi

informasi yang sangat pesat telah banyak merubah kegiatan bisnis perusahaan yaitu perubahan ke arah bisnis yang memanfaatkan teknologi internet. Penggunaan internet menurut survei dari APJII hampir 44,16 masyarakat indonesia menggunakan internet melalui *smartphone*

sedangkan pengguna internet melalui laptop 4,49 dan yang mengakses internet keduanya 39,28 seperti terlihat pada gambar berikut [1].

Pada level manajemen dalam perusahaan saat ini sangat populer penggunaan sistem manajemen untuk membina hubungan dengan pelanggan, di antaranya dengan menerapkan sistem *Customer Relationship Management* (CRM). Dengan adanya CRM dapat terjalin relasi yang lebih kuat di antara perusahaan dan pelanggan, dan pada gilirannya sedikit demi sedikit kepuasan pelanggan dapat tercapai. Tujuan utama dari program kepuasan pelanggan adalah untuk membangun hubungan dengan pelanggan sehingga mereka menjadi pelanggan setia perusahaan dalam jangka panjang [2].

CV Esindo Multi Tata merupakan perusahaan yang bergerak di bidang pelatihan perpajakan dan pengembangan sumber daya manusia dengan fokus pada bidang perpajakan, akuntansi, sumber daya manusia dan pengembangan *software* terkait [3]. Pada tahun 2014 CV Esindo Multi Tata dalam menunjang kebutuhan transaksi dan hubungan antara perusahaan dan pelanggan telah dibangun Sistem Aplikasi *Customer Relationship Management* (CRM) bernama Esi-Crm berbasis web, aplikasi Esi-CRM berbasis web ditemukan beberapa permasalahan yaitu pada proses melihat transaksi penjualan, pengelolaan produk pelatihan, dan pemesanan produk didapat kurang efisien.

Dari latar belakang tersebut penulis tertarik untuk mengembangkan aplikasi *Customer Relationship Management* (CRM) berbasis mobile menggunakan *platform android* dengan harapan dapat mengefisienkan pekerjaan pengguna aplikasi Esi-Crm khususnya pada divisi pemasaran.

1.1 Rumusan Masalah

Berdasarkan latar belakang yang penulis sampaikan dapat dirumuskan permasalahan sebagai berikut:

1. Apakah dengan membangun sistem aplikasi *client* berbasis *mobile* pada Esi-CRM dapat memudahkan dan mengefisienkan pekerjaan pada bagian pemasaran?
2. Apakah fitur-fitur aplikasi Esi-CRM berbasis *mobile* dapat berjalan baik pada platform android?

1.2 Tujuan dan Manfaat

Tujuan yang penulis harapkan dari penelitian ini yaitu:

1. Membangun aplikasi Esi-CRM berbasis *mobile*, untuk mengelola data transaksi pelanggan.
2. Memahami konsep *Customer Relationship Management* pada sebuah perusahaan.

Manfaat yang didapat dari penelitian ini antara lain:

1. Dapat mempermudah *user* pemasaran dalam pengelolaan sistem aplikasi Esi-CRM
2. Membantu Divisi *Sales and Marketing* CV Esindo Multi Tata dalam proses pengelolaan data pelanggan, penjualan, serta pembuatan *purchasing order*.

2. TINJAUAN PUSTAKA

2.1 CRM

Menurut Kotler dan Armstrong (2004), *Customer Relationship Management* (CRM) adalah proses membangun dan mempertahankan hubungan dengan konsumen yang menguntungkan dengan memberikan produk yang sangat bernilai bagi konsumen dan membuat konsumen puas [4]. Menurut Kotabe dan Helsen (2004), CRM adalah proses pengelolaan interaksi antara perusahaan dengan konsumennya secara strategis, dengan tujuan untuk memaksimalkan nilai konsumen bagi perusahaan dan memuaskan konsumen [5]. Dari kedua definisi di atas dapat disimpulkan bahwa CRM adalah proses pengelolaan konsumen untuk memaksimalkan nilai yang didapat dari pelanggan guna meningkatkan keberhasilan bisnis dan juga meningkatkan kepuasan pelanggan.

2.2 Android

Android merupakan sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak [6].

2.3 Web Service

Web Service merupakan sebuah mekanisme interaksi antar sistem yang menunjang interoperabilitas dengan tujuan untuk suatu kepentingan integrasi data yang dapat diakses melalui internet oleh banyak pihak dan media akses dapat memanfaatkan teknologi milik tiap-tiap pengguna. Selain itu, *web service* dapat diimplementasikan dengan menggunakan platform apapun dan dapat dibangun dengan menggunakan bahasa pemrograman apa saja. Kelebihan yang dimiliki oleh web service [7].

2.4 Waterfall

Menurut Ian Sommerville (2011, p30), metode *waterfall* memiliki tahapan utama dari *waterfall model* yang mencerminkan aktifitas pengembangan dasar [8]. Terdapat 5 (lima) tahapan pada metode *Waterfall*, yaitu:

1. *Requirement Analysis and Definition*
2. *System and Software Design*
3. *Implementation and Unit Testing*
4. *Integration and System Testing*
5. *Operation and Maintenance*

2.5 UML (*United Modelling Language*)

UML merupakan bahasa visual yang menjadi standar untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat [9].

Jenis-jenis diagram dalam UML yaitu:

1. Use Case Diagram
2. Class Diagram
3. Sequence Diagram
4. Collaboration Diagram
5. Activity Diagram
6. Component Diagram
7. Object Diagram

3. URAIAN PENELITIAN

Pada tahap ini berisi tentang tahapan-tahapan yang dilakukan penulis dalam penelitian diantaranya tahapan pengembangan sistem dengan metode *waterfall*. Rancangan penelitaian yang menjelaskan mengenai cara yang dilakukan penulis untuk menyelesaikan permasalahan, jenis penelitian yang dilakukan, rancangan analisis dan pengumpulan data, metode pengujian, tempat dan waktu dilakukan penelitiannya.

3.1 Analisis Sistem

Pada tahap ini dilakukan pengumpulan kebutuhan *user* di tingkat perangkat lunak yang dilakukan penulis pada CV Esindo Multi Tata. Dengan analisis ini, penulis akan mendapatkan data dan informasi yang diperlukan untuk membuat rancangan aplikasi. Hasil akhir dari tahapan ini adalah spesifikasi kebutuhan aplikasi Esi-CRM.

3.2 Desain Sistem

Pada tahap ini dilakukan pengumpulan kebutuhan *user* di tingkat perangkat lunak yang dilakukan penulis pada CV Esindo Multi Tata. Dengan analisis ini, penulis akan mendapatkan data dan informasi yang diperlukan untuk membuat rancangan aplikasi. Hasil akhir dari tahapan ini adalah spesifikasi kebutuhan aplikasi Esi-CRM.

3.3 Pengkodean Program

Pada tahap ini dilakukan implementasi dari tahap perancangan dan hasil rancangan kedalam baris kode program. Sehingga hasil dari tahap pengkodean program ini sesuai dengan desain yang telah dibuat di tahap desain yang di butuhkan dalam pengembangan aplikasi Esi-CRM pada CV ESINDO MULTI TATA.

3.4 Uji Coba Program

Pada tahap ini pengujian terlebih dahulu dilakukan pada setiap fungsi yang terdapat dalam Aplikasi. Jika setiap fungsi dan prosedur tersebut selesai diuji dan tidak terdapat masalah atau *error* dalam pembuatan Aplikasi Esi-CRM. Kemudian dilakukan pengujian di tingkat perangkat lunak

yang difokuskan pada pemeriksaan hasil pengujian dilakukan dengan menggunakan *black box testing*.

3.5 Pemeliharaan

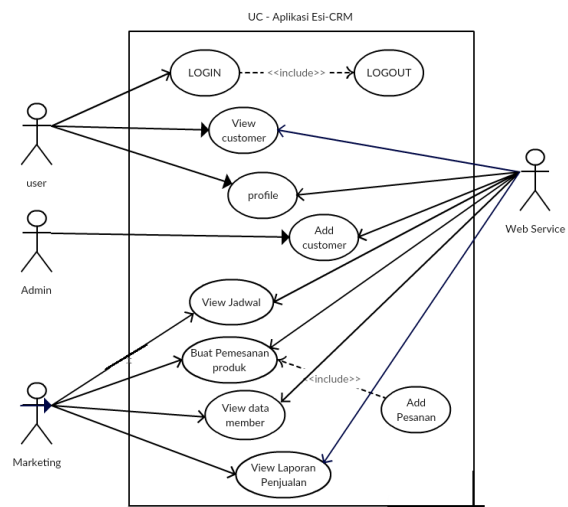
Pada tahap ini dilakukan pemeliharaan terhadap aplikasi Esi-CRM yang sudah dibuat, pemeliharaan yang dilakukan yaitu memperbaiki kesalahan yang tidak ditemukan pada tahap-tahap sebelumnya.

4. ANALISIS DAN PERANCANGAN

4.1 Analisis Sistem

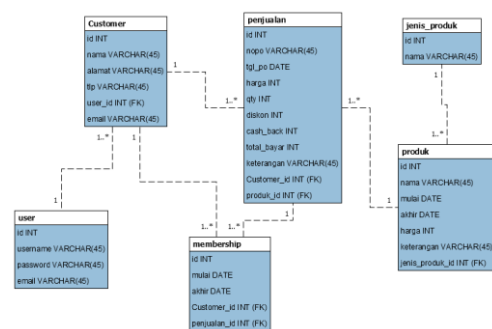
Pada tahapan ini dilakukan proses pengumpulan data (*requirement user*) dengan melakukan observasi untuk melihat bagaimana *user* menggunakan aplikasi CRM berbasis web, juga dilakukan wawancara dengan beberapa staff pada divisi *marketing* Esindo Multi Tata untuk mengetahui sejauh mana aplikasi yang sudah ada digunakan dan juga kekurangan dari aplikasi yang digunakan. Dari tahapan ini diharapkan didapatkan rancangan pengembangan aplikasi Esi-CRM berbasis *mobile*. Dari hasil analisis sitem yang berjalan diidentifikasi terdapat dua *user* yaitu *user admin* dan *user marketing*.

a. Use Case



Gambar 1. Use Case

b. Domain Model

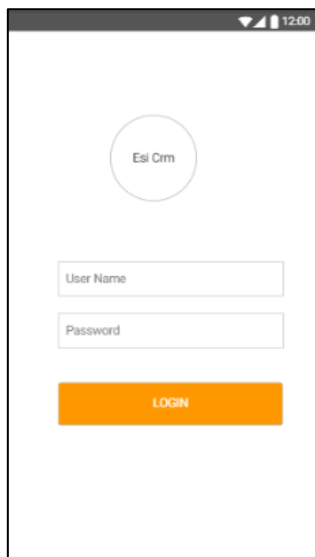


Gambar 2. Domain Model

4.2 Perancangan Sistem

Perancangan antarmuka merupakan tahap penelitian penulis gunakan untuk memberikan gambaran kasar mengenai tampilan aplikasi Esi-CRM yang akan dirancang menggunakan teknik *mock up*.

1. Tampilan *Login* digunakan sebagai tampilan awal aplikasi Esi-CRM, pada tampilan ini untuk user marketing ataupun admin. Jika user berhasil login maka akan masuk kehalaman *dashboard* sebagai halaman berikutnya, dan jika *user* gagal *login* maka akan kembali ke tampilan *login*.



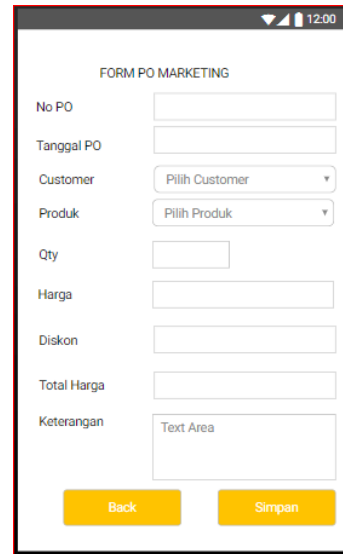
Gambar 3. Mockup Login

2. Tampilan *dashboard* adalah tampilan awal aplikasi Esi-Crm ketika user berhasil login, halaman ini *user* akan melihat data pribadi, list top penjualan *marketing* dan juga terdapat beberapa menu yaitu menu PO, menu *customer*, menu laporan, menu member, menu produk, menu penjualan.



Gambar 4. Dashboard

3. Tampilan menu PO adalah tampilan yang berisikan form-form untuk membuat pemesanan produk yang dilakukan oleh user marketing atau pun admin pada aplikasi Esi-CRM.



Gambar 5. Form PO

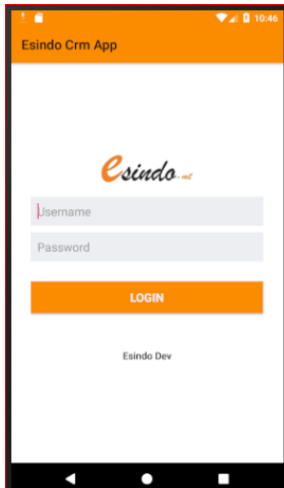
4. Tampilan menu *customer* adalah tampilan yang berisikan data-data *customer* pada menu *customer* user dapat melakukan pencarian data dengan memfilter berdasarkan nama *customer*.



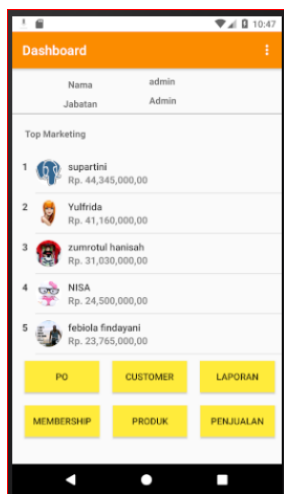
Gambar 6. Menu Customer

5. IMPLEMENTASI

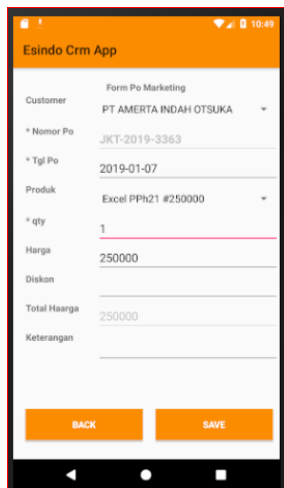
Implementasi dari sistem yang dibuat berdasarkan analisis kebutuhan user dan merupakan kelanjutan dari tahap perancangan sistem agar dapat dioperasikan. Adapun tujuan implementasi sistem yang dirancang adalah sebagai berikut:



Gambar 7. Halaman Login



Gambar 8. Halaman Dashboard



Gambar 9. Menu Form PO

Hasil kuisisioner aplikasi Esi-CRM dapat di lihat pada tabel 1:

Tabel 1. Hasil Kuisisioner Aplikasi

| No | Pertanyaan | Nilai | | | | |
|-----------------|---|-------|-----|----|----|-----|
| | | SS | S | CS | TS | STS |
| 1 | Apakah Aplikasi Esi CRM dapat terinstall dan berjalan di smartphone anda | 12 | 2 | 0 | 0 | 0 |
| 2 | Apakah Aplikasi Esi CRM dapat memudahkan anda mencari data pelanggan | 11 | 3 | 0 | 0 | 0 |
| 3 | Apakah Aplikasi Esi CRM dapat memudahkan anda membuat PO | 12 | 2 | 0 | 0 | 0 |
| 4 | Apakah penggunaan menu dan fitur mudah di gunakan | 8 | 6 | 0 | 0 | 0 |
| 5 | Apakah aplikasi Esi Crm dapat memudahkan anda melihat laporan penjualan | 12 | 2 | 0 | 0 | 0 |
| 6 | Apakah dengan menggunakan aplikasi Esi crm dapat menghemat waktu pengerjaan jika dibandingkn dengan CRM web | 6 | 8 | 0 | 0 | 0 |
| 7 | Apakah Fitur yang di gunakan sudah sesuai dengan kebutuhan | 6 | 8 | 0 | 0 | 0 |
| 8 | Apakah Aplikasi Dapat Bermanfaat bagi pengguna | 10 | 4 | 0 | 0 | 0 |
| JUMLAH | | 77 | 35 | 0 | 0 | 0 |
| JUMLAH SKOR | | 385 | 140 | 0 | 0 | 0 |
| ΣSKOR OBSERVASI | | 525 | | | | |
| PERSENTASE | | 93.75 | | | | |

6. KESIMPULAN

6.1 Simpulan

- Setelah dilakukan pengujian aplikasi dan kuisisioner tentang kemudahan dan efisiensi penggunaan didapat tingkat kelayakan 93.75% , yang menyatakan bahwa aplikasi Esi-CRM dapat memudahkan dan mengefisienkan pekerjaan pada bagian *marketing*.
- Fitur aplikasi Esi-CRM berbasis mobile dapat berjalan baik pada platform *android* ditunjukan dari hasil pengujian fungsional aplikasi menggunakan pengujian *black box* dan pengujian UAT dinyatakan 100% aplikasi berjalan dengan baik.

6.2 Saran

Pada penelitian yang dilakukan ini penulis hanya berfokus pada fungsional aplikasi terhadap divisi *marketing*. Dari hasil penelitian ini penulis menyarankan untuk pengembangan selanjutnya agar dapat mengembangkan

aplikasi aplikasi CRM *mobile* pada modul-modul yang lainnya.

DAFTAR PUSTAKA

- [1] Asosiasi Penyelenggara Jasa Internet Indonesia, "Profil Pengguna Internet 2017," 2017.
- [2] R. E. Kalalo, "Customer Relationship Management dan Kualitas Pelayanan Pengaruhnya terhadap Loyalitas Konsumen PT Matahari Dept. Store, Manado," Jurnal EMBA, Vol.1, No. 4, pp. 1553-1561, 2013.
- [3] "Profil CV Esindo Multi Tata," Esindo Multi Tata Web, <http://esindo.net/tentang-esindo/>, 2015.
- [4] Kotler & Armstrong, "Prinsip-Prinsip Marketing," edisi ketujuh. Jakarta: Salemba Empat, 2004.
- [5] M. Kotabe & K. Helsen, "Global Marketing Management," Third Edition, Wiley International Edition, 2004.

- [6] Arifianto & Teguh, "Membuat Interface Aplikasi Android Lebih Keren Dengan LWIT," Yogyakarta: Andi Offset, 2011.
- [7] E. Sutanta and K. Mustofa, "Kebutuhan *Web Service* untuk Sinkronisasi Data Antar Sistem Informasi dalam *E-Gov* di Pemkab Bantul Yogyakarta," Tek. Inform. - STMIK Bandung, 2012.
- [8] I. S. Mathiassen et al, "Metode *Waterfall*," p. 13, 2000.
- [9] Y. Sugiarti, "Analisis dan Perancangan UML (*Unified Modeling Language*)," Generated VB. 6, Yogyakarta: Graha Ilmu, 2013.
- [10] Arikunto, Suharsimi, "Dasar-dasar Evaluasi Pendidikan," Jakarta: Bumi, 2009.
- [11] Dyche, J. "*The CRM Handbook: A Business Guide to Customer Relationship Management*," Addison-Wesley, 2002.
- [12] R. Kalakota & M. Robinson, "*E-Business 2.0: Roadmap for Success*," 2001.
- [13] Nugroho, B., "*Database Relasional dengan MySQL*," Yogyakarta: Andi, 2004.
- [14] Payne, A, "*Handbook of CRM*. Elsevier Butterworth-Heinemann," 2005.
- [15] Pressman, Roger S., "Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku 1)," Edisi 2, Andi : Yogyakarta, 2002.



RANCANG BANGUN APLIKASI E-COMMERCE MENGGUNAKAN FRAMEWORK RUBY ON RAIL STUDI KASUS PT HUJAN RAHMAT NUSANTARA

Muhamad Ikhsan¹, Hilmy Abidzar Tawakal²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
mail.ichsan24@gmail.com, hilmi.tawakal@gmail.com

Abstract

PT Hujan Rahmat Nusantara is the only distributor company for AHASS Honda products, namely car wipers, and sells Harley Davidson products. PT Hujan Rahmat Nusantara has many resellers and is constantly increasing every month. So far, the existing management at PT Hujan Rahmat Nusantara has been managed manually using ledgers, Microsoft Word, and Microsoft Excel applications. Where a manual system in managing transaction data can result in data integrity and security that is not guaranteed as well as delays in information so that the time problem and the level of data validity are simplified to a minimum. This study aims to build a Web E-Commerce using Ruby programming involving the Ruby On Rails framework and PostgreSQL databases. The results of this study apply several features that can help sales, product data collection, serve client requests to manage resellers so that they can be easily managed by the management of PT Hujan Rahmat Nusantara.

Keywords: PT Hujan Rahmat Nusantara, Ruby, Ruby On Rails, PostgreSQL, E-Commerce

Abstrak

PT Hujan Rahmat Nusantara adalah perusahaan distributor tunggal produk AHASS Honda yakni Wiper mobil dan menjual produk-produk Harley Davidson. PT Hujan Rahmat Nusantara memiliki banyak *reseller* dan selalu bertambah tiap bulannya. Selama ini manajemen yang ada di PT Hujan Rahmat Nusantara dikelola secara manual menggunakan buku besar, aplikasi *Microsoft Word* dan *Microsoft Excel*. Dimana dengan sistem manual dalam pengelolaan data transaksi dapat mengakibatkan integritas dan keamanan data yang tidak terjamin serta keterlambatan akan informasi sehingga permasalahan waktu dan tingkat validitas data disederhanakan seminimal mungkin. Penelitian ini bertujuan untuk membangun *Web E-Commerce* menggunakan pemrograman Ruby yang melibatkan *framework Ruby On Rails* dan database *PostgreSQL*. Hasil penelitian ini menerapkan beberapa fitur yang dapat membantu penjualan, pendataan produk, melayani *request client* sampai mengatur *reseller* agar dapat mudah dikelola oleh manajemen PT Hujan Rahmat Nusantara.

Kata kunci: PT Hujan Rahmat Nusantara, Ruby, Ruby On Rails, PostgreSQL, E-Commerce

1. PENDAHULUAN

Dalam era globalisasi seperti saat ini dimana negara-negara di seluruh dunia menjadi satu kekuatan pasar yang semakin terintegrasi tanpa rintangan batas teritorial negara, mengakibatkan persaingan bisnis yang sangat ketat. Ditambah lagi dengan cepatnya arus informasi yang berkembang seiring dengan berkembangnya teknologi yang menuntut setiap perusahaan maupun individu untuk teru bergerak mengikutinya. Oleh karenanya perusahaan yang dapat menghadapi ketatnya persaingan dituntut untuk selalu berkembang dengan melakukan pengambilan keputusan secara cepat, tepat dan efisien.

E-Commerce merupakan salah satu cara untuk melakukan transaksi dan publikasi perniagaan secara online yang melibatkan internet agar dapat diakses oleh siapapun, kapanpun dan dimanapun. *E-Commerce* sebagai suatu bentuk kemajuan teknologi informasi telah membawa sejumlah perubahan, diantaranya menurunkan biaya interaksi antara pembeli dan penjual, interaksi menjadi lebih mudah tanpa batasan waktu dan tempat, lebih banyak alternatif dan mempermudah promosi, peluang memperluas pangsa pasar tanpa harus memiliki modal besar, peningkatan transparansi dan pelayanan kepada konsumen. *E-Commerce* secara strategis dapat berperan sebagai diferensiator yang dapat membentuk daya saing perusahaan

melalui sejumlah keunikan baik produk maupun sistem pelayanannya [1].

Pemrograman Ruby adalah sebuah bahasa pemrograman yang dikenal mudah digunakan, dikarenakan Ruby memiliki struktur bahasa yang mudah dimengerti. Disamping itu Ruby memiliki Framework berbasis web yakni *Ruby On Rails* yang dikenal pelopor MVC (*Model View Controller*) berbasis Web.

PT Hujan Rahmat Nusantara adalah perusahaan yang bergerak dalam bidang jual beli wiper mobil dan menjadi distributor tunggal dari AHASS di Indonesia berdiri pada tahun 2018.

Untuk meningkatkan usahannya, PT Hujan Rahmat Nusantara dalam hal ini mulai mencari terobosan baru dengan memanfaatkan *E-commerce*, sebagai salah satu alternatif media untuk meningkatkan omzet penjualan. Bertujuan untuk menyebarkan promosi, menampilkan katalog produk dan melakukan penjualan secara praktis. Selain itu, perusahaan juga harus dapat menjaga hubungan baik dengan pelanggan maupun antar sesama penjual. Pelanggan ingin terus mendapatkan informasi yang terbaru dari penjual, dan sesama penjual juga ingin saling berhubungan agar dapat saling bertukar informasi, serta membentuk komunitas, demi meningkatkan pelanggan dan tingkat penjualan mereka.

Atas dasar permasalahan yang peneliti paparkan di atas maka dibutuhkan sebuah website e-commerce berbasis web menggunakan *Ruby On Rails* yang akan memudahkan PT. Hujan Rahmat Nusantara dalam mengembangkan bisnisnya.

1.1 Rumusan Masalah

Dari uraian latar belakang di atas, maka peneliti merumuskan masalah sebagai berikut:

1. Apakah website *e-commerce* dapat membantu PT. Hujan Rahmat Nusantara dalam mengembangkan bisnisnya?
2. Apakah website *e-commerce* dapat memudahkan PT. Hujan Rahmat Nusantara untuk berbagi informasi dan berkomunikasi dengan pelanggan dan penjual lain?

1.2 Tujuan

Membangun website *e-commerce* berbasis Ruby pada PT. Hujan Rahmat Nusantara.

1.3 Manfaat

1. Mempermudah dalam para pelanggan melakukan transaksi jual-beli dan menjangkau lebih banyak pelanggan dari berbagai lini.
2. Mempermudah PT. Hujan Rahmat Nusantara untuk berbagi informasi dan berkomunikasi dengan pelanggan dan penjual lain.

1.4 Ruang Lingkup dan Batasan

Penelitian ini akan dilakukan di kantor PT. HUJAN RAHMAT NUSANTARA yang beralamat di City Market Pondok Gede Blok. C5 No.7 Pondok Cabe, Tangerang Selatan.

2. TINJAUAN PUSTAKA

2.1 PT Hujan Rahmat Nusantara

Berdiri pada tahun 2018 yang menjadi distributor tunggal di Indonesia yaitu Silicon Wiper untuk mobil dari Honda AHASS dan PT Hujan Rahmat adalah tempat dimana penulis meneliti kebutuhan teknis terhadap infrastruktur teknologi informasinya. Saat ini pengelolaan penjualan dan pendataan di PT Hujan Rahmat masih menggunakan cara konvensional. Oleh karena itu penulis berinisiatif untuk mengembangkan infrastruktur dan teknologinya yang salah satunya adalah membangun *E-Commerce* menggunakan pemrograman Ruby yang ber-framework *Ruby On Rails*.

2.2 Ruby

Berdiri pada tahun 2018 yang menjadi distributor tunggal di Indonesia yaitu Silicon Wiper untuk mobil dari Honda AHASS dan PT Hujan Rahmat adalah tempat dimana penulis meneliti kebutuhan teknis terhadap infrastruktur teknologi informasinya. Saat ini pengelolaan penjualan dan pendataan di PT Hujan Rahmat masih menggunakan cara konvensional. Oleh karena itu penulis berinisiatif untuk mengembangkan infrastruktur dan teknologinya yang salah satunya adalah membangun *E-Commerce* menggunakan pemrograman Ruby yang ber-framework *Ruby On Rails*.

2.3 E-Commerce

E-Commerce merupakan salah satu cara untuk melakukan transaksi dan publikasi perniagaan secara online yang melibatkan internet agar dapat diakses oleh siapapun, kapanpun dan dimanapun. E-Commerce sebagai suatu bentuk kemajuan teknologi informasi telah membawa sejumlah perubahan, diantaranya menurunkan biaya interaksi antara pembeli dan penjual, interaksi menjadi lebih mudah tanpa batasan waktu dan tempat, lebih banyak alternatif dan mempermudah promosi, peluang memperluas pangsa pasar tanpa harus memiliki modal besar, peningkatan transparansi dan pelayanan kepada konsumen. E-Commerce secara strategis dapat berperan sebagai diferensiator yang dapat membentuk daya saing perusahaan melalui sejumlah keunikan baik produk maupun sistem pelayanannya [1].

2.4 Metode Agile Software Development

Di dalam Agile Manifesto disebutkan bahwa untuk mengembangkan software dengan cara yang lebih baik ada 4 manifesto utama antara lain:

1. Mementingkan komunikasi/interaksi antar individu dibandingkan proses dan *tools*.
2. Mementingkan mengerjakan *software* dibandingkan dokumentasi yang lengkap.
3. Kolaborasi dengan para pengguna *software*. Para pengguna *software* dapat memberitahu seperti apa *software* yang mereka inginkan.
4. Respon terhadap perubahan. Dalam pengembangan *software* perubahan adalah sebuah realitas yang tidak bisa dihindari. Sehingga rencana proyek yang dibuat harus fleksibel dan memungkinkan perubahan.

3. URAIAN PENELITIAN

3.1 Studi Pendahuluan

Pada tahapan ini penulis melakukan dua jenis studi untuk mendapatkan semua data yang dibutuhkan dan terkait dengan penelitian yang dilakukan, dua jenis studi ini adalah Studi Literatur dan Studi Lapangan.

3.2 Analisis dan Perancangan

1. Analisis

Pada tahapan ini penulis melakukan pengumpulan data yang diperlukan untuk analisis kebutuhan, adapun teknik pengumpulan data yang dilakukan dalam penelitian ini adalah Wawancara dan Kuesioner.

2. Perancangan

Pada tahapan ini penulis melakukan analisis perancangan sistem yang akan diterapkan sesuai studi lapangan dan analisis yang telah dilakukan sebelumnya.

3.3 Implementasi

Setelah *software* dan teknologi pendukung disiapkan, tahapan dilanjutkan dengan instalasi *Ruby dan Ruby On Rails* lalu dilanjutkan dengan proses konfigurasi lalu *software* akan dilengkapi dengan data penjualan PT Hujan Rahmat Nusantara.

3.4 Uji Coba

Proses uji coba yang akan dilakukan dalam penelitian ini adalah dengan menggunakan metode *Black Box Testing*. *Black Box Testing* merupakan metode pengujian yang difokuskan pada spesifikasi fungsionalitas dari setiap kerangka aplikasi. Testing ini bertujuan untuk memastikan fungsionalitas setiap modul dalam Pemrograman *Ruby* yang diterapkan telah berjalan dengan baik dan sesuai dengan yang diharapkan.

3.5 Penarikan Kesimpulan dan Rekomendasi

Setelah bagian PT Hujan Rahmat Nusantara menggunakan aplikasi *E-Commerce* untuk mengimplementasikan penjualan berbasis website. Penulis akan melakukan analisis kesimpulan dan memberikan rekomendasi cara

kerja yang lebih baik dalam implementasi *Customer* PT Hujan Rahmat Nusantara.

4. ANALISIS DAN PERANCANGAN

A. User Requirement

Berdasarkan hasil wawancara dan observasi dapat disimpulkan beberapa kebutuhan *user* terhadap aplikasi *website e-commerce* pada PT Hujan Rahmat Nusantara berbasis *Ruby*.

Tabel 1. Kategori *User* Aplikasi

| <i>User</i> | Deskripsi |
|------------------------|---|
| Registrasi <i>User</i> | <i>User</i> yang terdaftar dalam sistem untuk masuk ke dalam aplikasi |
| Admin | <ul style="list-style-type: none"> • <i>User</i> dapat mengelola semua data, mengelola stok barang, input barang • <i>User</i> dapat <i>customize</i> mata uang yang dipakai • <i>User</i> dapat melihat laporan penjualan • Melayani interaksi antar pembeli |
| Client | <ul style="list-style-type: none"> • <i>User</i> dapat login • <i>User</i> dapat melihat list penjualan • <i>User</i> dapat belanja |

Pada tabel ini dijelaskan tentang rancang bangun website *e-commerce* ini terdiri dari beberapa *user* seperti admin dan *client* yang setiap kategori *user*-nya memiliki hak akses yang berbeda-beda.

B. Deskripsi Fitur

Pada tahapan ini dijelaskan mengenai fitur pada rancang bangun website *e-commerce* pada PT Hujan Rahmat Nusantara berbasis *Ruby*. Adapun deskripsi dari fitur tersebut dijelaskan dalam tabel berikut.

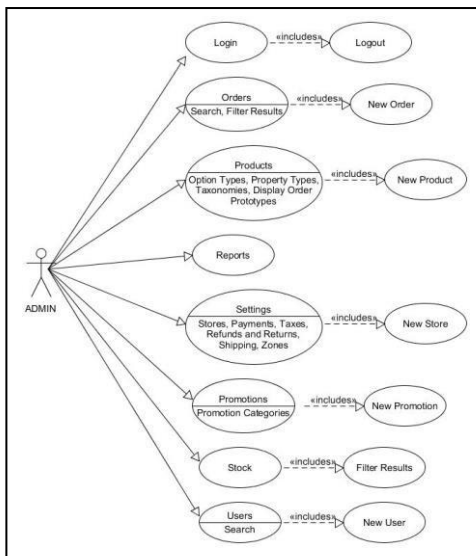
Tabel 2. Deskripsi Fitur

| Kode | Deskripsi |
|-------|---|
| R.001 | Orders untuk cek dan <i>approve</i> pemesanan produk dari <i>customer</i> |
| R.002 | Products diproyeksikan untuk memanfaatkan manajemen produk dengan metode CRUD (Create, Read, Update, Delete) |
| R.003 | Reports digunakan untuk melihat laporan-laporan transaksi |

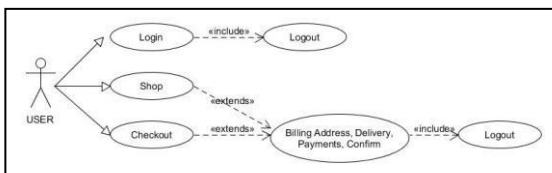
| | |
|-------|---|
| R.004 | Settings berguna untuk konfigurasi toko online seperti pembayaran, pajak, <i>refund</i> , pengantaran barang menggunakan jasa kurir, Wilayah dari pada jangkauan toko online |
| R.005 | Promotions untuk menambah daya tarik <i>customer</i> untuk menghemat perbelanjaannya |
| R.006 | Stock untuk memantau kesediaan produk dengan menggunakan metode CRUD (Create, Read, Update, Delete) |
| R.007 | Users untuk majemen pengguna dan dapat diatur produk dengan menggunakan metode CRUD (Create, Read, Update, Delete) |

C. Use Case Diagram

Pada gambar ini dijelaskan tentang *use case* diagram pada rancang bangun website *e-commerce* pada PT Hujan Rahmat Nusantara berbasis Ruby berdasarkan kebutuhan *user* dari hasil wawancara observasi kepada pihak terkait.



Gambar 1. Use Case Diagram Admin

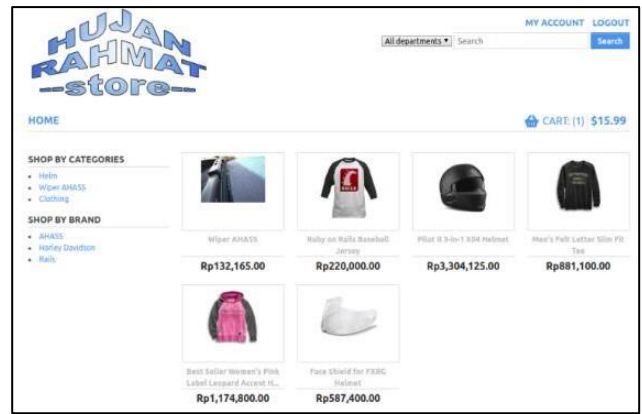


Gambar 2. Use Case Diagram User

5. IMPLEMENTASI DAN EVALUASI

5.1 Implementasi

Implementasi dan pengujian dilakukan bertujuan untuk memastikan bahwa aplikasi yang dibangun berjalan dengan baik dan sesuai dengan yang direncanakan.



Gambar 3. Halaman Store Front Hujan Rahmat

Pada gambar di atas adalah halaman depan *online store* PT Hujan Rahmat. Tampilan di atas dalam mode sesi sudah *login* dengan akun yang sudah di *register* oleh Admin. Ketika *user* tidak *login* pun tampilannya sama, yang membedakan hanya sesi *login*.

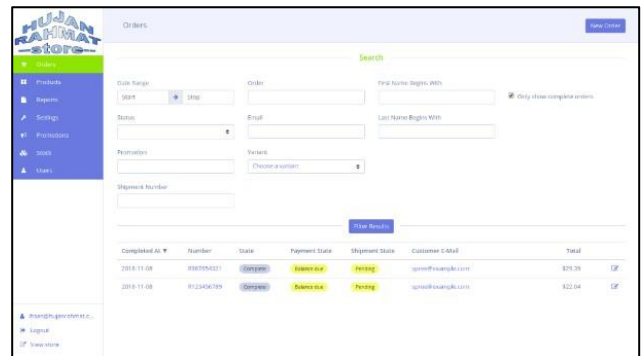
5.1.1 Halaman Login Admin



Gambar 4. Halaman Login Admin

Halaman *login* tampil dengan sangat sederhana berisilogo, *form login* untuk *email*, *password* dan *remember me*. Ketika *user* lupa dengan dengan *ID login* maka dapat menghubungi admin atau klik link *Forgot Password*.

5.1.2 Halaman pada Menu Orders

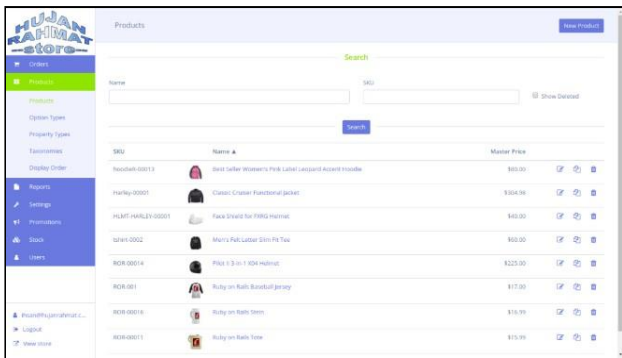


Gambar 5. Halaman pada Menu Orders

Menu Orders digunakan untuk menerima pemesanan baru dari customer yang mana dapat di proses secara lanjut seperti

konfirmasi pembayaran samapai pengantaran barang oleh kurir.

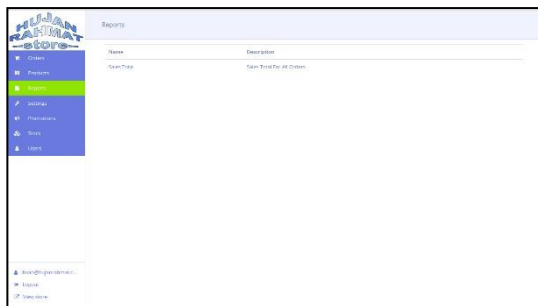
5.1.3 Halaman pada Menu *Products*



Gambar 6. Halaman pada Menu *Products*

Pada halaman ini disediakan list produk yang sudah di upload dan dengan halaman ini sudah di sediakan manajemen produk seperti membuat, melihat, edit, duplikat dan menghapus produk.

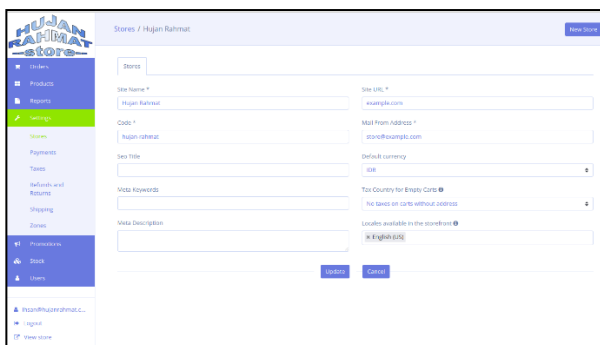
5.1.4 Halaman pada Menu *Reports*



Gambar 7. Halaman pada Menu *Reports*

Halaman *Reports* digunakan untuk melihat hasil kerja *sales* dalam memasarkan produk. Seperti list *sales* dan deskripsi yang melakukan transaksi untuk melayani *user*.

5.1.5 Halaman pada Meny *Settings*

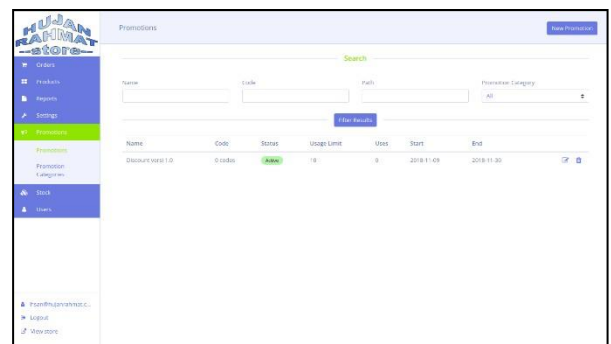


Gambar 8. Halaman pada Menu *Settings*

Pada halaman ini digunakan untuk mengatur online store yang tersedia, seperti konfigurasi pajak, pembayaran, sistem pengembalian barang dan zona wilayah. Selain itu pada

halamn ini dapat memiliki fitur *New Store* untuk menambah *Store Front* baru dengan situs URL yang berbeda.

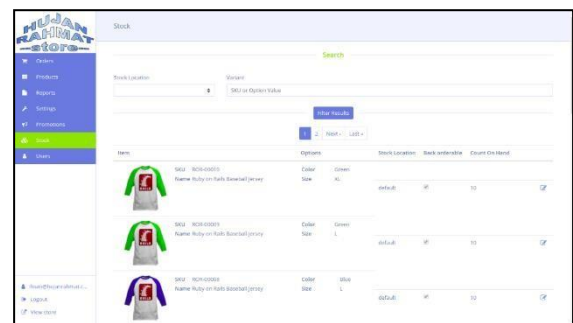
5.1.6 Halaman pada Menu *Promotions*



Gambar 9. Halaman pada Menu *Promotions*

Pada halaman *Promotions* digunakan untuk menambah daya tarik *customer* untuk membeli produk yang dijual seperti memberikan kupon dan promo dan pengaturan kupon yang lebih mendetail seperti sampai kapan berlakunya kupon yang sudah dibuat dan untuk produk tertentu.

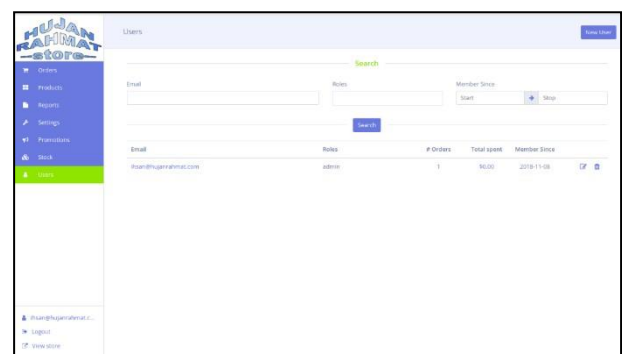
5.1.7 Halaamaan pada Menu *Stock*



Gambar 10. Halaman pada Menu *Stock*

Halaman *Stock* disediakan untuk menampilkan daftar stok barang yang tersedia atau sudah *sold-out*. Dengan halaman ini membantu admin untuk menyelaraskan stok gudang dengan stok di web.

5.1.8 Halaman pada Menu *User*



Gambar 11. Halaman pada Menu *User*

Halaman *User* digunakan untuk manajemen *User*. Fitur ini hanya dapat diakses oleh admin. Pada halaman ini dapat membuat, edit dan menghapus *user*.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan penelitian ini berhasil membangun Aplikasi *E-Commerce* pada PT Hujan Rahmat Nusantara berbasis Web menggunakan Bahasa pemrograman Ruby, *framework Ruby On Rails* dan PostgreSQL. dapat disimpulkan sebagai berikut:

Berdasarkan hasil pengujian *black box* dan pengujian *Usability*, Aplikasi *E-Commerce* pada PT Hujan Rahmat Nusantara berbasis Web mendapatkan hasil yang baik melebihi target yang ditentukan yaitu >60%, maka dapat disimpulkan, bahwa aplikasi Aplikasi *E-Commerce* pada PT Hujan Rahmat Nusantara berbasis Web, dapat membantu perusahaan dalam mengembangkan bisnisnya menjadi lebih efisien, serta dapat memudahkan PT. Hujan Rahmat Nusantara untuk berbagi informasi dan berkomunikasi dengan pelanggan dan penjual lain dengan cara yang lebih efektif dan efisien.

6.2 Saran

Pada penelitian ini penulis hanya membuat *E-Commerce* berbasis web pada PT Hujan Rahmat Nusantara yang sebelumnya pihak tersebut menggunakan aturan-aturan pencatatan manual, kemudian melakukan rekap pada aplikasi *Office* dan aplikasi *chatting* yang sudah sering digunakan, dikarenakan keterbatasan pengetahuan dalam menggunakan teknologi. Untuk selanjutnya sebaiknya bisa dikembangkan lagi agar dapat diakses secara *online* menggunakan protokol yang lebih aman seperti HTTPS dan tingkat keamanan lainnya, agar informasi penting tidak mudah diretas oleh pihak yang tidak bertanggung jawab. Serta dapat dibuatkan aplikasi *mobile*, yang dapat berjalan di lingkungan *smartphone*, sehingga *user* dapat melihat dan input informasi dimana pun dan kapan pun.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada pimpinan STT Terpadu Nurul Fikri, Pembimbing Tugas Akhir, Dewan penguji, Dosen STT Terpadu Nurul Fikri dan Pihak lain yang turut membantu dalam penelitian ini.

DAFTAR PUSTAKA

- [1] S. Kosasi. "Perancangan dan Pemanfaatan *E-Commerce* untuk Memperluas Pasar Produk *Furniture*," Seminar Nasional Teknologi Informasi dan Komunikasi, 2015.
- [2] Binus University, "Analisis dan Perancangan Aplikasi Jejaring Sosial Penjualan berbasis *Web*," 2008.
- [3] A. Alliance, "*Manifesto for Agile Software Development*," 2001, [Online]. Available: <http://www.agilemanifesto.org/>.
- [4] T. Handson, "*Model View Controller (MVC)*," p. 9, 2005.
- [5] ASM BSI Jakarta, "Definisi UML," *Rekayasa Perangkat Lunak Dengan Model Unified Process Studi Kasus: Sistem Informasi Journal*, 2016.
- [6] Anonim, "Menguasai Ruby dalam 20 Menit," 2007, [Online]. Available: <http://www.rubylang.org/id/documentation/quickstar>
- [7] *Language User Guide. 1st Edition*. Addison Wesley.
- [8] Nawawi Muhammad Ridwan, "Analisa dan Perancangan Aplikasi Jejaring Sosial Penjualan berbasis *Web*," 2008.
- [9] Rudy, George A. A. Ryan M. A. Yohanna, "Perancangan *E-Commerce* pada PT Daf," 2010.
- [10] S. Damayanti, H. Maryanti, E. S. Djatikusuma, "Sistem Informasi *E-Commerce* pada PT An-Nur Cahaya Indah Palembang," 2011.



PENERAPAN EFEKTIFITAS FIREWALL PADA *SOFTWARE DEFINED NETWORK* BERBASIS OPENFLOW

Fathul Muiin¹, Henry Saptono²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
fmuiin14@gmail.com, henry@nurulfikri.co.id

Abstract

The use of internet access in the world is growing and in line with increasingly complex computer network technology. Therefore, data security on a computer becomes a crucial part of a network. Thus, SDN is a solution to provide computer network needs today. Software-Defined Network (SDN) is an approach to network technology that simplifies network control and management. This network will use the OpenFlow protocol, whose primary principle is to separate the control plane and data plane functions on the device. The network control on a controller is programmable, so with the SDN, the network will be easily managed and more flexible. This firewall implementation and analysis uses a mininet emulator to create a simple network topology. In firewall, testing uses the XML language for data flow implementation, then uses the postman application as a tool to add a new flow table to the switch, and the controller used is opendaylight.

Keywords: *Software Defined Network; Openflow; Firewall; Controller; Opendaylight; Postman*

Abstrak

Penggunaan akses internet di dunia semakin berkembang, dan selaras dengan perkembangan teknologi jaringan komputer yang semakin kompleks. Oleh karena itu, keamanan data pada sebuah komputer menjadi salah satu bagian yang sangat penting dalam sebuah jaringan. Dan SDN merupakan sebuah solusi untuk menyediakan kebutuhan jaringan komputer saat ini. *Software Defined Network (SDN)* merupakan pendekatan pada teknologi jaringan yang melakukan penyederhanaan terhadap kontrol dan manajemen jaringan. Pada jaringan ini nantinya akan menggunakan protokol *openflow*, yang prinsip utamanya memisahkan fungsi *control plane* dan *data plane* pada perangkat. Kontrol jaringan pada sebuah *controller* bersifat *programmable*, jadi dengan adanya SDN maka jaringan akan mudah diatur dan lebih fleksibel. Implementasi dan analisis *firewall* ini menggunakan *emulator mininet* untuk membuat topologi jaringan yang sederhana. Dalam pengujian *firewall* menggunakan bahasa XML untuk implementasi aliran data, lalu menggunakan aplikasi *postman* sebagai alat untuk menambahkan *flow table* baru pada *switch*, dan *controller* yang digunakan adalah *Opendaylight*.

Kata kunci: *Software Defined Network, Openflow, Firewall, Controller, Opendaylight, Postman*

1. PENDAHULUAN

Layanan internet dan teknologi informasi berkembang dengan berbagai kompleksitas, desain, manajemen, dan operasional yang menyebabkan jumlah perangkat yang terhubung dan volume aliran data dalam jaringan telah meningkat dengan sangat pesat dalam beberapa tahun terakhir. Dalam hal penerapan teknologi perangkat jaringan, sangat dibutuhkan metode baru untuk melakukan kontrol dan pengelolaan jaringan yang efisien dan efektif. Disamping itu, kita juga perlu memperhatikan aspek keamanan dalam perangkat jaringan. Perusahaan dan penyedia layanan paham akan keterbatasan arsitektur jaringan konvensional yang tidak fleksibel dan membutuhkan biaya banyak.

Software Defined Network (SDN) adalah sebuah konsep pendekatan jaringan komputer dimana sistem pengontrol dari arus data (*data plane*) dipisahkan dari perangkat kerasnya (*control plane*). Pemisahan *data plane* dan *control plane* pada perangkat jaringan komputer seperti *router* dan *switch* memungkinkan kita memprogram perangkat tersebut sesuai dengan yang diinginkan secara terpusat [1].

Analisis efektivitas *firewall* pada SDN akan diteliti dengan cara mengatur *flow table* pada suatu topologi jaringan dengan menggunakan *tools postman* sebagai metode untuk *push* menuju *web service* yang disediakan oleh *controller opendaylight*, dan menggunakan bahasa XML. Cara

mengatur *flow table* adalah dengan menerapkan sistem *drop* pada sebuah topologi jaringan, karena pada setiap topologi jaringan, *default* kontrol akses dari jaringannya adalah *accept*. Dan peneliti menguji kontrol akses dengan cara *drop* menggunakan *IP Address*, *MAC Address*, *TCP*, *UDP*, dan *ICMP*. Sehingga pada nantinya akan dibuat kesimpulan dari hasil pengujian dan implementasi. Dalam pelaksanaannya, akan ada 3 topologi yang nanti di ujikan yaitu topologi *single*, topologi *linear*, dan topologi *tree*. Peneliti menggunakan macam-macam topologi karena pada penelitian ini ingin mengetahui pola dari tingkah laku jaringan SDN dan kinerja yang dimiliki terhadap jaringan yang semakin besar dan semakin kompleks.

2. METODOLOGI PENELITIAN

2.1 Studi Literatur

Tahapan ini dilakukan dengan mencari, mengumpulkan dan membaca artikel, jurnal ilmiah, buku elektronik (*e-book*), website maupun beberapa skripsi peneliti lainnya untuk mengkaji mengenai implementasi teknologi SDN.

Hasil dari studi literatur adalah pembuatan acuan rancangan penelitian serta acuan bagaimana penelitian harus dilakukan dan data apa saja yang diperlukan sehingga tujuan pada penelitian ini dapat tercapai.

2.2 Analisis Kebutuhan Sistem

Tahapan ini dilakukan untuk menganalisis apa saja kebutuhan yang diperlukan sebuah sistem dalam pengimplementasiannya pada *firewall*. *Firewall* yang diterapkan *software defined network*, kemudian dievaluasi apakah sudah optimal atau belum dalam penerapannya pada sebuah SDN.

2.3 Perancangan Sistem

Pada tahap ini akan dilakukan perancangan sistem yang dibuat dan diujikan, yang meliputi perancangan topologi untuk pengujian *firewall*.

2.4 Implementasi

Setelah dilakukan analisis tujuan dan perancangan sistem maka masuk ke fase implementasi, dimana pada fase ini dilakukan proses konfigurasi dari topologi yang telah dibuat ke simulasi jaringan hingga siap untuk dilakukan pengujian *firewall*.

2.5 Pengujian

Pada tahap ini akan dilakukan pengujian terhadap sistem yang akan dianalisis untuk mengukur bagaimana dampak *software defined network* yang diimplementasikan pada *firewall* dan seberapa baik perintah yang diberi berjalan dengan semestinya.

2.6 Analisis Hasil

Pada tahap ini berisi sebuah analisa hasil pengujian dari simulasi dan implementasi yang telah dibuat, apakah sesuai dengan perumusan masalah dan tujuan yang telah dibuat oleh peneliti atau tidak.

2.7 Penarikan Kesimpulan dan Saran

Dari hasil pengujian dapat ditarik kesimpulan apakah teknologi yang diterapkan dapat menjadi inovasi baru bagi keamanan sebuah *firewall* pada komputer atau tidak, dan menjadi solusi bagi banyak pihak serta memerlukan saran-saran untuk optimalisasi serta kelanjutan penelitian berikutnya.

3. ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan *Hardware*

Dalam melakukan penelitian ini, kebutuhan *hardware* minimal yang dapat digunakan untuk melakukan penelitian adalah sesuai dengan spesifikasi dari *Linux Ubuntu 16.04 LTS*. Dan spesifikasi minimal *hardware*nya adalah sebagai berikut:

Prosesor : Dual Core 2 GHz

RAM : 2GB

Hardisk : 25GB

VGA : 1024 x 768

Dalam penelitian kali ini, peneliti menggunakan laptop dengan spesifikasi sebagai berikut:

Prosesor : Intel® Core™ i5-3320M CPU@2.60GHz

RAM : 8GB

Hardisk : 125GB SSD

VGA : 1024 x 768

3.2 Analisis Kebutuhan *Software*

Software yang digunakan dalam penelitian ini adalah sebagai berikut:

Host : Mininet 2.3.0d1

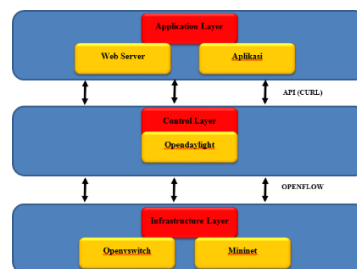
Switch : Openvswitch ovs-vswnchd 2.5.4

Controller : Opendaylight Nitrogen SR3(seventh release)

OS : Linux Ubuntu 16.04 LTS Xenial Xerus

3.3 Perancangan Arsitektur Sistem

Berikut adalah arsitektur sistem dalam pengujian efektivitas *firewall* pada SDN.



Gambar 1. Arsitektur Sistem

Ada tiga lapisan dalam perancangan arsitektur sistem yang akan dibuat, berikut penjelasan masing- masing dari tiga lapisan tersebut:

1. *Infrastructure Layer*

Pada lapisan ini, peneliti menggunakan *openvswitch* sebagai *switch* virtual yang digunakan untuk melakukan penelitian. Sedangkan *Mininet* adalah emulator berbasis CLI (*Command Line Interface*) yang digunakan untuk membuat sebuah topologi jaringan sesuai dengan kebutuhan yang dapat disesuaikan dengan penelitian.

2. *Control Layer*

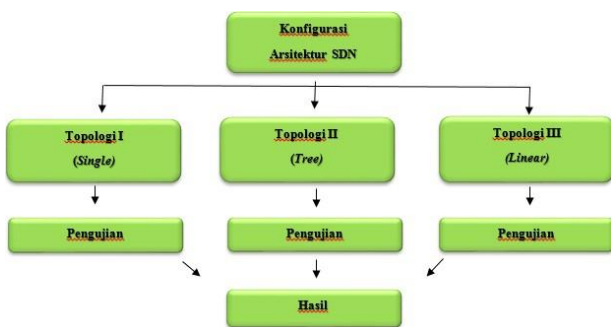
Pada lapisan *Control Layer*, peneliti menggunakan *software opendaylight* sebagai pengontrol data dan menjadi penghubung antara *Infrastructure Layer* (lapisan bawah) dan *Application Layer* (lapisan atas).

3. *Application Layer*

Merupakan tempat yang menjadi antar muka untuk memudahkan peneliti dalam melakukan fungsi konfigurasi, kontrol, dan evaluasi.

3.4 Rancangan Pengujian

Penelitian kali ini menggunakan 3 topologi yang dibuat menggunakan *software mininet*. Topologi yang dibuat yaitu topologi *single*, topologi *tree*, dan topologi *linear*. Hal yang di uji dalam *firewall* pada *software defined network* adalah *blocking* berdasarkan *mac address*, *blocking* koneksi *TCP*, *UDP*, *ICMP*, dan *blocking* berdasarkan *IP address*. Untuk lebih leTngkapnya, silahkan perhatikan gambar di bawah:



Gambar 2. Rancangan Pengujian dengan Topologi

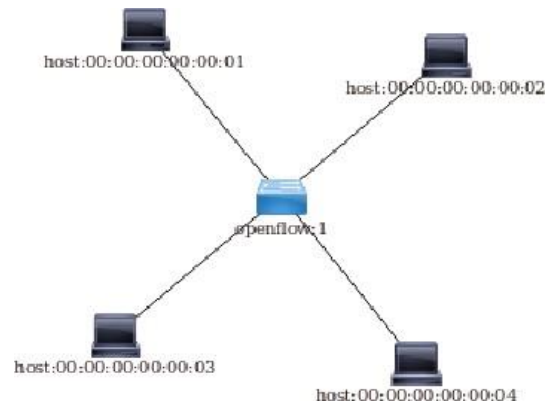
1. Konfigurasi Arsitektur SDN

Pada bagian ini, peneliti akan melakukan pengaturan terlebih dahulu untuk konfigurasi *Software Defined Network*, implementasi dengan menggunakan simulasi *mininet*, dan nantinya akan dilakukan penelitian dengan 3 jenis topologi:

a. Topologi 1 (*Single Topology*)

Topologi jaringan yang pertama dibuat menggunakan *software mininet*, dan menggunakan topologi *single*

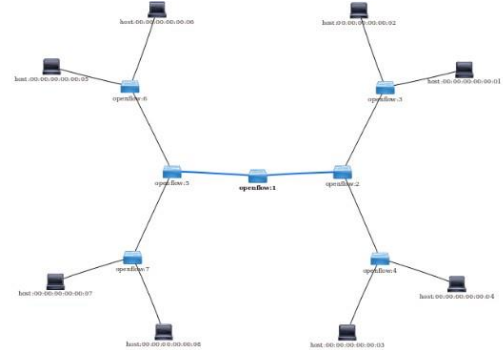
dengan 1 buah *switch* dan 4 *host*. Berikut gambar topologi *single* yang akan di uji:



Gambar 3. Single Topology

b. Topologi 2 (*Tree Topology*)

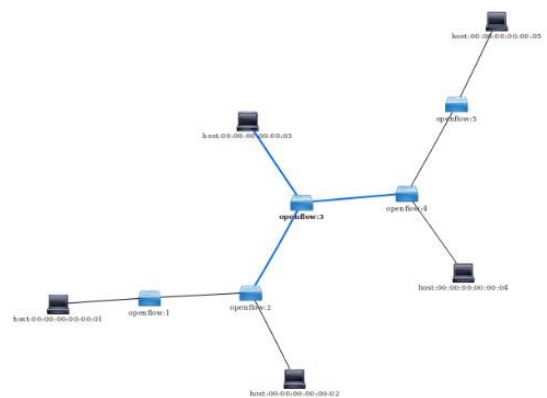
Topologi jaringan yang kedua dibuat menggunakan *software mininet*, dan menggunakan topologi *tree* dengan 7 buah *switch* dan 8 *host*. Berikut gambar topologi *tree* yang akan diuji:



Gambar 4. Tree Topology

c. Topologi 3 (*Linear Topology*)

Topologi jaringan yang ketiga dibuat menggunakan *software mininet*, dan menggunakan *topologi linear* dengan 5 buah *switch* dan 5 *host*. Berikut gambar topologi linear yang akan diuji:



Gambar 5. Linear Topology

2. Pengujian

Secara *default*, topologi jaringan yang dibangun dengan mininet ini tidak menerapkan akses kontrol (*firewall*). Untuk itu, dalam tahap pengujian ini peneliti akan melakukan pengaturan dan penerapan akses kontrol (*firewall*) dalam jaringan berbasis SDN yang dibangun dengan simulasi menggunakan *mininet*. Skenario pengaturan akses kontrol (*firewall*) tersebut bisa dilihat dari tabel berikut ini.

Tabel 1. Pengujian

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | 00:00:00:00:00:04 | ANY | ANY | ICMP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | 00:00:00:00:00:03 | 10.0.0.4 | 10.0.0.5 | TCP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:03 | 00:00:00:00:00:02 | 10.0.0.1 | 10.0.0.3 | UDP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:04 | 00:00:00:00:00:01 | ANY | 10.0.0.6 | UDP | ANY | DROP | SUCCESS |

Dari table di atas, peneliti akan menerapkan akses kontrol dengan kriteria-kriteria sebagai berikut:

- a. SRC MAC : Source Mac Address
- b. DST MAC: Destination Mac Address
- c. SRC IP: Source IP Address
- d. DST IP: Destination IP Address
- e. PROTOCOL: TCP/ UDP
- f. DST PORT: Destination Port
- g. ACTION: DROP/ ALLOW
- h. RESULT: SUCCESS/ DENY

3. Hasil

Hasil pengujian dari penerapan akses kontrol (*firewall*) yang dilakukan oleh peneliti dalam tahapan pengujian adalah suatu kondisi atau status keberhasilan dari skenario *rule* yang diterapkan, yang nantinya akan dibuat laporan sehingga menjadi referensi bagi penelitian pengujian *firewall* menggunakan *opendaylight* untuk penelitian selanjutnya.

4. HASIL PENELITIAN

A. *Blocking IP Address Topologi 1 (Single)*

Pengujian pertama melakukan pengalamatan *rule* adalah dengan *blocking* aliran data berdasarkan *IP Address*. Pengujian ini mencoba menambahkan *flow* pada tabel *flow openflow:4 (s1)*, sehingga saat ada aliran data menuju *host 4* dari *host 2*, koneksinya akan di blok. Berikut perintahnya dengan menggunakan *POSTMAN*.

Pada *Authorization*:

- Type : Basic Auth
- Username : admin
- Password : admin

Pada *Headers*:

- Key : Content-Type
- Value : application/xml

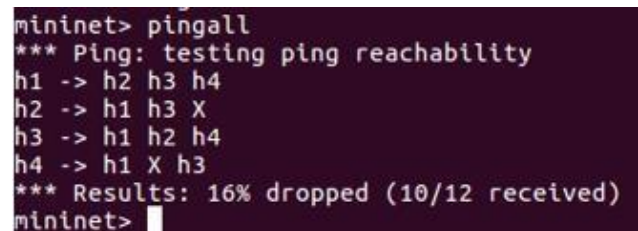
Pada *URL Bar*:

- PUT <http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/1>

Pada *Body (raw)*

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <id>1</id>
  <table_id>0</table_id>
  <match>
    <ethernet-match>
      <ethernet-type>
<type>2048</type> </ethernet-type>
    </ethernet-match>
    <ipv4-
destination>10.0.0.4/32</ipv4-destination>
    <in-port>2</in-port>
  </match>
  <priority>15</priority>
</flow>
```

Hasil dari *flow* yang baru ditambahkan sebagai berikut. Antara h2 dan h4 tidak bisa saling berkomunikasi, karena aliran datanya telah di blok.



Gambar 6. Hasil dari *Flow* Berdasarkan *IP Address Topologi 1*

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut.

Tabel 2. Pengujian Berdasarkan *IP Address Topologi 1*

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:02 | 00:00:00:00:00:04 | 10.0.0.2 | 10.0.0.4 | ANY | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:01 | 00:00:00:00:00:03 | 10.0.0.1 | 10.0.0.3 | ANY | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:01 | 00:00:00:00:00:04 | 10.0.0.1 | 10.0.0.4 | ANY | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:04 | 00:00:00:00:00:01 | 10.0.0.4 | 10.0.0.1 | ANY | ANY | DROP | SUCCESS |

Dapat terlihat bahwa hasil dari pengujian yang di lakukan dalam kontrol akses *firewall* pada *rule flow* berhasil dilakukan sesuai dengan perintah yang di jalankan.

B. *Blocking IP Address Topologi 2 (Tree)*

Selanjutnya *blocking* aliran data berdasarkan *IP Address Topologi 2*. Kali ini akan mencoba menambahkan *flow* pada *table flow openflow:5*, dengan aliran data membiarkan semua jaringan dapat berkomunikasi dengan h8, kecuali h5 dan h6 yang *rulanya* ditambahkan pada *openflow:5*, dengan menggunakan perintah *PUT* pada *address bar*, dan alamat yang ditujukan adalah sebagai berikut.


```
http://localhost:8181/restconf/config/.opendaylight-
inventory:nodes/node/openflow:5/table/0/flow/1
```

Pada *Body (raw)*

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <id>1</id>
  <table_id>0</table_id>
  <match>
    <ethernet-match>
      <ethernet-type>
<type>2048</type> </ethernet-type>
    </ethernet-match>
    <ipv4-
destination>10.0.0.8/32</ipv4-destination>
    <in-port>1</in-port>
  </match>
  <priority>20</priority>
</flow>
```

Hasilnya sebagai berikut:

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8
h2 -> h1 h3 h4 h5 h6 h7 h8
h3 -> h1 h2 h4 h5 h6 h7 h8
h4 -> h1 h2 h3 h5 h6 h7 h8
h5 -> h1 h2 h3 h4 h6 h7 X
h6 -> h1 h2 h3 h4 h5 h7 X
h7 -> h1 h2 h3 h4 h5 h6 h8
h8 -> h1 h2 h3 h4 X X h7
*** Results: 7% dropped (52/56 received)
mininet>
```

Gambar 7. Hasil dari Flow Berdasarkan IP Address Topologi 2

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut.

Tabel 3. Pengujian Berdasarkan IP Address Topologi 2

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:05 | 00:00:00:00:00:08 | 10.0.0.5 | 10.0.0.8 | ANY | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:01 | ANY | 10.0.0.1 | ANY | ANY | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:08 | 00:00:00:00:00:06 | 10.0.0.8 | 10.0.0.6 | ANY | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:04 | ANY | 10.0.0.4 | ANY | ANY | ANY | DROP | SUCCESS |

C. *Blocking IP Address Topologi 3 (Linear)*

Selanjutnya *blocking* aliran data berdasarkan IP Address Topologi 3. Kali ini akan mencoba untuk menambahkan flow pada table flow openflow:3, dengan aturan h3 tidak bisa melewati destination port openflow:3:2, namun jaringan selain destination port 2 bisa menjalankan aliran data. Berikut perintahnya dengan menggunakan *POSTMAN*.

Pada *Authorization*:

- Type : Basic Auth
- Username : admin
- Password : admin

Pada *Headers*:

- Key : Content-Type
- Value : application/xml

Pada *URL Bar*:

- PUT
 http://localhost:8181/restconf/config/.opendaylight-
 inventory:nodes/node/openflow:3/table/0/flow/2

Pada *Body (raw)*

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <id>2</id>
  <table_id>0</table_id>
  <match> <ethernet-match>
    <ethernet-type>
<type>2048</type>
  </ethernet-type>
  </ethernet-match>
  <ipv4-
destination>10.0.0.3/32</ipv4-destination>
  <in-port>2</in-port>
</match>
  <priority>29</priority>
</flow>
```

Hasilnya sebagai berikut:

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X h4 h5
h2 -> h1 X h4 h5
h3 -> X X h4 h5
h4 -> h1 h2 h3 h5
h5 -> h1 h2 h3 h4
*** Results: 20% dropped (16/20 received)
mininet>
```

Gambar 8. Hasil dari Flow Berdasarkan IP Address Topologi 3

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut:

Tabel 4. Pengujian Berdasarkan IP Address Topologi 3

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | 00:00:00:00:00:03 | 10.0.0.1 | 10.0.0.3 | ANY | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:05 | ANY | 10.0.0.5 | ANY | ANY | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:04 | ANY | 10.0.0.4 | ANY | ANY | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:02 | 00:00:00:00:00:03 | 10.0.0.2 | 10.0.0.3 | ANY | ANY | DROP | SUCCESS |

D. *Blocking MAC Address Topologi 1 (Single)*

Selanjutnya *blocking* aliran data berdasarkan MAC Address Topologi 1. Dan aturan yang akan di tambahkan adalah semua aliran data yang menuju atau dari h1 akan di blok, dengan menggunakan perintah *PUT* pada address bar, dan alamat yang ditujukan adalah sebagai berikut.

```
http://localhost:8181/restconf/config/opendayl
ight-
inventory:nodes/node/openflow:1/table/0/flow/1
```

Berikut teks *XML* yang perlu ditambahkan menggunakan *POSTMAN*:


```
<flow xmlns="urn:opendaylight:flow:inventory">
  <table_id>0</table_id>
  <id>1</id>
  <cookie_mask>255</cookie_mask>
  <installHw>false</installHw>
  <match>
    <ethernet-match>
      <ethernet-source>
        <address>00:00:00:00:00:01</address>
      </ethernet-source>
    </ethernet-match>
  </match>
  <priority>15</priority>
</flow>
```

Hasilnya sebagai berikut.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X
h2 -> X h3 h4
h3 -> X h2 h4
h4 -> X h2 h3
*** Results: 50% dropped (6/12 received)
mininet>
```

Gambar 9. Hasil dari Flow Berdasarkan MAC Address Topologi 1

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut:

Tabel 5. Pengujian Berdasarkan MAC Address Topologi 1

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | 00:00:00:00:00:04 | 10.0.0.1 | 10.0.0.4 | ANY | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | 00:00:00:00:00:05 | 10.0.0.2 | 10.0.0.5 | ANY | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:04 | 00:00:00:00:00:02 | 10.0.0.4 | 10.0.0.2 | ANY | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:05 | ANY | 10.0.0.5 | ANY | ANY | ANY | DROP | SUCCESS |

E. Blocking MAC Address Topologi 2 (Tree)

Selanjutnya blocking aliran data berdasarkan MAC Address Topologi 2. Dan aturannya adalah menambahkan aturan flow firewall pada openflow:5 (s5) untuk blocking ke h8, sehingga nantinya semua aliran data yang menuju/ dari h8 tidak bisa berkomunikasi dengan h8, kecuali h7, dengan menggunakan perintah PUT pada address bar, dan alamat yang ditujukan adalah sebagai berikut.

```
http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:5/table/0/flow/1
```

Berikut teks XML yang perlu ditambahkan menggunakan POSTMAN.

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <table_id>0</table_id>
  <id>1</id>
  <cookie_mask>255</cookie_mask>
  <installHw>false</installHw>
  <match>
    <ethernet-match>
      <ethernet-source>
        <address>00:00:00:00:00:08</address>
      </ethernet-source>
    </ethernet-match>
  </match>
  <priority>15</priority>
</flow>
```

Hasilnya sebagai berikut.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 X
h2 -> h1 h3 h4 h5 h6 h7 X
h3 -> h1 h2 h4 h5 h6 h7 X
h4 -> h1 h2 h3 h5 h6 h7 X
h5 -> h1 h2 h3 h4 h6 h7 X
h6 -> h1 h2 h3 h4 h5 h7 X
h7 -> h1 h2 h3 h4 h5 h6 h8
h8 -> X X X X X X h7
*** Results: 21% dropped (44/56 received)
mininet>
```

Gambar 10. Hasil dari Flow Berdasarkan MAC Address Topologi 2

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut.

Tabel 6. Pengujian Berdasarkan MAC Address Topologi 2

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | ANY | 10.0.0.1 | ANY | ICMP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | 00:00:00:00:00:01 | 10.0.0.2 | 10.0.0.1 | ICMP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:03 | 00:00:00:00:00:04 | 10.0.0.3 | 10.0.0.4 | ICMP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:05 | 00:00:00:00:00:01 | 10.0.0.5 | 10.0.0.1 | ICMP | ANY | DROP | SUCCESS |

F. Blocking MAC Address Topologi 3 (Linear)

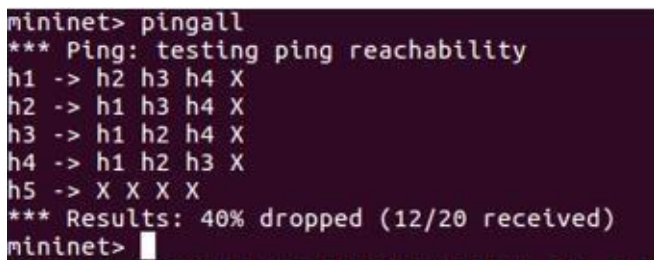
Selanjutnya blocking aliran data berdasarkan MAC Address Topologi 3. Dan aturan yang akan di tambahkan adalah semua aliran data yang menuju/ dari h5 akan di blok, dengan menggunakan perintah PUT pada address bar, dan alamat yang ditujukan adalah sebagai berikut.

```
http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:5/table/0/flow/1
```

Berikut text XML yang perlu ditambahkan menggunakan POSTMAN.

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <table_id>0</table_id>
  <id>1</id>
  <cookie_mask>255</cookie_mask>
  <installHw>false</installHw>
  <match>
    <ethernet-match>
      <ethernet-source>
        <address>00:00:00:00:00:05</address>
      </ethernet-source>
    </ethernet-match>
  </match>
  <priority>15</priority>
</flow>
```

Hasilnya sebagai berikut.



Gambar 11. Hasil dari Flow Berdasarkan MAC Address Topologi 3

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat dilihat pada tabel berikut.

Tabel 7. Pengujian Berdasarkan MAC Address Topologi 3

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | 00:00:00:00:00:04 | 10.0.0.1 | 10.0.0.4 | ANY | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | 00:00:00:00:00:05 | 10.0.0.2 | 10.0.0.5 | ANY | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:04 | 00:00:00:00:00:02 | 10.0.0.4 | 10.0.0.2 | ANY | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:05 | ANY | 10.0.0.5 | ANY | ANY | ANY | DROP | SUCCESS |

G. Blocking UDP Address Topologi 1 (Single)

Selanjutnya *blocking* aliran data berdasarkan *Port UDP* Topologi 1. Dan aturan yang akan di tambahkan adalah menutup komunikasi yang akan dilakukan oleh h2 pada *UDP* (Port 17), dengan menggunakan perintah *PUT* pada *address bar*, dan alamat yang ditujukan adalah sebagai berikut.

```
http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/1
```

Berikut teks *XML* yang perlu ditambahkan menggunakan *POSTMAN*.

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <table_id>0</table_id>
  <id>1</id>
  <cookie_mask>255</cookie_mask>
  <match>
    <ethernet-match>
      <ethernet-type>
        <type>2048</type>
      </ethernet-type>
    </ethernet-match>
    <ip-match>
      <ip-protocol>17</ip-protocol>
    </ip-match>
    <ipv4-destination>10.0.0.2/32</ipv4-
destination>
    <in-port>1</in-port>
  </match>
  <priority>50</priority>
  <hard-timeout>1200</hard-timeout>
  <cookie>8</cookie>
  <idle-timeout>3400</idle-timeout>
</flow>
```

Hasilnya sebagai berikut.



Gambar 12. Hasil dari Flow Berdasarkan UDP Address Topologi 1

Peneliti juga melakukan beberapa pengujianlainnya, dan hasilnya dapat di lihat pada tabel berikut.

Tabel 8. Pengujian Berdasarkan UDP Address Topologi 1

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | 00:00:00:00:00:04 | 10.0.0.1 | 10.0.0.4 | UDP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | 00:00:00:00:00:04 | 10.0.0.2 | 10.0.0.4 | UDP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:04 | 00:00:00:00:00:01 | 10.0.0.4 | 10.0.0.1 | UDP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:02 | 00:00:00:00:00:03 | 10.0.0.2 | 10.0.0.3 | UDP | ANY | DROP | SUCCESS |

H. Blocking UDP Address Topologi 2 (Tree)

Selanjutnya *blocking* aliran data berdasarkan *Port UDP* Topologi 2. Dan aturan yang akan di tambahkan adalah menutup komunikasi yang akan dilakukan oleh h2 pada *UDP* (Port 17), dengan menggunakan perintah *PUT* pada *address bar*, dan alamat yang ditujukan adalah sebagai berikut:

```
http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/1
```

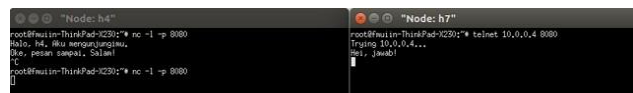
Berikut teks *XML* yang perlu ditambahkan menggunakan *POSTMAN*.

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <table_id>0</table_id>
  <id>1</id>
  <cookie_mask>255</cookie_mask>
  <match>
    <ethernet-match>
      <ethernet-type>
        <type>2048</type>
      </ethernet-type>
    </ethernet-match>
    <ip-match>
      <ip-protocol>17</ip-protocol>
    </ip-match>
    <ipv4-destination>10.0.0.1/32</ipv4-destination>
  </match>
  <in-port>2</in-port>
  </match>
  <priority>50</priority>
  <hard-timeout>1200</hard-timeout>
  <cookie>8</cookie>
  <idle-timeout>3400</idle-timeout>
</flow>
```

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <table_id>0</table_id>
  <id>1</id>
  <cookie_mask>255</cookie_mask>
  <match>
    <ethernet-match>
      <ethernet-type>
        <type>2048</type>
      </ethernet-type>
    </ethernet-match>
    <ip-match>
      <ip-protocol>6</ip-protocol>
    </ip-match>
    <ipv4-destination>10.0.0.4/32</ipv4-destination>
  </match>
  <in-port>2</in-port>
  </match>
  <priority>20</priority>
  <hard-timeout>1200</hard-timeout>
  <cookie>8</cookie>
  <idle-timeout>3400</idle-timeout>
</flow>
```

Saat coba untuk mengirim pesan menggunakan *port UDP* ke h1, maka hanya h2, h3, dan h4 saja yang bisa berkomunikasi dengan h1. Karena *port 2* pada s1 telah ditambahkan *rule* baru, sehingga h5, h6, h7, dan h8 tidak bisa berkomunikasi menggunakan protokol *UDP*.

Hasilnya sebagai berikut.



Gambar 13. Hasil dari Flow Berdasarkan UDP Address Topologi 3

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut.

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut.

Tabel 9. Pengujian Berdasarkan UDP Address Topologi 2

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:05 | 00:00:00:00:00:01 | 10.0.0.5 | 10.0.0.1 | UDP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:01 | 00:00:00:00:00:04 | 10.0.0.1 | 10.0.0.4 | UDP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:07 | 00:00:00:00:00:08 | 10.0.0.7 | 10.0.0.8 | UDP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:04 | 00:00:00:00:00:07 | 10.0.0.4 | 10.0.0.7 | UDP | ANY | DROP | SUCCESS |

Tabel 10. Pengujian Berdasarkan UDP Address Topologi 3

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:05 | 00:00:00:00:00:02 | 10.0.0.5 | 10.0.0.2 | UDP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | 00:00:00:00:00:01 | 10.0.0.2 | 10.0.0.1 | UDP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:05 | 00:00:00:00:00:01 | 10.0.0.5 | 10.0.0.1 | UDP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:03 | 00:00:00:00:00:01 | 10.0.0.3 | 10.0.0.1 | UDP | ANY | DROP | SUCCESS |

I. Blocking UDP Address Topologi 3 (Linear)

J. Blocking TCP Address Topologi 1 (Single)

Selanjutnya *blocking* aliran data berdasarkan *Port TCP* Topologi 2. Aturan yang akan di tambahkan adalah menutup jalur komunikasi untuk h4 pada *port 2* di s1, dengan menggunakan perintah *PUT* pada *address bar*, dan alamat yang ditunjukkan adalah sebagai berikut.

Selanjutnya *blocking* aliran data berdasarkan *Port TCP* Topologi 1. Dan aturan yang akan di tambahkan adalah menutup komunikasi yang akan dilakukan oleh h2 pada *TCP (Port 6)*, dengan menggunakan perintah *PUT* pada *address bar*, dan alamat yang ditunjukkan adalah sebagai berikut.

```
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/1
```

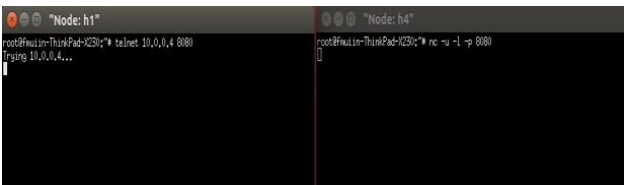
```
http://localhost:8181/restconf/config/opendaylight-inventory:nodes/node/openflow:1/table/0/flow/1
```

Berikut teks *XML* yang perlu ditambahkan menggunakan *POSTMAN*.

Berikut teks *XML* yang perlu ditambahkan menggunakan *POSTMAN*.

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <table_id>0</table_id>
  <id>1</id>
  <cookie_mask>255</cookie_mask>
  <match>
    <ethernet-match>
      <ethernet-type>
<type>2048</type>
      </ethernet-type>
    </ethernet-match>
    <ip-match>
      <ip-protocol>6</ip-protocol>
    </ip-match>
    <ipv4-destination>10.0.0.4/32</ipv4-
destination>
    <in-port>1</in-port>
  </match>
  <priority>20</priority>
  <hard-timeout>1200</hard-timeout>
  <cookie>8</cookie>
  <idle-timeout>3400</idle-timeout>
</flow>
```

Hasilnya sebagai berikut.



Gambar 14. Hasil dari Flow Berdasarkan TCP Address Topologi 1

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat dilihat pada tabel berikut.

Tabel 11. Pengujian Berdasarkan TCP Address Topologi 1

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | 00:00:00:00:00:04 | 10.0.0.1 | 10.0.0.4 | TCP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | 00:00:00:00:00:04 | 10.0.0.2 | 10.0.0.4 | TCP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:04 | 00:00:00:00:00:02 | 10.0.0.4 | 10.0.0.2 | TCP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:02 | 00:00:00:00:00:03 | 10.0.0.2 | 10.0.0.3 | TCP | ANY | DROP | SUCCESS |

K. Blocking TCP Address Topologi 2 (Tree)

Selanjutnya blocking aliran data berdasarkan Port TCP Topologi 2. Aturan yang akan di tambahkan adalah menutup jalur komunikasi untuk h4 pada port 2 di s1, dengan menggunakan perintah PUT pada address bar, dan alamat yang ditujukan adalah sebagai berikut.

```
http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/1
```

Berikut teks XML yang perlu ditambahkan menggunakan POSTMAN.

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <table_id>0</table_id>
  <id>1</id>
  <cookie_mask>255</cookie_mask>
  <match>
    <ethernet-match>
      <ethernet-type>
<type>2048</type>
      </ethernet-type>
    </ethernet-match>
    <ip-match>
      <ip-protocol>6</ip-protocol>
    </ip-match>
    <ipv4-destination>10.0.0.4/32</ipv4-
destination>
    <in-port>2</in-port>
  </match>
  <priority>20</priority>
  <hard-timeout>1200</hard-timeout>
  <cookie>8</cookie>
  <idle-timeout>3400</idle-timeout>
</flow>
```

Hasilnya sebagai berikut.



Gambar 15. Hasil dari Flow Berdasarkan TCP Address Topologi 2

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat dilihat pada tabel berikut.

Tabel 12. Pengujian Berdasarkan TCP Address Topologi 2

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | 00:00:00:00:00:02 | 10.0.0.1 | 10.0.0.2 | TCP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | 00:00:00:00:00:07 | 10.0.0.2 | 10.0.0.7 | TCP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:04 | 00:00:00:00:00:08 | 10.0.0.4 | 10.0.0.8 | TCP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:02 | 00:00:00:00:00:05 | 10.0.0.2 | 10.0.0.5 | TCP | ANY | DROP | SUCCESS |

L. Blocking TCP Address Topologi 3 (Linear)

Selanjutnya blocking aliran data berdasarkan Port TCP Topologi 3. Dan aturan yang akan di tambahkan adalah menutup komunikasi TCP pada h3 dengan destination port 2 pada s3, sehingga nantinya h4 dan h5 masih bisa komunikasi dengan h3, dengan menggunakan perintah PUT pada address bar, dan alamat yang ditujukan adalah sebagai berikut.

```
http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:3/table/0/flow/1
```

Berikut teks XML yang perlu ditambahkan menggunakan POSTMAN.

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <table_id>0</table_id>
  <id>1</id>
  <cookie_mask>255</cookie_mask>
  <match>
    <ethernet-match>
      <ethernet-type>
        <type>2048</type>
      </ethernet-type>
    </ethernet-match>
    <ip-match>
      <ip-protocol>6</ip-protocol>
    </ip-match>
    <ipv4-destination>10.0.0.3/32</ipv4-destination>
    <in-port>2</in-port>
  </match>
  <priority>20</priority>
  <hard-timeout>1200</hard-timeout>
  <cookie>8</cookie>
  <idle-timeout>3400</idle-timeout>
</flow>
```

Hasilnya sebagai berikut.



Gambar 16. Hasil dari Flow Berdasarkan TCP Address Topologi 3

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut.

Tabel 13. Pengujian Berdasarkan TCP Address Topologi 3

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | 00:00:00:00:00:02 | 10.0.0.1 | 10.0.0.2 | TCP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:03 | 00:00:00:00:00:05 | 10.0.0.3 | 10.0.0.5 | TCP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:05 | 00:00:00:00:00:02 | 10.0.0.5 | 10.0.0.2 | TCP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:02 | 00:00:00:00:00:05 | 10.0.0.2 | 10.0.0.5 | TCP | ANY | DROP | SUCCESS |

M. *Blocking ICMP Address Topologi 1 (Single)*

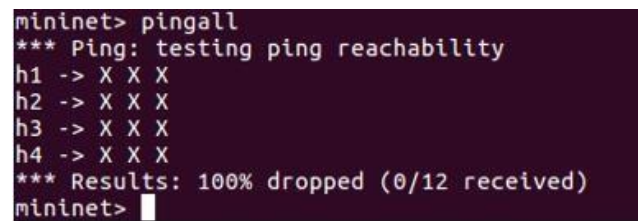
Selanjutnya *blocking* aliran data berdasarkan *ICMP* pada Topologi 1. Dan aturan yang akan di tambahkan adalah menutup semua komunikasi yang akan dilakukan, dengan menggunakan perintah *PUT* pada *address bar*, dan alamat yang ditujukan adalah sebagai berikut.

```
http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/1
```

Berikut teks *XML* yang perlu ditambahkan menggunakan *POSTMAN*.

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <priority>11</priority>
  <flow-name>Drop13</flow-name>
  <match>
    <ethernet-match>
      <ethernet-type>
        <type>2048</type>
      </ethernet-type>
    </ethernet-match>
    <ip-match>
      <ip-protocol>1</ip-protocol>
    </ip-match>
  </match>
  <id>1</id>
  <table_id>0</table_id>
  <instructions>
    <instruction>
      <order>0</order>
      <apply-actions>
        <action>
          <order>0</order>
          <drop-action/>
        </action>
      </apply-actions>
    </instruction>
  </instructions>
</flow>
```

Hasilnya sebagai berikut.



Gambar 17. Hasil dari Flow Berdasarkan ICMP Address Topologi 1

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut.

Tabel 14. Pengujian Berdasarkan ICMP Address Topologi 1

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|---------|----------|--------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | ANY | 10.0.0.1 | ANY | ICMP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | ANY | 10.0.0.2 | ANY | ICMP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:03 | ANY | 10.0.0.3 | ANY | ICMP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:04 | ANY | 10.0.0.4 | ANY | ICMP | ANY | DROP | SUCCESS |

N. *Blocking ICMP Address Topologi 2 (Tree)*

Selanjutnya *blocking* aliran data berdasarkan *ICMP* pada Topologi 2. Dan aturan yang akan di tambahkan adalah blok protokol *ICMP* pada s2, sehingga nantinya h1 h2 h3 h4 tidak bisa melakukan aliran data karena telah di *drop* saat melalui s2, dengan menggunakan perintah *PUT* pada *address bar*, dan alamat yang ditujukan adalah sebagai berikut.

```
http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/1
```

Berikut teks *XML* yang perlu ditambahkan menggunakan *POSTMAN*.


```
<flow
xmlns="urn:opendaylight:flow:inventory">
  <priority>25</priority>
  <flow-name>Drop13</flow-name>
  <match>
    <ethernet-match>
      <ethernet-type>
        <type>2048</type>
      </ethernet-type>
    </ethernet-match>
    <ip-match>
      <ip-protocol>1</ip-protocol>
    </ip-match>
  </match>
  <id>1</id>
  <table_id>0</table_id>
  <instructions>
    <instruction>
      <order>0</order>
      <apply-actions>
        <action>
          <order>0</order>
          <drop-action/>
        </action>
      </apply-actions>
    </instruction>
  </instructions>
</flow>
```

Hasilnya sebagai berikut.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 X X X X X X
h2 -> h1 X X X X X X
h3 -> X X h4 X X X X
h4 -> X X h3 X X X X
h5 -> X X X X h6 h7 h8
h6 -> X X X X h5 h7 h8
h7 -> X X X X h5 h6 h8
h8 -> X X X X h5 h6 h7
*** Results: 71% dropped (16/56 received)
mininet>
```

Gambar 18. Hasil dari Flow Berdasarkan ICMP Address Topologi 2

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut.

Tabel 15. Pengujian Berdasarkan ICMP Address Topologi 2

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:05 | 00:00:00:00:00:08 | 10.0.0.5 | 10.0.0.8 | ICMP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | 00:00:00:00:00:07 | 10.0.0.2 | 10.0.0.7 | ICMP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:03 | 00:00:00:00:00:05 | 10.0.0.3 | 10.0.0.5 | ICMP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:07 | 00:00:00:00:00:01 | 10.0.0.7 | 10.0.0.1 | ICMP | ANY | DROP | SUCCESS |

O. Blocking ICMP Address Topologi 3 (Linear)

Selanjutnya blocking aliran data berdasarkan ICMP pada Topologi 3. Dan aturan yang akan di tambahkan adalah akan blok protokol ICMP pada s1, sehingga semua aliran menuju h1 akan di drop, dengan menggunakan perintah PUT pada address bar, dan alamat yang ditujukan adalah sebagai berikut.

```
http://localhost:8181/restconf/config/opendaylight-
inventory:nodes/node/openflow:1/table/0/flow/1
```

Berikut teks XML yang perlu ditambahkan menggunakan POSTMAN.

```
<flow xmlns="urn:opendaylight:flow:inventory">
  <priority>25</priority>
  <flow-name>Drop13</flow-name>
  <match>
    <ethernet-match>
      <ethernet-type>
        <type>2048</type>
      </ethernet-type>
    </ethernet-match>
    <ip-match>
      <ip-protocol>1</ip-protocol>
    </ip-match>
  </match>
  <id>1</id>
  <table_id>0</table_id>
  <instructions>
    <instruction>
      <order>0</order>
      <apply-actions>
        <action>
          <order>0</order>
          <drop-action/>
        </action>
      </apply-actions>
    </instruction>
  </instructions>
</flow>
```

Hasilnya sebagai berikut.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X X
h2 -> X h3 h4 h5
h3 -> X h2 h4 h5
h4 -> X h2 h3 h5
h5 -> X h2 h3 h4
*** Results: 40% dropped (12/20 received)
mininet>
```

Gambar 19. Hasil dari Flow Berdasarkan ICMP Address Topologi 3

Peneliti juga melakukan beberapa pengujian lainnya, dan hasilnya dapat di lihat pada tabel berikut.

Tabel 15. Pengujian Berdasarkan ICMP Address Topologi 3

| Rule No | SRC MAC | DST MAC | SRC IP | DST IP | PROTOCOL | DST PORT | ACTION | RESULT |
|---------|-------------------|-------------------|----------|----------|----------|----------|--------|---------|
| 1 | 00:00:00:00:00:01 | ANY | 10.0.0.1 | ANY | ICMP | ANY | DROP | SUCCESS |
| 2 | 00:00:00:00:00:02 | 00:00:00:00:00:01 | 10.0.0.2 | 10.0.0.1 | ICMP | ANY | DROP | SUCCESS |
| 3 | 00:00:00:00:00:03 | 00:00:00:00:00:04 | 10.0.0.3 | 10.0.0.4 | ICMP | ANY | DROP | SUCCESS |
| 4 | 00:00:00:00:00:05 | 00:00:00:00:00:01 | 10.0.0.5 | 10.0.0.1 | ICMP | ANY | DROP | SUCCESS |

5. KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil implementasi dan analisis dari SDN berbasis openflow menggunakan controller opendaylight sebagai controller, maka dapat disimpulkan sebagai berikut:

1. Teknik yang digunakan dalam penerapan firewall pada arsitektur SDN adalah dengan cara menambahkan aturan rule baru pada flowtable yang ada di switch dan nantinya akan di sesuaikan dengan kebutuhan aliran data yang di inginkan. Dan teknologi yang digunakan adalah dengan menggunakan bahasa XML dan di terapkan pada switch virtual pada mininet

menggunakan *postman*, dan menerapkan kontrol *drop* (karena *defaultnya accept*).

2. Implementasi *firewall* pada *Software Defined Network* menggunakan *opendaylight* telah berhasil dilakukan karena semua aliran *flow* yang ingin di *drop* telah terlaksana dengan baik. Penerapan akses kontrol dalam mengatur *flow table firewall* pada penelitian ini menggunakan *IP Address, MAC Address, TCP, UDP, dan ICMP. Rule firewall* di *push* menggunakan *postman*.

5.2 Saran

Berikut adalah beberapa saran untuk kelanjutan penelitian yang dapat dilakukan, diantaranya:

1. Diuji lebih lanjut dengan topologi yang berbeda dengan jaringan yang semakin kompleks.
2. Melakukan penelitian efektivitas *firewall* dengan berbasis teknologi virtualisasi atau dalam lingkungan nyata dan bukan sekedar simulator.
3. Melakukan pengujian dengan metode *push rule firewall* dengan cara yang berbeda dan penambahan *rule* dengan menggunakan teknologi *web service*.
4. Diharapkan penelitian selanjutnya bisa untuk menciptakan antarmuka pengaturan *flow* yang berbasis *web*.

DAFTAR PUSTAKA

- [1] B. A. Linuwih, A. V., "Perancangan dan Analisis *Software Defined Network* pada Jaringan LAN: Penerapan dan Analisis Metode Penjaluran *Path Calculating* Menggunakan Algoritma Dijkstra," 2016.
- [2] A. R. Sudiyatmoko, S. N., "Analisis Performansi Perutingan *Link State* Menggunakan Algoritma Dijkstra pada Platform *Software Defined Network (SDN)*," 2016.
- [3] M. Syafrizal, "Pengantar Jaringan Komputer," Yogyakarta, 2005.
- [4] J. H. Green, "*Local Area Network A User's Guide For Business Professionals*," London: Scott, Foresman and Company, 1985.
- [5] M. Betts, Z. "Open Networking Foundation". SDN Architecture, 2014.
- [6] A. Kaur, V. S., "*Building L2-L4 Firewall using Software Defined Networking*," 2017.
- [7] R. Kartadie, E. U., "Prototipe *Infrastruktur Software Defined Network* dengan Protokol Openflow Menggunakan Ubuntu Sebagai Kontroller," 2014.
- [8] R. Kartadie, B. S., "Uji Performa Kontroller Floodlight dan Opendaylight Sebagai Komponen Utama Arsitektur *Software Defined Network*," 2015.
- [9] R. Tulloh, R. M., "Simulasi Virtual Local Area Network (VLAN) Berbasis *Software Defined Network (SDN)* Menggunakan POX Controller," 2015.
- [10] Noor, Juliansyah, "Metodologi Penelitian: Skripsi, Tesis, Disertasi, dan Karya Ilmiah." Jakarta: Kencana, 2017.



RANCANG BANGUN APLIKASI SISTEM INFORMASI SEKOLAH GO TO SCHOOL (GOS) BERBASIS WEB MENGGUNAKAN PHP

Chatur Nugroho¹, Suhendi²

^{1,2}Sistem Informasi, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
nugrohochatur@gmail.com, hendinf13@gmail.com

Abstract

The school information system is currently a reference for parents in registering their children in schools. The current system is still ineffective because the school information presented is fragmented, and comparisons must be made one by one. In terms of comfort and efficiency, it makes users uncomfortable. It tends to prefer manual methods, namely by coming directly to schools to find detailed information about the school of interest. Therefore, the authors conclude that the current system in presenting school information systems is still ineffective even though it can be accessed via the internet. Based on the previous view, the writer will discuss designing and implementing the Go To School (GoS) application based on the website using PHP with the waterfall method's software development. In collecting the data, the author uses the questionnaire and interview method and testing the application using the Likert scale and User Acceptance Testing (UAT). From the results of research conducted by the author, the results to be obtained are that the Go To School (GoS) application can be used smoothly and becomes a practical application in school information systems.

Keywords: Aplikasi Go To School (GoS), PHP, Waterfall, Skala Likert dan User Acceptance Testing (UAT)

Abstrak

Sistem informasi sekolah saat ini menjadi acuan bagi orang tua dalam mendaftarkan anaknya ke sekolah- sekolah. Sistem yang ada saat ini masih terbilang kurang efektif karena data informasi sekolah yang disajikan masih terpisah-pisah dan harus dilakukan perbandingan secara satu persatu. Dalam hal kenyamanan dan efisiensi hal itu sangat membuat pengguna tidak nyaman dan cenderung lebih memilih cara manual yaitu dengan datang langsung ke sekolah dalam rangka mencari informasi secara detail tentang sekolah yang diminati tersebut. Maka dari itu penulis menyimpulkan bahwa sistem yang ada saat ini dalam penyajian sistem informasi sekolah masih terbilang kurang efektif walaupun sudah bisa diakses melalui internet. Berlandaskan dari pandangan sebelumnya, penulis akan membahas mengenai proses perancangan dan implementasi Aplikasi Go To School (GoS) berbasis website menggunakan PHP dengan metode pengembangan perangkat lunak menggunakan metode waterfall dan didalam pengumpulan data penulis menggunakan metode kuesioner dan wawancara serta melakukan pengujian aplikasi menggunakan Skala Likert dan User Acceptance Testing (UAT). Dari hasil penelitian yang dilakukan penulis, hasil yang ingin didapat adalah Aplikasi Go To School (GoS) bisa digunakan dengan lancar dan menjadi aplikasi yang efektif dalam sistem informasi sekolah.

Kata kunci: Aplikasi Go To School (GoS), PHP, Waterfall, Skala Likert dan User Acceptance Testing (UAT)

1. PENDAHULUAN

Dalam rangka mengembangkan sumber daya manusia, bidang Pendidikan merupakan tempat yang sangat baik dalam pembinaan sumber daya manusia. Oleh sebab itu pendidikan perlu mendapat perhatian yang lebih serius baik itu oleh pemerintah, keluarga maupun pengelola pendidikan khususnya. Dengan perkembangan teknologi yang semakin pesat, sarana dan prasarana pendidikan pun harus mengikutinya, salah satunya dengan media komputerisasi yaitu internet.

Sekolah adalah tempat belajar para peserta didik yang menjadi sarana tumbuh kembangnya ilmu pengetahuan dari segala bidang terutama di bidang teknologi. Berdasarkan survey yang dilakukan oleh APJII (Asosiasi Penyelenggara Jasa Internet Indonesia) tentang Pertumbuhan Pengguna Internet Indonesia tahun 2017, menunjukkan dari tahun ke tahun peningkatan pengguna internet selalu mengalami kenaikan yang signifikan.

Berdasarkan data yang dilakukan oleh APJII, pengguna internet sangatlah banyak terutama di Indonesia. Manfaat dari internet salah satunya adalah dengan internet kita bisa mengetahui informasi apa saja, kapan saja dan dimana saja. Dengan adanya internet, banyak sekolah mulai berlomba-lomba memperkenalkan masing-masing sekolahnya dengan membuat *website* informasi guna mempromosikan sekolahnya secara global. Namun seiring perkembangnya waktu, hal ini dikira kurang efektif dikarenakan informasi yang dihasilkan terbilang lebih menyoroti dan memberikan nilai yang baik-baik untuk sekolah itu sendiri. Padahal kenyataannya bertolak belakang dengan informasi real di lapangan. Berdasarkan hal tersebut dibutuhkan sistem informasi yang bersifat selektif dan berimbang guna memberikan informasi yang terpercaya bagi semua orang terkhususnya orang tua yang ingin menyekolahkan anaknya

Menanggapi permasalahan yang terjadi, penulis mencoba merancang dan menerapkan aplikasi yang bertujuan memberikan informasi semua sekolah yang ada di regional tertentu dan memberikan informasi berdasarkan data real di lapangan secara mendetail. Tidak hanya itu, aplikasi yang akan dibuat oleh penulis juga mencoba memberikan fitur pendukung yang diharapkan dapat mempermudah orang tua untuk bisa mendaftarkan anaknya tanpa harus datang ke sekolah. Oleh karena itu diharapkan aplikasi yang akan diterapkan penulis ini dapat terealisasi dan memberikan dampak yang efektif bagi orang banyak terutama orang tua.

2. PEMBAHASAN

Penelitian ini menggunakan metode deskriptif dengan penarikan kesimpulan secara kuantitatif.

Penelitian deskriptif kuantitatif ini bertujuan untuk mendeskripsikan apa-apa yang saat ini berlaku. Di dalamnya terdapat upaya mendeskripsikan, mencatat, analisis dan menginterpretasikan kondisi yang sekarang ini terjadi atau ada. Dengan kata lain penelitian deskriptif kuantitatif ini bertujuan untuk memperoleh informasi-informasi mengenai keadaan yang ada.



Gambar 1. Rancangan Metode Menggunakan Waterfall

Pada perancangan Sistem Aplikasi Informasi berbasis Website ini, penulis menganalisis kebutuhan data dan

informasi apa saja yang akan digunakan dalam aplikasi GoS. Berikut hasil analisisnya:

- Data Sekolah**
Data sekolah merupakan data informasi sekolah yang terperinci yang terdiri dari nama sekolah, fasilitas yang dimiliki sekolah tersebut, prestasi yang dimiliki oleh sekolah tersebut, luas wilayah dan akreditasi yang didapat oleh sekolah tersebut.
- Data Survey Kebutuhan**
Data survey kebutuhan adalah hasil data survey yang disebar ke orang tua yang ingin mendaftarkan anaknya. Adapun isi dari survei mengenai kebutuhan apa saja yang dibutuhkan orang tua tentang sekolah yang harus ditampilkan dalam aplikasi yang penulis buat.

| No | Kesimpulan |
|----|---|
| 1 | Dari banyaknya orang tua yang mengisi kuesioner, mereka telah melihat dan merasakan sistem informasi yang menawarkan data tentang sekolah-sekolah |
| 2 | Masih banyak orang tua yang merasakan kurang efektif dan belum memuaskannya sistem yang ada saat ini |
| 3 | Berdasarkan hasil masukan orang tua, masih kurangnya informasi yang diberikan oleh sistem saat ini dalam memberikan informasi secara terperinci baik itu tentang sekolah, kegiatan-kegiatan, info beasiswa dan sebagainya |

Gambar 2. Kesimpulan Hasil Wawancara

Sumber data dalam penelitian yang dilakukan oleh penulis dalam tugas akhir ini adalah data primer dan data sekunder.

- Data primer** adalah data yang didapatkan secara langsung yaitu orang tua, guru, dan siswa. Proses untuk mendapatkan data primer ini melalui observasi (pengamatan) dan melakukan proses wawancara terhadap user yang terlibat langsung dengan penggunaan aplikasi.
- Data sekunder** adalah data yang didapat dari sumber yang sudah ada seperti yang dilakukan oleh APJII.

Data-data ini sangat berguna bagi penulis dalam melakukan proses penelitian tugas akhir ini.

Berdasarkan **Gambar 1** menggambarkan rancangan penelitian yang akan dibuat. Analisis ini diawali dengan mengumpulkan informasi kebutuhan dari keseluruhan sistem yang akan diterapkan ke dalam bentuk *software*. Hal ini sangat penting, mengingat *software* harus dapat berinteraksi dengan elemen-elemen yang lain seperti *hardware*, *database*, dsb. Tahap ini sering disebut dengan *Project Definition*.

Pada tahap ini peneliti melakukan observasi dan wawancara langsung dengan pihak sekolah yang peneliti kenal. Pada Observasi ini terfokus pada permasalahan yang menjadi keluhan dari pihak *user*. Setelah mendapatkan beberapa data, peneliti menyimpan data tersebut sebagai bahan acuan dalam tahap analisis.

Dalam penelitian ini adapun perangkat keras yang digunakan untuk membangun aplikasi GoS antara lain laptop dengan prosesor Intel-Core i5, *smartphone* minimal ram 1 Gb, kapasitas RAM 10 GB. Dari segi *software* yang digunakan : Xampp, Netbean, SQL Database, Browser Chrome, Sistem Operasi Windows, Framework Yii2, Photoshop, Paint.

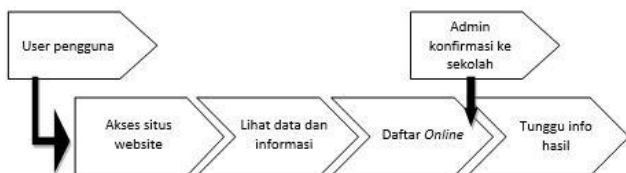
3. URAIAN PENELITIAN

3.1 Analisis Perancangan Sistem

Analisis sistem merupakan penguraian dari sistem informasi yang utuh kedalam bagian-bagian komponennya yang lebih rinci dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan yang ditemukan, kesempatan-kesempatan, hambatan yang terjadi, dan kebutuhan-kebutuhan yang diharapkan, sehingga dapat diusulkan penulis, untuk membangun sebuah aplikasi website yang menghasilkan informasi sekolah yang ada di regional tertentu secara terperinci dan real.

Perancangan Sistem Aplikasi Informasi Sekolah Berbasis *Website* ini merupakan alat bantu yang diperuntukan kepada orang tua yang ingin mencari informasi yang bagus untuk anaknya secara terperinci dan *real*. Tujuannya adalah mengurangi aktifitas fisik yang dilakukan orang tua ketika akan mendaftarkan anaknya ke sekolah yang diinginkan. Pada tahap perancangan ini penulis akan memulai dari tahap analisis dan beberapa tahapan dari tahap desain

3.2 Tahapan penelitian



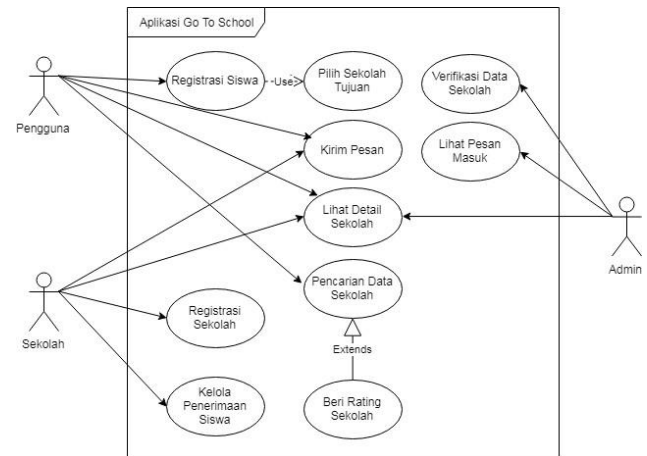
Gambar 3. Alur Sistem Usulan

Pada alur sistem di atas, terlihat bahwa proses yang berlaku ialah admin menginput info data sekolah yang memang sudah disetujui oleh kedua belah pihak. Setelah info sekolah sudah diinput, orang tua bisa mencari data dan informasi tentang sekolah di *website* informasi yang telah dibuat oleh penulis. Informasi yang ditawarkan dalam website yang dibuat oleh penulis diusahakan sangat detail dan real sesuai dengan fakta yang ada. Setelah orang tua sudah menemukan sekolah yang diminati, orang tua bisa akses sistem daftar online yang disediakan dalam situs *website* yang dibuat oleh penulis.

Ketika data pendaftaran *online* masuk ke dalam sistem, maka admin akan meneruskan ke sekolah terkait guna mendapatkan hasil dari sekolah terkait apakah data tersebut layak diterima di sekolah tersebut. Terakhir setelah hasil didapat maka admin akan mengirimkan info kepada orang tua bahwa data yang telah terdaftar diterima atau tidak di sekolah yang dipilih oleh orang tua tersebut.

3.3 Use Case Diagram

Pemodelan *Use Case Diagram* pada aplikasi GoS yang akan dirancang oleh penulis ini melibatkan satu buah aktor yaitu orang tua calon siswa yang ingin mendaftarkan anaknya sekolah dengan *Use Case* sebagai berikut:

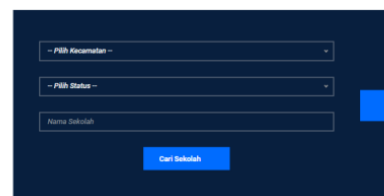


Gambar 4. Diagram Use Case

Pada *Use Case Diagram* yang penulis rancang, terlihat bahwa aktor utama yang terlibat dalam penggunaan aplikasi ini adalah orang tua calon siswa yang ingin mendaftarkan anaknya bersekolah. Selain itu ada juga sekolah yang berperan sebagai *client* yang menerima data siswa dan mempunyai wewenang untuk bisa menerima atau menolak data siswa yang telah mendaftar di aplikasi GoS.

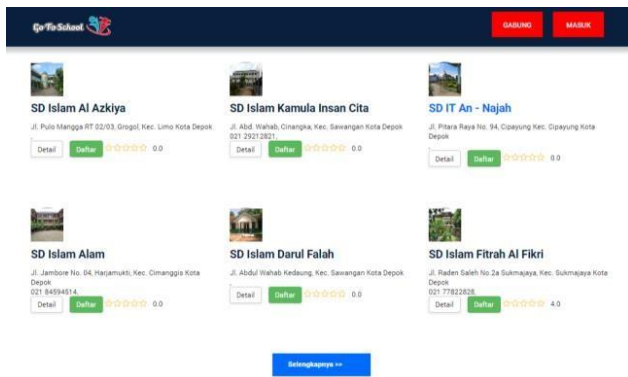
4. HASIL PENELITIAN

Hasil dari penelitian yang dilakukan oleh penulis bisa dilihat pada beberapa gambar aplikasi yang telah diimplementasikan dan hasil pengujian aplikasi menggunakan metode *Skala Linker* dan *User Acceptance Testing (UAT)*.



Gambar 5. Tampilan Halaman Utama

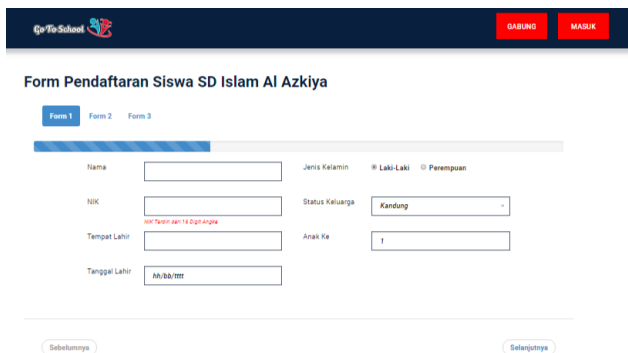
Halaman utama merupakan halaman yang menampilkan beranda website dari aplikasi Go To School. Dalam halaman ini menampilkan form pencarian sekolah yang ingin dicari oleh pengguna. Dalam form pencarian, pengguna bisa memasukkan data yang mereka inginkan seperti nama sekolah dan kecamatan. Dari data yang dimasukkan akan menampilkan sekolah-sekolah yang ada dalam sistem database aplikasi GoS.



Gambar 6. Tampilan Halaman Hasil Pencarian

Halaman hasil pencarian merupakan halaman yang menampilkan data pencarian yang diinginkan oleh pengguna. Hasil pencarian tergantung dari kata yang dimasukkan oleh pengguna yang akan disesuaikan dengan database yang ada dalam sistem aplikasi Go To School.

Dari hasil pencarian, pengguna bisa memilih daftar untuk mendaftarkan diri ke sekolah tersebut, ada pilihan untuk melihat detail dari data sekolah tersebut serta pengguna juga bisa memberikan hasil kepuasan mereka dengan memberikan bintang yang ada dalam aplikasi ini.



Gambar 7. Tampilan Halaman Daftar

Halaman daftar merupakan halaman yang menampilkan form pengisian untuk pengguna agar mereka bisa mendaftarkan diri dan anak mereka secara online yang kemudian akan disimpan dalam aplikasi database Go To School dan diteruskan ke sekolah yang mereka tuju.

Pada proses pengujian yang pertama dilakukan penulis dengan hasil sebagai output dokumentasi hasil pengujian yang dapat dijadikan bukti bahwa aplikasi sudah dapat diterima dan sudah memenuhi kebutuhan user.

| | |
|----|---------------|
| TS | Tidak Setuju |
| S | Setuju |
| SS | Sangat Setuju |

| Jawaban | Bobot |
|--------------------|-------|
| SS : Sangat Setuju | 3 |
| S : Setuju | 2 |
| TS : Tidak Setuju | 1 |

| No | Keuseioner Pengguna |
|----|---|
| 1 | Sistem informasi sekolah berbasis website dapat digunakan dengan mudah dan efektif |
| 2 | Sistem informasi sekolah dapat membantu memenuhi kebutuhan orang tua yang ingin mendaftarkan anaknya ke sekolah yang diinginkan |
| 3 | Proses sistem informasi berjalan baik dan <i>user friendly</i> ketika digunakan |
| 4 | Sudah efektif alur pendaftaran online yang ada pada sistem |
| 5 | Tidak ada bug yang terjadi pada sistem informasi ketika berjalan |

| Pertanyaan | Jawaban Pengguna | | | | | |
|------------|------------------|-----|---|-----|----|-----|
| | TS | % | S | % | SS | % |
| 1 | 0 | 0% | 6 | 60% | 4 | 40% |
| 2 | 0 | 0% | 5 | 50% | 5 | 50% |
| 3 | 1 | 10% | 6 | 60% | 3 | 30% |
| 4 | 4 | 40% | 5 | 50% | 0 | 0% |
| 5 | 6 | 60% | 4 | 40% | 0 | 0% |

| Pertanyaan | Jawaban Pengguna | | | Jumlah |
|------------|------------------|-----|------|--------|
| | TS*1 | S*2 | SS*3 | |
| 1 | 0 | 12 | 12 | 24 |
| 2 | 0 | 10 | 15 | 25 |
| 3 | 1 | 12 | 15 | 28 |
| 4 | 4 | 10 | 0 | 14 |
| 5 | 6 | 8 | 10 | 24 |

Gambar 8. Hasil Pengujian Skala Linker

- A. Analisa Pertanyaan Pertama
 Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 10 responden untuk pertanyaan pertama adalah 24. Nilai rata-ratanya adalah $24/10 = 2,4$. Prosentase nilainya adalah $2,4/3 \times 100\% = 80\%$.
- B. Analisa Pertanyaan Kedua
 Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 10 responden untuk pertanyaan kedua adalah 25. Nilai rata-ratanya adalah $25/10 = 2,5$. Prosentase nilainya adalah $2,5/3 \times 100\% = 83,33\%$.
- C. Analisa Pertanyaan Ketiga
 Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 10 responden untuk pertanyaan ketiga adalah 28. Nilai rata-ratanya adalah $28/10 = 2,8$. Prosentase nilainya adalah $2,8/3 \times 100\% = 93,33\%$.
- D. Analisa Pertanyaan Keempat
 Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 10 responden untuk pertanyaan keempat adalah 14. Nilai rata-ratanya adalah $14/10 = 1,4$. Prosentase nilainya adalah $1,4/3 \times 100\% = 46,66\%$.
- E. Analisa Pertanyaan Kelima
 Dari tabel di atas dapat dilihat bahwa jumlah nilai dari 10 responden untuk pertanyaan kelima adalah 24. Nilai rata-ratanya adalah $24/10 = 2,4$. Prosentase nilainya adalah $2,4/3 \times 100\% = 80\%$.

Pada proses pengujian yang kedua dilakukan penulis dengan hasil sebagai output dokumentasi hasil pengujian dan menjadi pendukung hasil dari pengujian yang pertama.

| No | Kelas Uji | Butir Uji | Hasil Pengujian | Catatan |
|----|---|--|-----------------|---------|
| 1 | Kelola masuk <i>user</i> siswa | Daftar siswa | Sukses | |
| | | Simpan data siswa | Sukses | |
| 2 | Kelola gabung <i>user</i> sekolah | Daftar sekolah | Sukses | |
| | | Simpan data sekolah | Sukses | |
| 3 | <i>User</i> pilih tombol detail | <i>View</i> data detail sekolah | Sukses | |
| 4 | <i>User</i> pilih tombol daftar | Masuk kedalam <i>form</i> daftar siswa | Sukses | |
| 5 | <i>User</i> pilih bintang | Memberikan <i>rating</i> kepuasan | Sukses | |
| 6 | <i>User</i> pilih pencarian kecamatan | <i>View</i> sekolah berdasarkan kecamatan | Sukses | |
| 7 | <i>User</i> pilih pencarian status | <i>View</i> sekolah berdasarkan status sekolah | Sukses | |
| 8 | <i>User</i> pilih pencarian nama sekolah | <i>View</i> sekolah berdasarkan nama sekolah | Sukses | |
| 9 | <i>User</i> mengisi <i>form</i> <i>feedback</i> | Memberikan masukan dan tanggapan | Sukses | |
| 10 | Admin kelola <i>verifikasi</i> | Admin terima atau tolak data yang masuk | Sukses | |
| 11 | Admin kelola <i>feedback</i> | Admin menanggapi <i>feedback</i> yang masuk | Sukses | |
| 12 | Sekolah kelola <i>verifikasi</i> | Sekolah terima atau tolak siswa yang masuk | Sukses | |

Gambar 9. Hasil Pengujian UAT

Dari hasil pengujian UAT oleh *user* dengan acuan dari 9 butir yang ada diatas menyimpulkan bahwa aplikasi Go To School dinyatakan 100% berjalan dengan lancar dan sukses. Adapun beberapa catatan yang diberikan oleh penguji akan masuk dalam tahap pengembangan dari aplikasi Go To School.

5. KESIMPULAN

Pada penelitian yang telah dilakukan oleh penulis, maka dapat disimpulkan bahwa aplikasi sistem informasi Go To School dapat menjawab rumusan masalah dengan kesimpulan sebagai berikut:

1. Perancangan aplikasi sistem informasi Go To School berbasis website dengan Framework PHP ini dapat berjalan dengan baik dan lancar serta dapat memenuhi kebutuhan informasi sekolah dan menjadi alat bantu dalam melaksanakan kegiatan penerimaan siswa baru berdasarkan hasil pengujian sistem menggunakan pengujian Skala Linkert dan *User Acceptance Testing* (UAT) yang dilakukan oleh pengguna lain.
2. Perancangan aplikasi sistem informasi Go To School berbasis *website* dengan Framework PHP ini sangat efektif dan menjadi alternatif solusi bagi orang tua dalam mencari informasi tentang sekolah secara detail dan mendaftarkan anaknya menggunakan fitur

pendaftaran online yang tersedia dalam aplikasi Go To School. (Masukkan berdasarkan hasil penilaian UAT)

Ucapan Terima Kasih

Terima kasih yang sebesar-besarnya kepada dosen pembimbing dan penguji STT Terpadu Nurul Fikri atas masukan dan dukungannya, dan juga kepada orang tua dan teman-teman yang telah membantu, sehingga penelitian ini bisa berjalan dengan baik.

DAFTAR PUSTAKA

- [1] S. Aditya, "Materi Tentang PHP," *Hypertext Processor* (PHP), 2014.
- [2] O. Makplus, "Definisi dan Pengertian Informasi Menurut Para Ahli," 2015.
- [3] Aditandespur, "Materi PHP," 2014, 11 2014. [Online]. Available: <http://aditandespur.blogspot.com/2014/11/materi-php.html>. [Diakses 24 11 2018]
- [4] O. Makplus, "Definisi dan Pengertian Informasi," 3 2015. [Online]. Available: <http://www.definisi-pengertian.com/2015/03/definisi-dan-pengertian-informasi.html>. [Diakses 11 2018]
- [5] L. Muhria, "Definisi dan Fungsi Sekolah," 2 2018. [Online]. Available: <https://www.lyceum.id/definisi-dan-fungsi-sekolah/>. [Diakses 11 2018]
- [6] P. Halawa, "Sistem Informasi Penerimaan Siswa Baru," 5 2015. [Online]. Available: <http://periamanhalawa19911206.blogspot.com/>. [Diakses 11 2018]
- [7] F. Galandi, "Metode Waterfall Definisi Tahapan," 9 2016. [Online]. Available: <http://www.pengetahuandanteknologi.com/2016/09/metode-waterfall-definisi-tahapan.html>. [Diakses 11 2018]
- [8] Metopel, "Pengertian Skala Likert Menurut Para Ahli," 4 2016. [Online]. Available: <https://www.afdhalilahi.com/2016/04/pengertian-skala-likert-menurut-para-ahli.html>. [Diakses 12 2018].
- [9] A. Trip, "*User Acceptance Test* (UAT) Apa Itu?," 8 2016. [Online]. Available: <https://adetrip.blogspot.com/2016/08/user-acceptance-test-uat-apa-itu.html>. [Diakses 12 2018]
- [10] M. Nazih, "Analisis Perancangan Prototipe Sistem Informasi Biaya Studi Mahasiswa Menggunakan Framework *Mobile* Onsen UI (Studi Kasus: STT Terpadu Nurul Fikri)," 2017.



EVALUASI KINERJA HDFS SEBAGAI INFRASTRUKTUR PEMBANGUNAN BIG DATA

Yunita Surahman¹, Henry Saptono²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
yunitasurahman1@gmail.com , henry@nurulfikri.co.id

Abstract

This research is proposing the implementation and analysis of HDFS as a big data development infrastructure. In this study, there are steps to implement HDFS and testing to determine the factors used to determine HDFS performance and determine HDFS performance as a big data development infrastructure. There are five tests, namely, testing the read and write execution time using the TestDFSIO benchmarks application, NNbench, changing blocksize, deleting files, and testing availability. The results obtained from testing on HDFS as a big data development infrastructure are the factors that determine the performance of the HDFS system design are the file size and block size. And also stated that the larger the size of the block size makes the execution time faster.

Keywords: Performance, HDFS, TestDFSIO, NNbench, Blocksize

Abstrak

Penelitian ini diajukan untuk implementasi dan analisis HDFS sebagai infrastruktur pembangunan *big data*. Pada penelitian ini terdapat langkah-langkah yang dilakukan dalam implementasi HDFS dan pengujian untuk mengetahui faktor-faktor yang digunakan untuk menentukan kinerja HDFS dan mengetahui kinerja HDFS sebagai infrastruktur pembangunan *big data*. Terdapat 5 pengujian yaitu, pengujian waktu eksekusi *read* dan *write* menggunakan aplikasi *benchmarks* TestDFSIO, NNbench, perubahan *blocksize*, penghapusan file, dan pengujian *availability*. Hasil yang didapat dari pengujian pada HDFS sebagai infrastruktur pembangunan *big data* ini yaitu Faktor-faktor yang menentukan kinerja dari rancangan sistem HDFS adalah ukuran file dan ukuran *blocksize*. Dan juga menyatakan bahwa semakin besar ukuran dari *blocksize* membuat waktu eksekusinya menjadi lebih cepat.

Kata kunci: Kinerja, HDFS, TestDFSIO, NNbench, Blocksize

1. PENDAHULUAN

Seiring perkembangan teknologi informasi yang cepat memberikan konsekuensi pertumbuhan dan peningkatan jumlah data, di dalam sebuah riset dari *Vcould News* mencatat pada tahun 2015, pertumbuhan data statistik per hari mencapai 2,5 quintillion (10 pangkat 18) byte atau sekitar Giga Byte (GB) perdetik. Pada tahun 2018 pertumbuhan data ini diproyeksikan mencapai 50 ribu GB perdetik [1]. Dengan angka sebesar ini membuat suatu perusahaan atau organisasi mencari cara untuk menyimpan serta mengolah data yang banyak tersebut. Data yang banyak dapat diolah serta dianalisis sehingga melahirkan sebuah *value*. Nilai yang merupakan hasil analisis tersebut dapat dijadikan sebuah pertimbangan dalam pengambilan keputusan disuatu perusahaan atau organisasi.

Pertumbuhan data yang besar menyebabkan teknologi *big data* yang ada saat ini menjadi solusi untuk menyimpan serta mengolah data yang banyak tersebut, maka dibutuhkan juga suatu sistem yang mampu manajemen data-data tersebut hingga menghasilkan suatu *value*. *Project open source* bernama Hadoop hadir sebagai salah satu sistem yang mampumanajemen data secara terdistribusi

Pada tahun 2005, Doug Cutting dan Mike Cafarella menciptakan Hadoop saat bekerja pada perusahaan Yahoo! Hadoop adalah inspirasi yang didapatkan dari mainan gajah kecil berwarna kuning milik anak Doug Cutting. Hadoop versi 0.1.0 akhirnya rilis pada bulan April 2006, sampai versi terakhir Hadoop yang rilis pada Maret 2017 adalah Apache Hadoop 2.8. Pada versi terbaru ini, layanan yang diberikan Hadoop juga termasuk untuk HDFS (*Hadoop*

Distributed File System), Yarn (*Yet Another Resource Negotiator*) dan MapReduce [2].

Hadoop sendiri menawarkan teknologinya yaitu HDFS (Hadoop Distributed File System) dimana data-data yang ada akan di distribusikan dalam bentuk *block-block* data untuk disimpan dalam setiap media penyimpanan (*node*) dalam sebuah *cluster*. Penggunaan Hadoop di era *cloud computing* seperti saat ini sangatlah dibutuhkan karena hadoop sendiri menyebdiakan *file system* yang dapat menyimpan data dalam ukuran besar secara *scalable*. MapReduce juga termasuk kedalam project Apache Hadoop yang dimana HDFS dan MapReduce yang berguna untuk menyelesaikan permasalahan data besar berbasis *Java* dan *open source*. Hadoop *file system* juga mengklaim dirinya memiliki fitur toleransi tinggi, ketersediaan tinggi, keandalan data, replikasi, *scalability*, dan *distributed storage*.

1.1 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, rumusan masalah dari penelitian ini adalah:

1. Langkah-langkah apa saja yang dilakukan dalam implementasi HDFS sebagai infrastruktur pembangunan Big Data?
2. Faktor-faktor apa saja yang menentukan kinerja dari rancangan sistem HDFS yang diteliti?
3. Bagaimanakah kinerja HDFS sebagai infrastruktur pembangunan Big Data?

1.2 Tujuan Penelitian

1. Faktor-faktor yang digunakan untuk menentukan kinerja HDFS.
2. Melakukan penelitian atau pengukuran unjuk kerja HDFS.

1.3 Manfaat Penelitian

1. Memahami langkah-langkah yang dilakukan.
2. Mengetahui faktor-faktor yang menentukan kinerja HDFS.
3. Menghasilkan suatu karya tulis yang bisa menjadi rujukan didalam mengimplementasikan HDFS sebagai infrastruktur pembangunan *big data*.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini adalah:

1. Lingkungan penelitian adalah lingkungan Virtualisasi, dimana sistem yang dipasang perangkat Hadoop adalah *Virtual Machine*.
2. Sistem HDFS ini diimplementasikan dalam konsep sistem terdistribusi dengan jumlah *node* adalah 3.
3. Penilaian kinerja peneliti menggunakan peralatan atau *tools* yang sudah ada atau tidak membuatnya sendiri.
4. *Platform* Sistem Operasi yang digunakan Homogen

(Linux).

5. Faktor yang dianalisa untuk menentukan kinerja HDFS adalah waktu eksekusi faktor I/O atau *read & write* menggunakan aplikasi *benchmark* Hadoop NNbench & TestDFSIO, pengaruh *block size* pada HDFS, dan pengujian *availability* atau ketersediaan data.

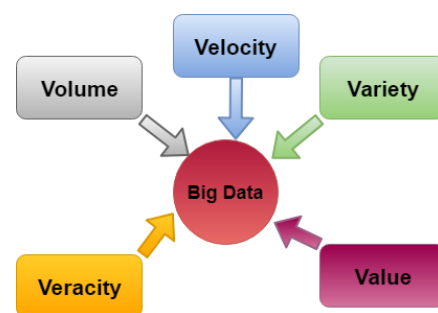
2. LANDASAN TEORI

2.1 Big Data

Big data adalah data yang sudah sangat sulit untuk dikoleksi, disimpan, dikelola maupun dianalisa dengan menggunakan sistem database biasa karena volumenya yang terus berlipat[3]. Dapat disimpulkan bahwa big data adalah data dengan ciri berukuran sangat besar, sangat variatif, sangat cepat pertumbuhannya dan mungkin tidak terstruktur yang perlu diolah khusus dengan teknologi inovatif sehingga mendapatkan informasi yang mendalam dan dapat membantu pengambilan keputusan yang lebih baik serta tidak dapat di proses menggunakan alat tradisional biasa dan harus menggunakan cara dan alat baru untuk mengolah data sehingga menjadi nilai.

Big data memiliki 5 karakteristik yaitu:

1. *Volume* yaitu sejumlah data besar yang dihasilkan perdetiknya.
2. *Velocity* yaitu kecepatan data baru yang dihasilkan dan pertumbuhan serta perubahannya.
3. *Variety* yaitu mengacu pada berbagai jenis data.
4. *Veracity* yaitu kebenaran serta keakuratan dari suatu data.
5. *Value* yaitu kemampuan kita mengubah data yang ada menjadi sebuah nilai.



Gambar 1. Karakteristik Big Data

2.2 Kinerja

Menurut kamus besar bahasa Indonesia, kinerja adalah kemampuan kerja. Kinerja suatu *software* bisa menurun karena suatu hal, dan kinerja suatu *software* juga bisa dioptimalkan. Untuk mengoptimalkan kinerja suatu *software* kita harus mengetahui faktor-faktor yang dapat mengoptimalkannya

2.3 Hadoop

Hadoop atau Apache Hadoop adalah *software open source* yang ditulis menggunakan bahasa *Java* untuk dijalankan

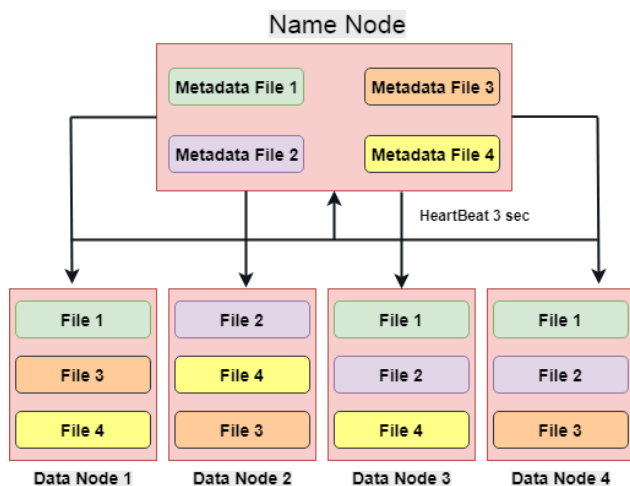
secara terdistribusi dan *scalable*. Hadoop dibangun berdasarkan algoritma MapReduce dari Google Inc. Hadoop menyediakan penyimpanan besar-besaran untuk jenis data, serta kemampuan untuk menangani tugas-tugas atau pekerjaan secara bersamaan.

Hadoop sendiri sejak tanggal 23 Januari 2008 telah menjadi proyek tingkat atas yang dimiliki lingkungan Apache Software Foundation dan dikembangkan secara terbuka oleh komunitas kontributor secara global. Owen O'Malley adalah orang yang diikuti sertakan ke proyek Hadoop pada Maret 2006, dan hadoop 0.1.0 dirilis pada April 2006.

2.4 Hadoop File System (HDFS)

HDFS adalah sistem file yang dirancang untuk menyimpan file yang sangat besar dengan pola akses data streaming, berjalan di cluster pada perangkat keras komoditas [4]. Hadoop File System atau HDFS adalah sebuah file terdistribusi, skalabel, dan portable yang ditulis menggunakan bahasa pemrograman Java untuk kerangka Hadoop. HDFS menyimpan file besar (biasanya dalam kisaran gigabytes hingga terabytes) di dalam beberapa mesin. HDFS akan melakukan proses pemecahan file besar menjadi bagian-bagian lebih kecil yang kemudian akan didistribusikan ke *cluster-cluster* dari komputer. Data yang di potong menjadi lebih kecil disebut *block* dan ukuran *block* dapat diatur sesuai kebutuhan.

Setiap data atau file yang kita simpan di HDFS akan selalu memiliki lebih dari satu *copy* data atau file tersebut. Hal tersebut di namakan Replication Factor (RF) dimana satu file disimpan di 3 data *node* sehingga jika ada satu DataNode yang rusak, maka DataNode yang lainnya bisa memberikan filenya.



Gambar 2. HDFS

2.5 Block Size

Block size adalah potongan-potongan data yang tersimpan di dalam HDFS [5]. Ukuran block size pada HDFS umumnya adalah 128 MB. Dengan begitu, file HDFS dipotong menjadi potongan 128 MB, dan jika

memungkinkan setiap potongan akan berada dalam DataNode yang berbeda.

2.6 MapReduce

Kerangka kerja perangkat lunak untuk menulis aplikasi dengan mudah yang memproses sejumlah besar data (kumpulan data multi-terabyte) secara paralel pada kelompok besar (ribuan node) perangkat keras komoditas dengan cara yang dapat diandalkan dan toleran kesalahan [6].

2.7 Yet Another Resource Negotiator (YARN)

YARN adalah platform sumber daya generik untuk mengelola sumber daya dalam sebuah typical cluster. YARN diperkenalkan pada Hadoop 2.0, yang merupakan kerangka kerja pemrosesan sumber terbuka yang terdistorsi dari Apache Software Foundation. YARN juga diciptakan oleh nama MapReduce 2.0. ini sejak apache hadoop mapreduce telah dirancang ulang apache hadoop YARN.YARN sebagai bahan komputasi generik untuk mendukung MapReduce dan paradigma aplikasi lainnya dalam cluster hadoop yang sama.

2.8 Java Development Kit (JDK)

Java Development Kit (JDK) adalah implementasi salah satu Platform Java, Edisi Standar, Platform Java, Edisi Enterprise, atau platform Micro Edition yang dirilis oleh Oracle Corporation dalam bentuk produk biner yang ditujukan untuk Pengembang Java di Solaris, Linux, MacOS atau Windows.

2.9 Secure Shell (SSH)

Secure Shell adalah sebuah protokol jaringan kriptografi untuk komunikasi data yang aman, login antarmuka baris perintah, perintah eksekusi jarak jauh, dan layanan jaringan lainnya antara dua jaringan komputer. Semua komunikasi antara klien dan server dienkripsi dengan aman dan dilindungi dari modifikasi [7].

2.10 TestDFSIO

TestDFSIO adalah aplikasi *benchmark* hadoop yang berfungsi menguji kinerja I/O pada HDFS dan juga ini sangat membantu untuk tugas-tugas seperti stress testing HDFS, untuk menemukan bottleneck kinerja di jaringan [8].

2.11 NameNode Benchmarks (NNBENCH)

NNBench adalah aplikasi *benchmark* hadoop yang memiliki tujuan untuk memeriksa konfigurasi NameNode pada multinode cluster hadoop. Pengujian ini dilakukan dengan menggunakan pekerjaan MapReduce sebagai cara yang nyaman untuk membacadan menulis file secara paralel. Tolok ukur nbench berguna untuk menguji-beban namenode. Benchmark ini mensimulasikan volume tinggi permintaan manipulasi file terhadap HDFS untuk "stress-test" kemampuan namenode untuk mengelola HDFS [9].

2.12 Throughput

Throughput adalah kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data. Biasanya throughput selalu dikaitkan dengan bandwidth, karena throughput bisa juga disebut dengan bandwidth. Akan tetapi bandwidth bersifat *fix*, sedangkan throughput sifatnya adalah dinamis tergantung *traffic* yang sedang terjadi [10].

3. ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan Hardware

Tabel 1. Analisis Kebutuhan Hardware

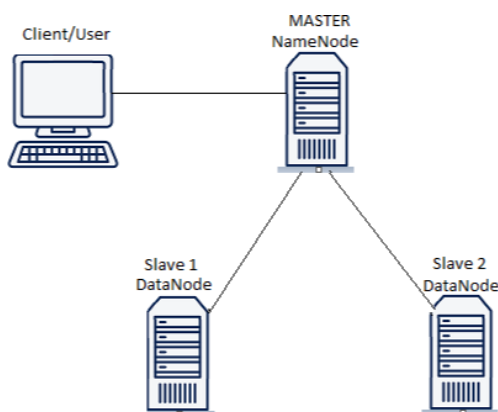
| Spesifikasi | NameNode | DataNode1 | DataNode2 |
|------------------|--------------------|--------------------|--------------------|
| Processor | One Core | One Core | One Core |
| RAM | 1.5 GB | 1.5 GB | 1.5 GB |
| System Operation | Linux Ubuntu 16.04 | Linux Ubuntu 16.04 | Linux Ubuntu 16.04 |

3.2 Analisis Kebutuhan Software

Dalam penelitian analisis kinerja HDFS sebagai infrastruktur pembangunan *big data*, peneliti menggunakan perangkat lunak (*software*) yang akan digunakan:

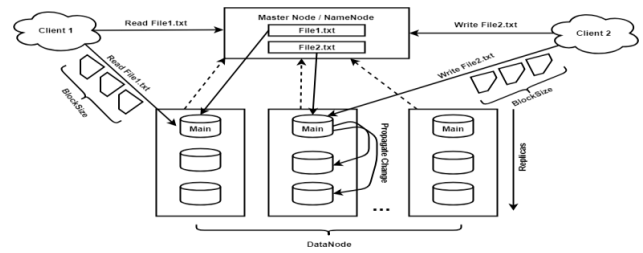
- VirtualBox
- Java Development Kit (JDK) 1.8
- Secure Shell (SSH)
- OS Linux Ubuntu 16.04
- Hadoop 2.7.2
- MapReduce/YARN

3.3 Perancangan Sistem



Gambar 3. Perancangan Sistem

3.4 Rancangan Pengujian



Gambar 4. Rancangan Pengujian

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

1. Instalasi Java pada NameNode dan DataNode dengan mendownload pada websitenya. Lalu ekstrak dengan perintah: `sudo tar xvzf jdk-8u191-linux-x64.tar.gz`
2. Membuat User Hadoop yang akan dijadikan user untuk menjalankan hadoop di NameNode dan DataNode, dengan perintah:

```
$sudo su
$sudo addgroup hadoop
$sudo adduser hduser
$sudo adduser hduser hadoop
```

3. Menambah alamat IP pada konfigurasi NameNode dan Datanode dengan perintah:

```
$sudo nano /etc/hosts
```

4. Melakukan instalasi ssh dengan perintah:

```
$sudo apt-get install openssh-server
```

5. Disable IPV6 dengan perintah: `$sudo nano /etc/sysctl.conf`

6. Sebelum melakukan instalasi, download hadoop terlebih dahulu kemudian dapat melakukan instalasi hadoop-2.7.2 dengan perintah:

```
$cd /usr/local
$sudo tar -xvf hadoop-2.7.2.tar.gz
$sudo ln -s hadoop-2.7.2 hadoop
```

7. Karena hadoop berplatform aplikasi java maka penulis menambahkan konfigurasi pada aplikasi hadoop:

```
$sudo nano
/usr/local/hadoop/etc/hadoop/hadoop-env.sh
#export
JAVA_HOME=/usr/local/java/jdk1.8.0_191
```

8. Kemudian konfigurasi hadoop untuk merubah parameter dalam memproses dokumen.

```
$ sudo nano
```

```

/usr/local/hadoop/etc/hadoop/core-site.xml
$ sudo nano
/usr/local/hadoop/etc/hadoop/hdfs-site.xml
$ sudo nano
/usr/local/hadoop/etc/hadoop/yarn-site.xml
$ sudo nano
/usr/local/hadoop/etc/hadoop/mapred-site.xml

```

9. Format NameNode dengan perintah:

```
$hdfs namenode -format
```

10. Menjalankan hadoop HDFS:

```
$ start-dfs.sh
```

11. Menjalankan YARN dengan perintah:

```
$ start-yarn.sh
```

4.2 Pengujian

A. Menjalankan Aplikasi TestDFSIO

```

$ hadoop jar
/usr/local/hadoop/share/hadoop/mapr
educe/hadoop-mapreduce-client-
jobclient-2.7.2-tests.jar TestDFSIO
-write -nrFiles 1 -fileSize 500
$ hadoop jar
/usr/local/hadoop/share/hadoop/mapr
educe/hadoop-mapreduce-client-
jobclient-2.7.2-tests.jar TestDFSIO
-read -nrFiles 1 -fileSize 500

```

B. Menjalankan Aplikasi NNBeach

```

$hadoop jar
/usr/local/hadoop/share/hadoop/mapr
educe/hadoop-mapreduce-client-
jobclient-2.7.2-tests.jar nnbench -
operation create_write -
numberOfFiles500
$hadoop jar
/usr/local/hadoop/share/hadoop/mapr
educe/hadoop-mapreduce-client-
jobclient-2.7.2-tests.jar nnbench -
operation open_read -
numberOfFiles500

```

C. Pengubahan *Blocksize*

```

$ start=$(date +%s%N/1000000) &&
hadoop fs -D dfs.block.size=
201326592 -put Desktop/perc1
/pengujian1 && echo "it took $($
date +%s%N/1000000) - $start)

```

miliseconds”

D. Penghapusan File

```

$ start=$(date +%s%N/1000000) &&
hadoop fs -rmr -R
/pengujian1/perc1 && echo "it
took $($date +%s%N/1000000) -
$start) miliseconds”

```

E. Pengujian *Availability*

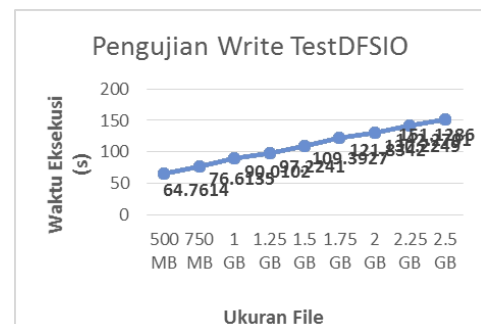
Pengujian *availability* atau ketersediaan data menggunakan skenario dimana salah satu datanode di non-aktifkan, dan namenode melakukan penulisan (*write*) ke HDFS. Pada pengujian ini, peneliti menjadikan datanode2 sebagai datanode yang di-non aktifkan saat menyalin atau menuliskan file ke dalam HDFS. Setelah namenode selesai melakukan penulisan pada HDFS, datanode2 yang di non-aktifkan akan diaktifkan kembali, lalu dicek apakah datanode2 tersebut bisa mengakses atau terdapat data yang namenode sudah tuliskan atau salin dari *local* komputer ke HDFS.

4.3 Hasil Pengujian

A. Pengujian menggunakan TestDFSIO

Tabel 2. Rata-Rata Hasil Pengujian Waktu Eksekusi *Write* TestDFSIO

| Ukuran Data | Rata-Rata Waktu |
|-------------|-----------------|
| 500 MB | 64.7614 |
| 750 MB | 76.6135 |
| 1 GB | 90.0102 |
| 1.25 GB | 97.2241 |
| 1.5 GB | 109.3927 |
| 1.75 GB | 121.8342 |
| 2 GB | 130.2249 |
| 2.25 GB | 142.2701 |
| 2.5 GB | 151.1286 |



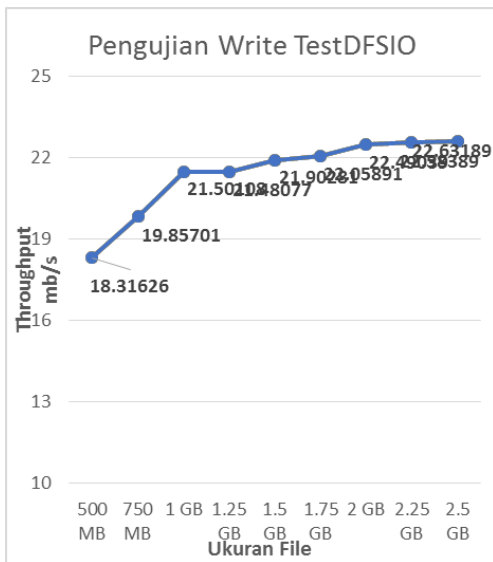
Gambar 5. Grafik Pengujian Waktu Eksekusi *Write* TestDFSIO

Dari hasil pengujian waktu eksekusi *write* TestDFSIO, menghasilkan waktu eksekusi (ms) yang kemudian peneliti ambil rata-rata waktu eksekusi tersebut. Dari tabel 2 dan gambar grafik 5 dapat dilihat bahwa terdapat peningkatan waktu dalam eksekusi *write* file pada tiap size file yang berbeda dan dapat disimpulkan dari hasil percobaan, bahwa

semakin besar data yang diinputkan atau dituliskan kedalam HDFS maka dibutuhkan waktu eksekusi yang lebih banyak untuk menyelesaikannya atau mendistribute file tersebut kedalam multinode cluster.

Tabel 3. Rata-Rata Hasil Pengujian Throughput Write TestDFSIO

| Ukuran Data | Rata-Rata Throughput |
|-------------|----------------------|
| 500 MB | 18.31626 |
| 750 MB | 19.85701 |
| 1 GB | 21.50108 |
| 1.25 GB | 21.48077 |
| 1.5 GB | 21.90281 |
| 1.75 GB | 22.05891 |
| 2 GB | 22.49039 |
| 2.25 GB | 22.58389 |
| 2.5 GB | 22.63189 |

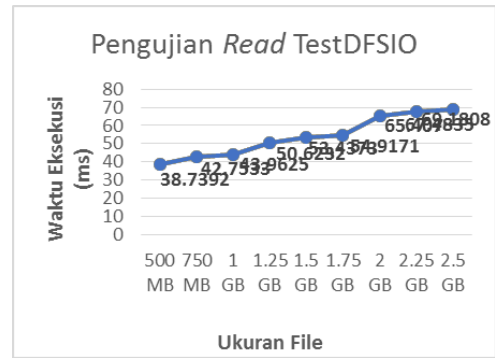


Gambar 6. Grafik Pengujian Throughput Write TestDFSIO

Dari hasil pengujian throughput write TestDFSIO menghasilkan throughput yang kemudian peneliti ambil rata-rata waktu throughput tersebut. Throughput merupakan rate atau kecepatan ideal proses transfer data. Dari tabel 3 dan gambar grafik 6 dapat dilihat bahwa terdapat penurunan dan peningkatan throughput dalam eksekusi write file pada tiap size file yang berbeda.

Tabel 4. Rata-Rata Hasil Pengujian Waktu Eksekusi Read TestDFSIO

| Ukuran Data | Rata-Rata Waktu |
|-------------|-----------------|
| 500 MB | 38.7392 |
| 750 MB | 42.7533 |
| 1 GB | 43.9625 |
| 1.25 GB | 50.6232 |
| 1.5 GB | 53.4373 |
| 1.75 GB | 54.9171 |
| 2 GB | 65.4070 |
| 2.25 GB | 67.4835 |
| 2.5 GB | 69.1808 |

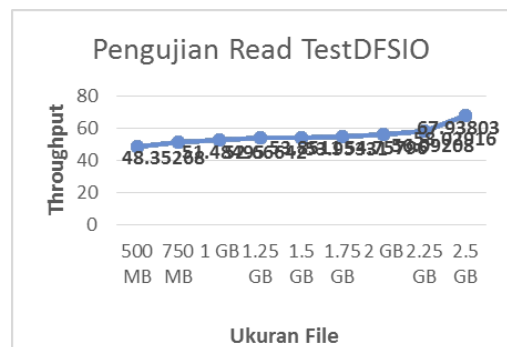


Gambar 7. Grafik Pengujian Eksekusi Read TestDFSIO

Dari pengujian waktu eksekusi read TestDFSIO, menghasilkan waktu eksekusi (ms) yang kemudian peneliti ambil rata-rata waktu eksekusi tersebut. Dari tabel 4 dan gambar grafik 7 dapat dilihat bahwa terdapat peningkatan waktu dalam eksekusi read file pada tiap size file yang berbeda dimana semakin banyak jumlah file yang di create maka membutuhkan waktu yang lebih banyak.

Tabel 5. Rata-Rata Hasil Pengujian Waktu Throughput Read TestDFSIO

| Ukuran Data | Rata-Rata Throughput |
|-------------|----------------------|
| 500 MB | 48.35268 |
| 750 MB | 51.48495 |
| 1 GB | 52.66642 |
| 1.25 GB | 53.85110 |
| 1.5 GB | 53.95331 |
| 1.75 GB | 54.75796 |
| 2 GB | 56.09208 |
| 2.25 GB | 58.02916 |
| 2.5 GB | 67.93803 |



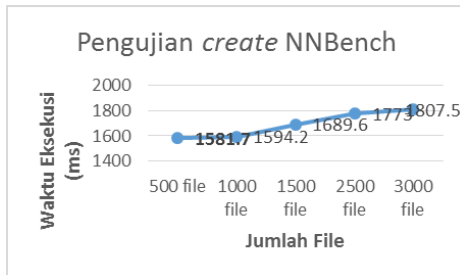
Gambar 8. Grafik Hasil Pengujian Throughput Read TestDFSIO

Dari hasil pengujian throughput read TestDFSIO, menghasilkan throughput yang kemudian peneliti ambil rata-rata waktu throughput tersebut. Throughput merupakan rate atau kecepatan ideal proses transfer data. Dari tabel 5 dan gambar grafik 8 dapat dilihat bahwa terdapat penurunan dan peningkatan throughput dalam eksekusi read file pada tiap size file yang berbeda.

B. Pengujian Menggunakan Aplikasi Benchmarks NNbench

Tabel 6. Rata-Rata Hasil Pengujian *Create* NNbench

| Ukuran data/file | Rata-Rata Waktu |
|------------------|-----------------|
| 500 File | 1.5817 |
| 1000 File | 1.5942 |
| 1500 File | 1.6896 |
| 2500 File | 1.7730 |
| 3000 File | 1.8075 |

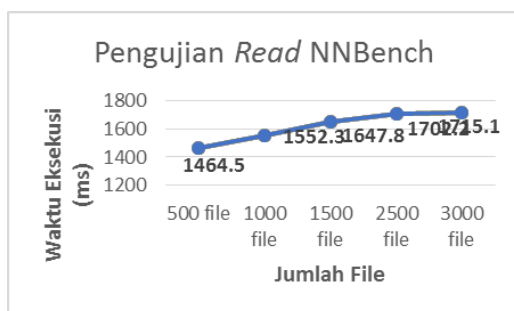


Gambar 9. Grafik Hasil Pengujian *Create* NNbench

Dari hasil pengujian *create* NNbench, kemudian peneliti ambil rata-rata waktu eksekusi tersebut. Dari tabel 6 dan gambar grafik 9 dapat dilihat bahwa terdapat peningkatan waktu dalam eksekusi *create* file pada tiap jumlah file yang berbeda, dimana semakin banyak file yang *create* atau *write* maka akan menyebabkan bertambahnya atau semakin lamanya waktu eksekusinya.

Tabel 7. Rata-Rata Hasil Pengujian *Read* NNbench

| Ukuran data/file | Rata-Rata Waktu |
|------------------|-----------------|
| 500 File | 1.4645 |
| 1000 File | 1.5523 |
| 1500 File | 1.6478 |
| 2500 File | 1.7022 |
| 3000 File | 1.7151 |



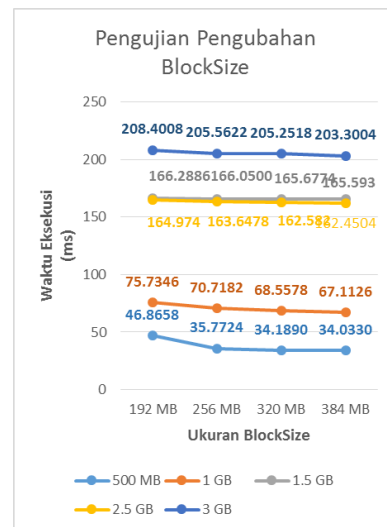
Gambar 10. Grafik Hasil Pengujian *Read* NNbench

Dari hasil pengujian *read* NNbench, kemudian peneliti ambil rata-rata waktu eksekusi tersebut. Dari tabel 7 dan gambar grafik 10 tersebut dapat dilihat bahwa terdapat penurunan serta peningkatan waktu dalam eksekusi *read* file pada tiap jumlah file yang berbeda. Peneliti menyimpulkan dari hasil percobaan, bahwa semakin banyak file yang dieksekusi *read* maka dibutuhkan waktu eksekusi yang lebih banyak untuk menyelesaikannya.

C. Pengujian Pengubahan *BlockSize*

Tabel 8. Rata-Rata Hasil Pengujian *Blocksize*

| Ukuran Data | Blocksize 192 MB | Blocksize 256 MB | Blocksize 320 MB | Blocksize 384 MB |
|-------------|------------------|------------------|------------------|------------------|
| 500 MB | 46.8658 | 35.7724 | 34.1890 | 34.0330 |
| 1 GB | 75.7346 | 70.7182 | 68.5578 | 67.1126 |
| 1.5 GB | 166.2886 | 166.0500 | 165.6774 | 165.693 |
| 2.5 GB | 164.974 | 163.6478 | 162.582 | 162.4504 |
| 3 GB | 208.4008 | 205.5622 | 205.2518 | 203.3004 |



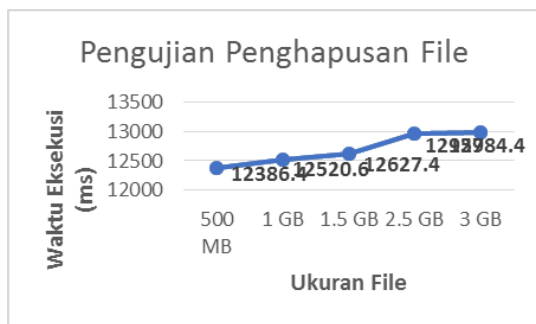
Gambar 11. Grafik Hasil Pengujian *Blocksize*

Dari tabel 8 dan gambar grafik 11 pengujian pengubahan *blocksize* peneliti menyimpulkan dengan pengubahan *blocksize* dapat mempercepat waktu eksekusi penulisan file atau data ke HDFS. File yang berukuran 500 MB akan dibagi atau *split* menjadi beberapa ukuran *block*, jika ukuran masing-masing *block* 192 MB maka akan menghasilkan 3 *block* dimana *block* 1 berukuran 192 MB, *block* 2 berukuran 192 MB, dan *block* 3 berukuran 116 MB. Jika ukuran *block*nya menjadi 384 MB, maka akan menghasilkan 2 *block* dimana *block* 1 berukuran 384 MB dan *block* 2 berukuran 116 MB. Jika jumlah *block* semakin sedikit maka akan mengurangi kerja dari namenode.

D. Pengujian Penghapusan Data

Tabel 9. Rata-Rata Hasil Pengujian Penghapusan Data

| Ukuran data/file | Rata-Rata Waktu |
|------------------|-----------------|
| 500 MB | 12.3864 |
| 1 GB | 12.5206 |
| 1.5 GB | 12.6274 |
| 2.5 GB | 12.9570 |
| 3 GB | 12.9844 |



Gambar 12. Grafik Hasil Pengujian Penghapusan File

Dari tabel 9 dan gambar grafik 12 menunjukkan peningkatan waktu eksekusi penghapusan file pada ukuran yang berbeda. Peneliti dapat menyimpulkan bahwa semakin besar ukuran file, maka akan membutuhkan waktu yang lebih lama untuk melakukan penghapusan file pada HDFS.

E. Pengujian *Availability*

Pada pengujian *availability* data ini peneliti menarik kesimpulan bahwa HDFS tingkat ketersediaan datanya baik, karena telah dilakukan percobaan menonaktifkan salah satu datanode yaitu datanode2. Kerika namenode menginputkan atau menuliskan data ke HDFS, setelah selesai namenode menuliskan data ke HDFS, maka datanode2 diaktifkan kembali. Lalu dicek atau dipastikan apakah datanode2 mendapatkan data yang namenode sudah tuliskan. Dan hasilnya *space* harddisk datanode2 berkurang, yang membuktikan bahwa saat keadaan dinonaktifkan pun datanode2 tetap mendapatkan data percobaan yang tadi dituliskan.

5. KESIMPULAN

5.1 Kesimpulan

- Langkah yang dilakukan dalam implementasi HDFS sebagai infrastruktur pembangunan *big data* adalah menyiapkan lingkungan kerja HDFS dan *big data* dengan melakukan instalasi Oracle Java JDK, pembuatan *user* hadoop, mendaftarkan IP Address dan Hostname, instalasi dan konfigurasi SSH, melakukan disable IPV6, dan instalasi dan konfigurasi Hadoop pada 3 node menggunakan teknik virtualisasi.
- Faktor-faktor yang menentukan kinerja dari rancangan sistem HDFS adalah ukuran file dan ukuran *blocksize*.
- Kinerja HDFS sebagai infrastruktur pembangunan *big data* menyatakan bahwa semakin besar ukuran dari *blocksize* membuat waktu eksekusinya menjadi lebih cepat.

5.2 Saran

- Penelitian selanjutnya dapat diimplementasikan pada sistem *real* atau tidak dalam lingkungan virtualisasi

- Penelitian selanjutnya dapat dikembangkan dengan menambah jumlah datanode lebih dari 2.
- Penelitian selanjutnya dapat dikembangkan menggunakan aplikasi *benchmarks* hadoop sejenis dengan TestDFSIO.
- Size* atau ukuran data yang jauh lebih besar dan lebih banyak dari penelitian ini untuk dijadikan bahan didalam pengujian.

DAFTAR PUSTAKA

- D. J. Barrent, et al., “*SSH The Secure Shell: The Devinitive Guide*”, United Of America: O'Reilly Media Inch, 2005.
- “Data Statistik Tumbuh 50.000 GB per detik,” Databooks 3 Maret 2017. (Online) Available: <https://databoks.katadata.co.id/datapublish/2017/03/2018-data-statistik-tumbuh-50000-gb-per-detik> [diakses 20 Januari 2018]
- D. deRoos, “*Hadoop For Dummies*,” John Wiley & Sons, 2014.
- gtBlogger, “Mengulas Lengkap Tentang Hadoop: *Software* Pengelolaan Big Data,” blog.gamatechno.com, 15 Juni 2017. (Online) Available: <https://blog.gamatechno.com/software-hadoop-big-data/> [diakses 20 Januari 2018]
- “*HDFS Architecture Guide*,” The Apache Software Foundation, 08 April 2013. (Online) Available: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html#Data+Blocks [diakses 9 Mei 2018]
- Herman Yuliandoko S.T., M., “*Jaringan Komputer Wire dan Wireless Beserta Penerapannya*,” Sleman: Deepublish, 2018.
- “*MapReduce Tutorial*,” The Apache Software Foundation, 08 April 2013. (Online) Available: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html [diakses 8 Mei 2018]
- E. Paulson, “*HDFS Benchmark*,” 28 Agustus 2013. (Online) Available: <http://blog.unit1127.com/blog/2013/08/28/benchmarks/#nnbench> [diakses 9 Mei 2018]
- W. M. Wijaya, B. M., “*Teknologi BIGDATA*,” Samudera Hindia, 2015.
- M.H. Yuliandoko, “*Jaringan Komputer Wire dan Wireless Beserta Penerapannya*,” Sleman: Deepublish, 2018.

Jurnal Informatika Terpadu

Vol. 4 No. 2 Tahun 2018

Daftar Isi

- Rancang Bangun Aplikasi *Mobile Client* CRM berbasis Android Studi Kasus CV Esindo Multi Tata** 34
Riyan Wahyudi, Sirojul Munir
- Rancang Bangun Aplikasi *E-Commerce* menggunakan Framework Ruby on Rail Studi Kasus PT Hujan Rahmat Nusantara** 40
Muhamad Ikhsan, Hilmy Abidzar Tawakal
- Penerapan Efektifitas Firewall pada *Software Defind Network* berbasis Openflow** 46
Fathul Muiin, Henry Saptono
- Rancang Bangun Aplikasi Sistem Informasi Sekolah Go To School (GOS) berbasis Web menggunakan PHP** 58
Chatur Nugroho, Suhendi
- Evaluasi Kinerja HDFS sebagai Infrastruktur Pembangunan Big Data** 63
Yunita Surahman, Henry Saptono

Published by:

LPPM STT Terpadu Nurul Fikri

Jln. Raya Lenteng Agung, no. 20, Srengseng Sawah,
Jagakarsa, Jakarta Selatan, DKI Jakarta 12640

Telp. 021 - 786 3191

Email : lppm@nurulfikri.ac.id

Website : <https://journal.nurulfikri.ac.id/index.php/jit>

