

**Tugas** **Pendahuluan** **Proyek** **Akhir**

**DESAIN & IMPLEMENTASI BIG DATA PADA CLOUD-BASED IDS BERBASIS SNORT**

**DIMAS RIZKY HARSOYO PUTRO** **2110141011**

**D4** **TEKNIK** **INFORMATIKA** **DEPARTEMEN** **TEKNIK** **INFORMATIKA** **DAN** **KOMPUTER**

**POLITEKNIK** **ELEKTRONIKA** **NEGERI** **SURABAYA** **2017**

**A.** **JUDUL** **PROYEK** **AKHIR**

Desain & Implementasi Big Data pada Cloud-Based IDS Berbasis Snort

**B.** **PENDAHULUAN**

Cepatnya peningkatan volume data yang dihasilkan di internet oleh banyak kalangan mulai dari data pribadi, keuangan, hukum dan lainnya mengundang *hacker, cracker* atau *cyber-criminals*  yang lain untuk melakukan *cyber-crime*. Semakin meningkatnya jumlah serangan yang terjadi, dibutuhkan sistem yang dapat mencegah kegiatan ini untuk semakin berkembang. Solusi yang dibutuhkan juga semakin kompleks seiring dengan meningkatnya data secara heterogen dan semakin kompleks-nya arsitektur jaringan yang dihadapi. Ditambah lagi dengan adanya *grid & cloud computing environment* yang bersifat terdistribusi, menjadikannya target favorit oleh para *intruders* untuk mencari kelemahan yang dapat dimanfaatkan untuk melakukan serangan [1].

Untuk mengatasi hal tersebut, intrusion-detection systems (IDSs) digunakan untuk menginvestigasi konfigurasi, *logs*, *network traffic*, dan *user action* untuk mengidentifikasi pola serangan yang ada [2]. Meskipun demikian, untuk mengatasi masalah yang sudah disebutkan di atas secara tuntas, IDS harus dapat berjalan di *grid & cloud computing environment*. IDS harus dapat mengamati tiap node, dan ketika serangan terjadi, dapat segera memberitahu node lainnya [1]. Untuk dapat memungkinkan adanya komunikasi antar node, diperlukan kompatibilitas antar host yang berbeda-beda, mekanisme komunikasi yang bermacam-macam, dan *permission control* terhadap sistem yang ada [3].

Mata Garuda merupakan *IDS (Intrusion Detection System)* yang memonitor lalu lintas data pada jaringan internet yang ada di Indonesia. Mata Garuda akan melaporkan suatu kejadian yang cirinya telah didefinisikan dalam sebuah rule. Kejadian-kejadian tersebut didapatkan melalui sensor yang terpasang di setiap *Network Access Point* yang ada di Indonesia. Sensor tersebut berfungsi untuk mengambil paket lalu meneruskannya ke *defense center*. Sensor yang digunakan oleh Mata Garuda adalah sensor berbasis *snort* yang bersifat private untuk setiap jaringan yang terpasang sensor. Mata Garuda menggunakan teknologi RDBMS untuk melakukan *storing* *data* serta pemrosesan data secara terpusat di *defense center*.

Dengan kondisi yang ada sudah dipaparkan pada paragraf 1 & 2 di atas, tentunya Mata Garuda membutuhkan kapabilitas untuk dapat menangani pertumbuhan volume data yang semakin masif dan dapat berjalan baik di *cloud computing environment* agar menjadi solusi dari permasalahan yang ada. Oleh karena itu, pengembangan framework & arsitektur sistem Mata Garuda menjadi sistem yang mengadopsi konsep Big Data dan berjalan baik di *cloud computing environment* harus dilakukan agar dapat menyelesaikan permasalahan yang ada.

**C.** **PERUMUSAN** **MASALAH**

Mata Garuda saat ini menggunakan implementasi *RDBMS* dan penggunaan *data warehouse* untuk melakukan *storing data*. Hal tersebut mengakibatkan Mata Garuda tidak dapat mengadopsi konsep Big Data yang sangat dibutuhkan untuk memproses analisa data *event log* yang berukuran masif. Selain itu, sistemnya masih bekerja secara *centralized* yang artinya setiap pemrosesan dan analisa data *event log* yang didapatkan dari sensor diproses hanya dengan menggunakan satu *defense center*. Hal ini tentunya akan membatasi sensor yang digunakan pada Mata Garuda pada batas *private sensors* dan tentunya tidak akan berjalan baik pada *cloud computing environment*.

**D.** **TINJAUAN** **PUSTAKA**

Tinjauan pustaka ini membahas tentang teori-teori penunjang dalam penyelesaian proyek akhir ini. Beberapa teori penunjang tersebut adalah :

*1.* *Intrusion* *Detection* *System*

*Intrusion Detection System* (IDS)dapat dideskripsikan secara sederhana yaitu detektor yang bertugas untuk memproses informasi dari sistem yang akan dilindungi [2]. IDS setidaknya memiliki dua jenis, yaitu *host-based IDS* (HIDS) dan *network-based IDS* (NIDS). Pada masing masing jenis IDS, juga memiliki dua tipe pendeteksian yaitu, *misuse (rule-based)* dan *anomaly-based*.

2. Snort

Snort merupakan *Network-Based Intrusion Detection System* berlisensi *open source* yang dapat melakukan *packet logging* dan *traffic analysis* secara real-time. Dibuat oleh Martin Roesch pada tahun 1998. Snort merupakan *rule-based NIDS* yang artinya Snort menggunakan *rule* untuk mendeteksi adanya serangan pada jaringan. Snort akan melakukan tindakan yang sebelumnya telah ditentukan ketika mendeteksi adanya serangan yang sesuai dengan *ruleset* yang telah ditentukan.

3*.* HDFS

*Hadoop Distributed File System* adalah *distributed file system* berbasis *Java* yang digunakan untuk menyimpan file dalam jumlah yang masif dan disimpan secara terdistribusi di dalam banyak node komputer yang saling berhubungan. HDFS dibangun berdasarkan konsep *MapReduce* dan *Google File System* yang dikembangkan Google.

4. Apache Spark

Apache Spark adalah *engine cluster cumputing* open sourceyang khusus didesain untuk memproses big data. Dilengkapi dengan modul untuk melakukan *streaming,* SQL, *machine learning* dan *graph processing.* Apache Spark dapat berjalan berjalan di platform yang berbeda seperti Hadoop, Mesos, dan lainnya, serta dapat mengakses data dari sumber yang berbeda seperti HDFS, Cassandra, HBase, S3 ataupun sumber data yang lain. Spark memungkinkan untuk pembuatan aplikasi parallel dengan bahasa pemrograman yang berbeda-beda meliputi Java, Scala, Python, dan R.

Apache Spark dapat dikonfigurasikan agar berjalan sebagai *distributed system* menggunakan fitur RDDs (Resilient Distributed Datasets yang memungkinkan pemrosesan data dilakukan secara paralel.

5. Mata Garuda

Mata Garuda adalah IDS berbasis Snort IDS. Mata Garuda memantau lalu lintas jaringan internet di Indoneisa dan mengenali sebuah serangan atau kejadian lain yang sesuai dengan *ruleset* yang ada lalu menampilkannya dalam bentuk laporan yang dapat dengan mudah dibaca.

6. Apache Kafka

Apache Kafka merupakan *message broker* untuk mengalirkan data dari suatu sumber data ke sistem target terdistribusi secara cepat dan efektif. Kafka didesain khusus agar berjalan baik di system terdistribusi yang memiliki tingkat *throughput* yang sangat tinggi. Dibandingkan dengan *message broker* yang lain, kafka memiliki *throughput* yang lebih baik, *built-in partitioning, replication* dan *fault-tolerance*.

**E.** **PENELITIAN** **TERKAIT**

Berikut adalah penelitian yang pernah dilakukan dan relevan dengan proyek akhir ini.

1. IDS Log Analisis Menggunakan Hadoop dan Mahout untuk Data Mining Pada

Mata Garuda oleh M. Hisyam, F. A. Saputra and J. Akhmad [4]

Penelitian ini mencoba untuk melakukan pemrosesan Snort *log* *file* dengan menggunakan prinsip *Big* *Data* di aplikasi Mata Garuda. Dengan sistem terdistribusi, metode *data* *mining* dilakukan terhadap data geolocation untuk mendapatkan lokasi serangan yang terjadi. Penulis menggunakan UDTF untuk melakukan *query* serta membandingkannya dengan *join* *query*. Sedangkan algo-ritma yang diterapkan dalam proses *mining* tersebut adalah K-means *clustering* untuk mendapatkan *cluster* dari GeoIP serangan. Hasilnya, UDTF mampu meruduksi waktu komputasi menjadi 0.08 detik daripada *join* *query* yang me-makan waktu 3561 detik. Pada penelitian ini juga membuktikan bahwa penggunaan *distributed* *processing* untuk mengolah data Mata Garuda merupa-kan pilihan yang tepat karena data dapat diolah lebih cepat.

2. Real-Time Intrusion Detection System Using Multi-agent System oleh W. Laftah Al-Yaseen, Z. Ali Othman, dan M. Zakree Ahmad Nazri [5]

Penelitian ini mencoba untuk meningkatkan performa dari sistem *IDS* dengan cara mengurangi waktu pemrosesan data pada saat *IDS* melakukan analisa serangan pada lalu lintas data yang ada menggunakan *Multi-agent System*. Jumlah *agent* yang dilibatkan bersifat adaptif dan dapat berubah-ubah sesuai dengan ukuran lalu lintas data dan ketersediaan *resource* pada sistem, sehingga penggunaan *multi-agent* tidak membebani performa dari host. Hasil yang didapatkan menggunakan *Multi-agent System IDS* mampu mengurangi waktu pemrosesan analisa serangan sebesar 81% dibandingkan dengan *IDS* tradisional dengan tetap mempertahankan akurasi dari analisanya

3. Scalable Distributed Real-Time Clustering for Big Data Streams oleh Antonio Loureiro Severien [6]

Penelitian ini merupakan project pembuatan framework untuk penerapan algoritma *machine learning* pada data stream dengan volume yang sangat besar. Konsepnya adalah menggabungkan antara *machine learning, streaming model* dan *distributed computing* menjadi satu kesatuan. Hasil akhirnya berupa aplikasi bernama SAMOA (Scalable Advanced Massive Online Analysis) dengan fokusan hasilnya adalah pada aspek *scalability* dan *clustering quality*. Hasil penelitiannya menunjukan bahwa SAMOA dapat berjalan dengan baik pada *distributed environment* dan memiliki kapabilitas *scaling-up* untuk mengatasi *loads* yang lebih besar.

4. Real-time Hybrid Intrusion Detection System using Apache Storm oleh Goutam Mylvarapu, Johnson Thomas, Ashwin Kumar TK [7]

Penelitian ini mencoba menyelesaikan permasalahan lambatnya suatu IDS dalam menangani data dengan volume yang sangat besar. Solusi yang ditawarkan adalah dengan membangun *real-time hybrid IDS* dengan menggunakan Apache Storm. Apache Storm bertindak sebagai prosesor *real-time* untuk *big data stream-*nya, yang terdistribusi dan *fault tolerant*. Hybrid yang dimaksudkan adalah dengan adanya penggunaan dua algoritma *neural network* yaitu CC4 sebagai *anomaly-based detection* dan Multi Layer Perceptron sebagai *misuse-based detection*. Hasilnya menunjukan dengan *hybrid IDS* ini dapat berjalan di *distributed environment* dengan akurasi sebesar 89% dengan 4.32% tingkat *false positive*.

5. Performance Analysis of Network Intrusion Detection Schemes using Apache Spark oleh Manish Kulariya, PriyankaSaraf, Raushan Ranjan, Govind P. Gupta [8]

Penelitian ini melakukan pembandingan performa antara lima algoritma *machine learning* pada sebuah IDS dengan *distributed processing system* sebagai *engine* pemrosesan datanya. Apache spark digunakan sebagai *engine-*nya dan lima algoritma yang diimplementasikan antara lain adalah *logistic regression, support vector machines, random forest, gradient boosted decision* dan *naive bayes*. Komparasi dilakukan menggunakan dataset KDD99 sebagai variabel kontrolnya, dan hasilnya algoritma *random forest* yang memiliki performa terbaik yang dapat berjalan di *distributed environment* Apache spark.

**F.** **TUJUAN** **PROYEK** **AKHIR**

Tujuan dari proyek akhir ini adalah untuk membangun framework Mata Garuda dengan arsitektur yang mengadopsi konsep Big Data dan dapat berjalan menggunakan *public sensor* pada *cloud computing environment*.

**G.** **KONTRIBUSI** **PROYEK** **AKHIR**

Hasil dari proyek akhir ini dapat digunakan untuk meningkatkan kapabilitas Mata Garuda dalam menangani volume data yang masif dan dapat menerima data dari *public sensor* dalam *cloud computing environment*.

**H.** **METODE** **PROYEK** **AKHIR**

Untuk menyelesaikan proyek akhir ini langkah-langkah yang diambil ialah :

1. Studi Literatur

Studi literatur merupakan langkah awal dalam pengerjaan proyek akhir ini. Tahapan ini merupakan tahap yang penting untuk memelajari teori-teori serta konsep teknis maupun non-teknis yang menunjang dalam pengerjaan proyek akhir ini.

2. Pengumpulan data

Langkah selanjutnya dari penelitian ini adalah pengumpulan data log file dari Bro sebagai bahan analisa awal. Setelah data terkumpul, maka data ter-sebutakandiolahdenganteknik*textextraction*untukmembentukalgoritma dalam log parser.

3. Perancangan sistem

Setelah tahap studi literatur, langkah yang diambil selanjutnya adalah melakukan perancangan sistem.

Secara umum, konfigurasi dari blok diagram sistemnya adalah sebagai berikut

Spark Streaming

MLlib

Apache Spark

Kafka

HDFS

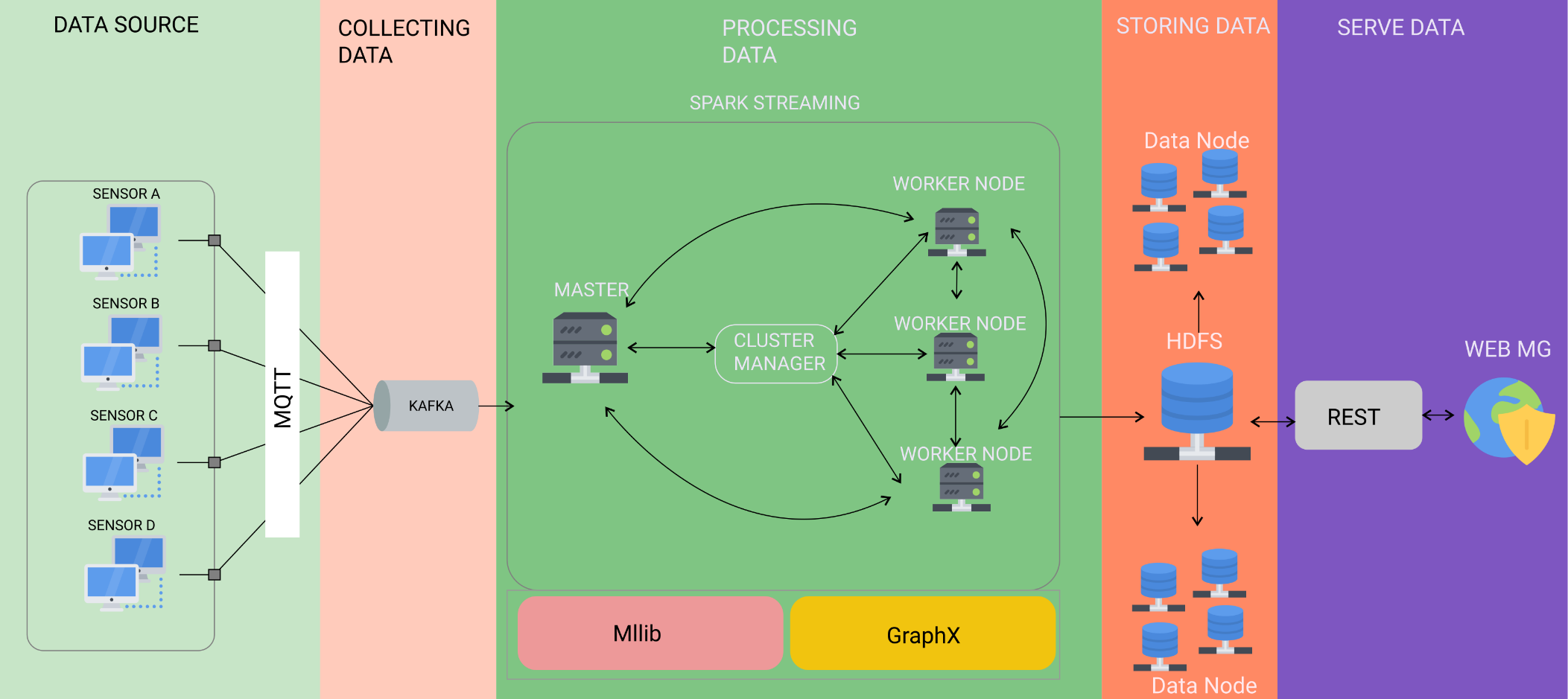
REST API

Web Interface

Sensor

GrarphX

Gambar 1. Blok diagram sistem secara umum



Gambar 2. Alur Desain Sistem Keseluruhan

1. Sensor merupakan sensor mata garuda yang menerima kejadian-kejadian yang ada di jaringan. Kejadian tersebut berupa *log file* yang berukuran besar. Untuk setiap kejadian yang terjadi akan dikirimkan *log file* ke *defense center* lalu akan diproses dan dianalisa disana. Sensor dapat ditempatkan sebagai *public sensor* maupun *private sensor*.

Kafka

MQTT

Ruleset1

Ruleset2

Public Sensor

Private Sensor

MQTT

Ruleset1

Ruleset2

Gambar 2. Pengambilan data dari data source

1. Kafka

Kafka akan berperan sebagai message broker yang mengirimkan output dari sensor menuju ke *spark core engine* untuk nantinya diproses dan dimasukan kedalam *storage* HDFS

Message Queue

Penerima Data (Apache Spark)

Pengirim Data

Kafka

Penerima Data (Apache Spark)

Gambar

1. Rancangan Software

Software yang dihasilkan akan berupa web apps dengan fitur dasar pada sebuah framework meliputi,

* + - Login
    - Admin Dashboard (CRUD) & Dashboard User
    - Mengatur HDFS
    - Konfigurasi
    - Membuat API key untuk sensor
    - Melihat Profile
    - Mengakses Report & Metric yang ada
    - Authentication untuk sensor baru
    - Read data melalui data source

1. REST API

REST API merupakan service yang digunakan untuk melakukan pengambilan data dari database HDFS untuk ditampilkan ke web interface.

* + - Metric Mata Garuda

Dalam service-nya digunakan metric mata garuda antara lain adalah, *event monitoring, event statistics, top signature, top protocols, sensor statistics*, *daily-monthly-annually report, attack trend, event analysis, user-role-profile-menu management*

1. Web Interface

Web interface akan digunakan untuk menampilkan hasil query yang dilakukan oleh REST API. Laporan yang ditampilkan adalah berupa laporan terkait serangan dan kejadian yang ditangkap oleh sensor.

1. Apache Spark

Spark bekerja sebagai execution engine untuk pemrosesan data yang berasal dari sensor. Spark diintegrasikan dengan menggunakan SQL Hive. Spark bertugas untuk melakukan eksekusi query yang dilakukan oleh REST API

* 1. Streaming SQL

Dapat memungkinkan spark memproses data secara stream, sehingga memungkinkan untuk melakukan *write job* secara streaming

* 1. Mllib

Mllib merupakan library machine learning dari spark, bertugas untuk melakukan klustering data yang ada pada storage HDFS

* 1. GraphX

Merupakan library spark untuk melakuka komputasi graph-paralel

4. Implementasi

Dalam tahap ini akan dibangun arsitektur dan desain untuk mengaplikasikan konsep Big Data. Pengaplikasian dimulai dari mengkonfigurasikan bentuk output dari sensor Snort dengan menggunakan *Snort MQTT* agar dapat sesuai dengan sensor sensor yang lain, lalu mengkonfigurasikan dan mengintegrasikan apache kafka sebagai *pub-sub* dengan apache spark core engine agar dapat mengirimkan dan menerima pesan dari/ke spark engine.

Setelah itu akan dilakukan instalasi mesin master & slave node apache spark engine yang digunakan untuk melakukan *stream processing* terhadap data yang masuk ke engine, serta membangun konfigurasi *file system* menggunakan HDFS.

Tahap terakhir adalah membuat frontend web REST API untuk masing-masing metric yang ada. Melakukan debugging dan peningkatan performa pada program web

5. Pengujian dan analisa

Pada tahap ini, dilakukan pengujian terhadap implementasi desain sistem dengan menggunakan dataset skala lab yang sesuai dengan desain intrusi yang ada. Dan melakukan komparasi performa dengan desain sistem yang ada saat ini. Selain itu juga akan dilakukan testing dengan dataset dengan volume besar dan pemasangan sensor publik untuk melakukan uji kehandalan di *cloud computing environment*.

**I.** **JADWAL** **PELAKSANAAN**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kegiatan | Bulan | | | | | | | | | | | | |
| 8 | 9 | 10 | 11 | 12 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Proposal ProyekAkhir |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pengambilan Data |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
| Implementasi/Coding |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
| Debugging |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing &Analisa |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan Buku PA |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
| Sidang PA |  |  |  |  |  |  |  |  |  |  |  |  |  |

**J.** **PERSONALIA** **PROYEK** **AKHIR**

Mahasiswa

Nama

NRP Jurusan Agama

Jenis kelamin

Dosen Pembimbing 1 Nama

NIP

: Dimas Rizky H.P.

: 2110141011

: Diploma IV - Teknik Informatika : Islam

: Laki-laki

: Ferry Astika Saputra, ST, M.Sc

: 197708232001121002

Departemen

Bidang keahlian

Dosen pembimbing 2

Nama NIP

Departemen

Bidang keahlian

: Teknik Informatika

: Computer Network, Network Security

: Jauari Akhmad S.ST

: 2000000052

: Teknik Informatika

: Rekayasa Perangkat Lunak

**K.** **PERKIRAAN** **BIAYA** **PROYEK** **AKHIR**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Uraian | Jumlah | Harga Satuan(Rp) | Total (Rp) |
| 1 | RAM 8GB | 4 buah | 800.000 | 3.200.000 |
| 2 | Kertas A4 HVS 80gr | 1 rim | 40.000 | 40.000 |
| 3 | Kertas A5 | 1 rim | 40.000 | 40.000 |
| 4 | Tinta Printer | 4 buah | 25.000 | 100.000 |
| 5 | Internet | 8 bulan | 100.000 | 800.000 |
| 6 | Dokumentasi Laporan | 6 bundle | 25.000 | 150.000 |
| **TOTAL** **PENGELUARAN** | | | | **4.330.000** |

**L.** **DAFTAR** **PUSTAKA**

|  |  |
| --- | --- |
| [1] | K. Vieira, A. Schulter, C. B. Westphall and C. M. Westphall, "Intrusion Detection for Grid and Cloud Computing," *IT Professional,* vol. 12, no. 4, pp. 38-43, 2010. |
| [2] | H. Debar, M. Dacier and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer Networks,* vol. 31, no. 8, pp. 805-822, 1999. |
| [3] | I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, "A Security Architecture for Computational Grids," in *CCS '98 Proceedings of the 5th ACM conference on Computer and communications*, San Francisco, 1998. |
| [4] | M. Hisyam, F. A. Saputra and J. A. Nur Hasyim, IDS Log Analisis Menggunakan Hadoop dan Mahout untuk Data Mining Pada Mata Garuda, Surabaya: Politeknik Elektronika Negeri Surabaya, 2015. |
| [5] | W. L. Al-Yaseen, Z. A. Othman and M. Z. Ahmad Nazri, "Real-Time Intrusion Detection System Using Multi-agent System," *IAENG International Journal of Computer Science,* pp. 1-11, 2016. |
| [6] | A. L. Severien, "Scalable Distributed Real-Time Clustering for Big Data Streams," p. 2013, 25 June 2013. |
| [7] | G. Mylvarapu, J. Thomas and A. K. TK, "Real-time Hybrid Intrusion Detection System using Apache Storm," in *IEEE 17th International Conference on High Performance Computing and Communications*, New York, 2015. |
| [8] | M. Kulariya, R. Ranjan and P. G. Gupta, "Performance Analysis of Network Intrusion Detection Schemes using Apache Spark," in *International Conference on Communication and Signal Processing*, India, 2016. |