

# PENGEMBANGAN DASHBOARD SISTEM PENCATATAN LOG SERVER MENGGUNAKAN ELASTICSEARCH-FLUENTD- KIBANA (EFK) STACK

*by* John Doe

---

**Submission date:** 02-Jun-2020 07:10PM (UTC+0700)

**Submission ID:** 1336478891

**File name:** Paper-Publikasi-Ilmiah.pdf (655.07K)

**Word count:** 2571

**Character count:** 16317

# PENGEMBANGAN DASBOR SISTEM PENCATATAN *LOG SERVER* MENGGUNAKAN ELASTICSEARCH-FLUENTD-KIBANA (EFK) STACK

8  
Fakultas Teknologi Industri  
Universitas Islam Indonesia  
Yogyakarta, Indonesia

Fakultas Teknologi Industri  
Universitas Islam Indonesia  
Yogyakarta, Indonesia

14  
**Abstract**—Badan Sistem Informasi (BSI) yang ada di Universitas Islam Indonesia (UII) adalah sebuah badan yang bertugas untuk menyediakan layanan sistem informasi dan juga internet di lingkup kampus UII. Bertambahnya kebutuhan civitas akademika UII akan layanan aplikasi yang disediakan oleh BSI menyebabkan semakin bertambahnya jumlah *server* aplikasi yang harus dikelola oleh BSI. *Server* yang berisi *log* ini tentunya membantu kinerja *SysAdmin* dalam memantau kondisi *server-server* yang ada. Namun banyaknya *log* yang ada disetiap *server* tersebut menyita banyak waktu *SysAdmin* dalam mencari *log* yang berisikan error terlebih bila harus dilakukan pengecekan *server* satu persatu. Solusi dari masalah tersebut adalah dibuatnya sebuah *Centralized Log* yang menyimpan semua *log* yang ada untuk dilakukan monitoring oleh *SysAdmin*. *Centralized Log* ini menggunakan teknologi Elasticsearch, Fluentd, Kibana (EFK) Stack dan juga Fluentbit. Data *log* yang diambil oleh Fluentbit tersebut pada masing-masing *server* kemudian akan diolah di EFK stack untuk ditampilkan dalam bentuk visualisasi yang mudah untuk dibaca oleh *SysAdmin*. Hasil akhir dari penerapan teknologi EFK stack ini adalah *SysAdmin* dapat menemukan masalah/error dengan tepat dan mengurangi waktu untuk memperbaiki masalah pada *server* tersebut.

**Keywords**—Elasticsearch, Fluentd, Kibana, Centralized log

## I. PENDAHULUAN

Badan Sistem Informasi (BSI) secara umum bertugas mengawal perencanaan, pengembangan, operasi, serta layanan sistem dan teknologi informasi di lingkungan Universitas Islam Indonesia (UII) dengan jumlah pengguna yang mencapai kurang lebih 30.000 pengguna yang terdiri dari mahasiswa, dosen tenaga pendidikan, dan para pemangku kepentingan lainnya. Layanan yang diberikan oleh BSI sendiri terus bertambah menyesuaikan kebutuhan sivitas akademika baik itu karyawan ataupun mahasiswa. Sebagian layanan berupa pengembangan sistem informasi atau perangkat lunak lainnya untuk mendukung kebutuhan bisnis organisasi. Semua sistem informasi baik *desktop* maupun berbasis web tersebut membutuhkan sebuah *server* untuk berjalan dan di dalam *server* tersebut terdapat sebuah *log* yang sangat membantu kinerja *SysAdmin* dalam memonitor keadaan *server* untuk dilakukan pemeliharaan *server*. Selain memonitor, tugas lain dari *SysAdmin* adalah melakukan perbaikan apabila terjadi *error* pada sebuah *server* atau layanan yang sedang berjalan.

Perbaikan ini tentunya diharapkan diselesaikan dengan waktu singkat mengingat layanan tersebut digunakan oleh banyak orang baik luar maupun dalam lingkungan UII. Maka dari itu solusi untuk mempersingkat waktu yang diperlukan oleh *SysAdmin* adalah membuat sebuah dasbor *log* yang terpusat sehingga memudahkan pekerjaan *SysAdmin* untuk mencari *error log* yang ada serta tidak harus membuang waktu dengan mencari *log* satu persatu pada tiap *server* yang ada.

Dasbor *log* adalah sebuah sistem informasi yang bertujuan untuk menampilkan data *log* yang ada pada sebuah *server*. Dasbor *log* juga dapat dikembangkan menjadi sebuah dasbor *log* terpusat yang menampilkan semua data *server* dan menggabungkannya pada satu titik tertentu. Dasbor *log* yang mencatat *log* secara terpusat ini sedang dibutuhkan oleh Badan Sistem Informasi (BSI) yang ada di Universitas Islam Indonesia (UII). Sistem dasbor *log* terpusat ini menggunakan teknologi Elasticsearch, Fluentd, Kibana stack (EFK stack) serta teknologi tambahan yaitu Fluentbit.

Pada pekerjaan magang ini EFK stack digunakan untuk mengolah dan memonitor data *log* yang terdapat pada sebuah *server/node*. Selain EFK stack digunakan juga teknologi bernama Fluentbit. Penjelasan dari 4 teknologi yang digunakan adalah sebagai berikut : Fluentbit merupakan turunan dari Fluentd yang berfungsi sebagai pengumpul *log/log collector* yang kemudian semua *log* yang telah diambil dari berbagai *server* tersebut diteruskan ke Fluentd yang berfungsi sebagai *aggregator* untuk melakukan filtering data *log*. Setelah data di filtering kemudian Fluentd meneruskan data tersebut untuk disimpan di Elasticsearch yang berfungsi sebagai penyimpanan/storage dan kemudian dilakukan indexing. Setelah data terindeks maka tugas dari Kibana adalah memvisualisasikan data tersebut agar dapat terbaca dengan baik.

Guna mempermudah pembaca mengetahui isi dari setiap bab yang disampaikan, berikut adalah penjelasan dari masing masing bab yang ditulis : bab I berisikan pokok masalah yang akan diselesaikan dalam makalah ini, bab II berisikan teori-teori pendukung untuk menyelesaikan makalah ini, bab III berisikan proses dalam menyelesaikan sistem ini, bab IV berisikan hasil dan pembahasan pada sistem yang sudah dibuat, bab V berisikan kesimpulan yang bisa diambil dalam menyelesaikan sistem tersebut dan saran yang bisa digunakan

kedepannya untuk meningkatkan kinerja sistem yang telah dibuat, dan terakhir bab VI berisikan referensi pendukung sebagai alat pendukung dalam menyelesaikan makalah ini.

## II. DASAR TEORI DAN STUDI PUSTAKA

### A. ELASTICSEARCH

Elasticsearch adalah sebuah sistem yang berfungsi sebagai *database* yang biasa digunakan dalam EFK stack dan ELK stack yang ada. Elasticsearch sendiri merupakan *database* noSQL yang berfokus pada *search engine database* yang ada dan sifatnya terbilang cukup unik karena di Elasticsearch kita bisa mengamsumsi indeks sebagai sebuah *database*, *types* sebagai *table*, dokumen sebagai *record* atau *row*, dan *mapping* sebagai skema tabel [1].

Cara kerja dari Elasticsearch adalah setelah menerima *data flow* dari sebuah *aggregator* atau sumber yang mengirimkan data tersebut (baik berupa *log*, *metric* sistem, dan web aplikasi). Kemudian data di *ingest* atau lebih tepatnya terjadi *Data Ingestion* di Elasticsearch yang memproses data tersebut (*parse*, *normalized*, dan *enriched*) sebelum kemudian dilakukan *indexing*/dimasukkan ke dalam Elasticsearch. Setelah data masuk ke dalam Elasticsearch maka *user* dapat melakukan *query* kompleks terhadap data mereka dan mengambil ringkasan data untuk ditampilkan di Kibana [2].

Adapun penelitian yang dilakukan oleh [3] menggunakan Elasticsearch sebagai *query* data untuk mengolah data Portal Pengembangan dan Pembinaan Sumber Daya Manusia (PPSDM) yang dimiliki oleh Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah (LKPP) dalam menangani masalah monitoring dan evaluasi data pada saat terjadinya sertifikasi dan pelatihan yang ada di Portal PPSDM sehingga latensi data yang terjadi berkurang selama 2,5 detik.

### B. FLUENTD

Fluentd adalah sebuah sistem/service yang berfungsi sebagai *data collector* yang menyatukan berbagai jenis *log* dari semua macam-macam *log* yang ada [4]. Sistem yang dikembangkan oleh perusahaan Treasure Data ini bersifat *open source* dan sampai saat ini telah banyak digunakan oleh berbagai perusahaan untuk menangani masalah *log* yang ada. Sifat dari Fluentd juga dapat sebagai *Aggregator* yang memungkinkan Fluentd melakukan *filtering* data *log* yang masuk sebelum mengarahkan data tersebut ke sebuah *Storage*, dalam kasus kali ini adalah Elasticsearch.

Data yang disimpan di Fluentd berformat JSON sebisanya mungkin karena *downstream* data *processing* lebih mudah dan fleksibel dengan menggunakan format JSON. Plugins yang ada di Fluentd juga terbilang cukup banyak yaitu lebih dari 500 plugin dan terus bertambah dengan seiringnya waktu dari bantuan perusahaan pengembang maupun user yang memakai Fluentd, saat ini telah digunakan lebih dari 2000 perusahaan dan di *deploy* di 50000 *server* dan terus bertambah. Selain hal tersebut Fluentd juga membutuhkan sumberdaya yang sedikit saat digunakan dan juga memiliki kelebihan *memory file* berbasis *buffering* sebagai tindakan pencegahan untuk kehilangan data [5].

### C. KIBANA

Kibana adalah sebuah *open source* aplikasi *front end* yang berada diatas Elastic-stack dan menyediakan pencarian dan visualisasi data yang telah di *index* di dalam Elasticsearch. Kibana juga bertindak sebagai *user interface* yang bertujuan untuk memonitor, mengelola, dan mengamankan sebuah

Elastic cluster/stack. Dikembangkan mulai dari tahun 2013 di komunitas Elastic hingga sekarang Kibana telah menjadi jendela bagi Elastic stack itu sendiri dan menjadi sebuah portal bagi *user* dan perusahaan [6].

Pada penelitian yang dilakukan oleh [7] menggunakan Kibana sebagai visualisasi data yang digunakan untuk melihat kualitas *internet mobile broadband* dari berbagai macam Internet Service Provider (ISP) untuk pengguna yang ada di daerahnya tersebut.

Adapun penelitian yang dilakukan oleh [8] yang berkebetulan bertempat di BSI juga menggunakan Kibana sebagai visualisasi data *log* yang ada sehingga memudahkan proses monitoring untuk *log server* yang ada di BSI tersebut.

### D. FLUENTBIT

Fluentbit adalah sebuah *open source* platform yang digunakan sebagai *log processor* atau *log forwarder* yang mengambil data *log* dari berbagai sumber yang ada dan menggabungkan *log-log* tersebut kemudian mengerimkannya ke berbagai lokasi yang ada dan dalam kasus kali ini adalah Fluentd [9]. Lebih mudahnya Fluentbit merupakan versi mini dari Fluentd yang cocok dipakai di *environment* yang membutuhkan kapasitas terbatas.

Perbedaan mendasar lainnya diantara Fluentd dan Fluentbit adalah hanya memakai Bahasa C, *lightweight*/ringan sekali karena hanya memakan sekitar 450KB *memory*, dan tidak bisa digunakan sebagai *aggregator*. Pada pekerjaan kali ini Fluentbit digunakan di *host/server* yang ada untuk mengambil data *log* yang diperlukan. Hal lainnya yang perlu diperhatikan adalah kombinasi Fluentd dan Fluentbit kali ini sangat populer digunakan dalam kluster kubernetes.

### E. CENTRALIZED LOG

*Centralized Log* atau sering disebut dengan *Centralized Log Management* (CLM) adalah sebuah tipe sistem *logging* solusi yang menggabungkan semua data *log* yang dimiliki dan kemudian di arahkan menuju sebuah pusat yang mudah diakses dan memiliki UI yang mudah digunakan. *Centralized Log* dibuat untuk memudahkan pekerjaan seorang *engineer*. CLM tidak hanya menyediakan fitur bagi *engineer* untuk mengambil data saja, selain fitur tersebut CLM juga menggabungkan, menganalisis, dan memberikan sebuah gambaran informasi dengan cepat dan jelas [10].

## III. METODOLOGI

Dalam pengerjaan makalah ini telah ditentukan langkah-langkah yang harus diambil berdasarkan kebutuhan pekerjaan yang telah didiskusikan oleh pengampu kepentingan sistem ini (dalam kondisi ini adalah tim Site Reliability Engineer BSD), hasil dari langkah-langkah yang diambil adalah sebagai berikut :

- Perancangan Arsitektur Centralized Log
- Konfigurasi Fluentbit
- Konfigurasi Fluentd
- Konfigurasi Elasticsearch
- Konfigurasi Kibana

### A. PERANCANGAN ARSITEKTUR CENTRALIZED LOG

Setelah melakukan diskusi dengan Tim Site Reliability Engineer (SRE) yang ada di BSI maka hasil arsitektur yang disepakati dan akan dilakukan pengembangan adalah sebagai Gambar 3.1 :



Gambar 3.1 Arsitektur Centralized Log

Adapun penjelasan yang bisa dijabarkan berdasarkan arsitektur Gambar 3.1 Arsitektur Centralized Log adalah sebagai berikut :

- Fluentbit - digunakan sebagai pengumpul log (log collector) yang ada di setiap server dan kemudian meneruskan semua log yang sudah diambil ke Fluentd.
- Fluentd - sebuah aggregator yang digunakan sebagai penerima log dari berbagai log collector yang ada, dalam kasus kali ini adalah Fluentbit.
- Elasticsearch - database yang digunakan untuk menyimpan data log yang sudah dikirimkan oleh Fluentd dan sebagai query log untuk Kibana.
- Kibana – adalah sebuah sistem untuk melakukan visualisasi terhadap log yang sudah tersimpan di dalam Elasticsearch, data log yang tersimpan di Elasticsearch di-query oleh Kibana untuk kemudian ditampilkan dengan bentuk chart.

### B. KONFIGURASI FLUENT BIT

Sebelum melakukan konfigurasi untuk Fluentbit sebagai log collector, dilakukan terlebih dahulu sebuah instalasi aplikasi pendukung pada server yaitu :

- Compiler : GCC atau clang
- Cmake
- Flex dan Bison
- Fluenbit versi terbaru

Setelah meng-install 4 aplikasi pendukung diatas, maka dilanjutkan dengan mengkonfigurasi sebuah file untuk mengambil data log yang diinginkan dan mengirimkan data log tersebut ke Fluentd sebagai aggregator yang ada. Contoh konfigurasi file untuk mengambil data syslog dan ssh auth log yang ada pada sebuah server pada Gambar 3.2 Konfigurasi Fluentbit

```
[SERVICE]
  Flush 5
  Daemon on
  Log_Level debug

#Mengambil syslog di diri sendiri
[INPUT]
  Name tail
  Tag syslog
  Path /var/log/syslog
  Refresh_Interval 60

#Mengambil ssh di diri sendiri
[INPUT]
  Name tail
  Tag ssh
  Path /var/log/auth.log
  Refresh_Interval 60

#Forward syslog ke aggregator
[OUTPUT]
  Name forward
  Match *
  Host [REDACTED]
  Port 8888
```

Gambar 3.2 Konfigurasi Fluentbit

### C. KONFIGURASI FLUENTD

Sama seperti Fluentbit yaitu sebelum melakukan sebuah konfigurasi untuk log aggregator, maka dilakukan terlebih dahulu meng-install aplikasi pendukung yaitu :

- Ruby Gem
- Fluentd
- Plugins Fluentd untuk elasticsearch

Setelah meng-install 3 aplikasi pendukung diatas maka Fluentd sudah bisa digunakan sebagai log aggregator yang berfungsi sebagai penerima semua log collector yang berjalan. Gambar 3.3 adalah contoh konfigurasi Fluentd sebagai log aggregator :

```
<source>
  @type forward
  port 8888
  bind 0.0.0.0
</source>

<match **>
  @type copy
  <store>
    @type elasticsearch
    host [REDACTED]
    port 9200
    flush_interval 10s
    logstash_format true
  </store>
</match>
```

Gambar 3.3 Konfigurasi Fluentd

#### D. KONFIGURASI ELASTICSEARCH

Konfigurasi yang dilakukan pada sisi Elasticsearch cukup mudah karena hanya perlu melakukan instalasi dan pemberian nama cluster saja di *server* yang akan menjadi tempat penyimpanan/storage data *log* ini. **Gambar 3.4** menunjukkan hasil konfigurasi Elasticsearch di *server* yang ada :

```
"name" : "node-masterefk",
"cluster_name" : "visualisasiefk",
"cluster_uuid" : "nRFqBIFGSRe6cEjeDhfcIA",
"version" : {
  "number" : "7.6.0",
  "build_flavor" : "default",
  "build_type" : "deb",
  "build_hash" : "7f634e9f44034fbc12724506cc1da681b0c3b1e3",
  "build_date" : "2020-02-06T00:09:00.449973Z",
  "build_snapshot" : false,
  "lucene_version" : "8.4.0",
  "minimum_wire_compatibility_version" : "6.8.0",
  "minimum_index_compatibility_version" : "6.0.0-beta1"
},
"tagline" : "You Know, for Search"
```

Gambar 3.4 Cluster Elasticsearch

#### E. KONFIGURASI KIBANA

Konfigurasi di Kibana juga tidak terlalu rumit dan sama dengan konfigurasi yang telah dilakukan di Elasticsearch yaitu hanya perlu untuk meng-install Kibana pada *server* yang akan dijadikan tempat untuk visualisasi data *log* ini **Gambar 3.5** adalah hasil dari instalasi yang dilakukan, halaman awal Kibana



Gambar 3.5 Hasil Instalasi Kibana

Setelah itu dilakukan pembuatan visualisasi satu persatu untuk *log* yang telah masuk ke dalam Elasticsearch, 3 indeks ini adalah hasil visualisasi yang telah dibuat **Gambar 3.6** :

| Title         | Type | Description                            | Actions |
|---------------|------|--|---------|
| Monitor Nginx | Pie  | Memonitor Nginx yang ada di website    |         |
| SSH attack    | Pie  | Attack yang berusaha menaruh leter ssh |         |
| Sylog Monitor | Pie  | Monitor Behavior Server                |         |

Gambar 3.6 3 Visualisasi yang telah dibuat

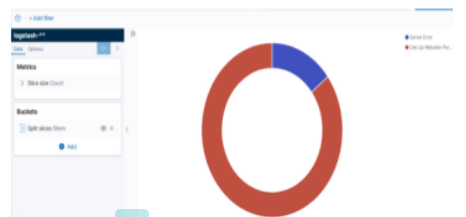
Hasil dan gambaran lebih jelas dari 3 visualisasi tersebut dan juga hasil dari dashboard yang telah dibuat akan disajikan pada Bab selanjutnya yaitu Bab IV Hasil dan Pembahasan.

#### IV. HASIL DAN PEMBAHASAN

Setelah melakukan instalasi dan konfigurasi yang diperlukan dalam membuat *Centralized Log* ini dapat diambil berbagai manfaat dari EFK + Fluenbit stack ini yaitu :

- Fluentbit sebagai *log collector* yang dapat mengambil berbagai *log* yang diinginkan serta dapat meneruskannya ke berbagai tujuan yang hanya membutuhkan seminimalnya 1 konfigurasi saja.
- Fluentd sebagai *log aggregator* yang menerima semua data *log* dari berbagai sumber kemudian memasukkan semua data tersebut ke Elasticsearch yang bertindak sebagai penyimpanan *log* juga hanya membutuhkan seminimalnya 1 konfigurasi saja.
- Kibana sebagai visualisasi yang tidak terbatas hanya dengan 1 visualisasi saja untuk memonitor berbagai jenis *log* yang diinginkan dan dapat diskustomisasi dengan mudah dan sesuai dengan kebutuhan yang ada.

Hal diatas tidaklah mungkin dilakukan tanpa bantuan Elasticsearch sebagai tempat penyimpanan yang handal dan dapat bekerja dengan baik berbarengan dengan Kibana sebagai visualisasi data yang ada di dalam Elasticsearch tersebut. Pada **Gambar 3.7**, **Gambar 3.8**, **Gambar 3.9**, dan **Gambar 3.10** adalah hasil visualisasi dan dashboard tersebut.



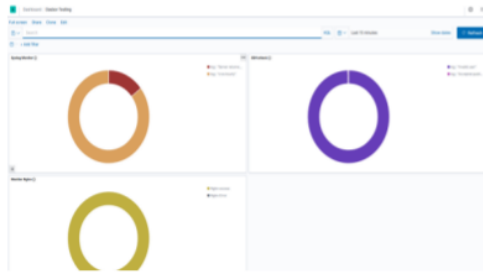
Gambar 3.7 Monitor Syslog



Gambar 3.8 Monitor Nginx Website



Gambar 3.9 Monitor SSH Log



Gambar 3.10 Dasbor Monitor 3 Visualisasi

Setelah dilakukannya penerapan EFK + Fluentbit stack ini dapat disimpulkan pula bahwa hal ini membantu *SysAdmin* dalam meminimalisir pekerjaan yang dilakukan tanpa harus mengecek *server* satu-persatu. Adapun perbandingan hasil dari penerapan ini bisa dilihat pada **Gambar 3.11** dan **Gambar 3.12**.



Gambar 3.11 Cek Log secara Manual



Gambar 3.12 Cek Log menggunakan Visualisasi

Hasil dari perbandingan 2 gambar diatas menunjukkan bahwa dasbor monitor sangat membantu kinerja *SysAdmin* dalam mencari *log* yang bermasalah atau *server* yang sedang butuh dilakukan *maintenance* dengan cepat tanpa perlu membuang waktu mencari *log server* satu-persatu.

### V. KESIMPULAN DAN SARAN

#### A. Kesimpulan

Kesimpulan yang diambil dari pengembangan dasbor sistem pencatatan *log* menggunakan penerapan EFK + Fluentbit stack ini adalah sistem mampu membantu kinerja *SysAdmin* dalam memonitor *server* yang ada berdasarkan sistem monitor *log* dasbor yang telah dibuat. Dasbor tersebut membantu kinerja *SysAdmin* dalam mengurangi waktu dan dengan tepat menemukan sebuah *error* pada *server* yang sedang dilakukan perbaikan. Sehingga tidak perlu mengecek *log* secara manual lewat *command line* di *server* satu persatu sebagaimana hasil perbandingan tampilan *log* pada bab 3 yaitu pada **Gambar 3.11** dan **Gambar 3.12**.

#### B. Saran

Pengembangan yang dilakukan dengan EFK +Fluentbit stack dalam pekerjaan dan makalah ini dapat dikembangkan lebih lanjut guna memaksimalkan penggunaan dasbor pada sistem yang telah dibuat. Beberapa saran tersebut antara lain adalah :

- Menggunakan teknologi kubernetes sehingga dalam meng-install Fluentbit dapat dilakukan dengan *daemonset* (otomatis).
- Membuat konfigurasi Fluentbit agar otomatis berjalan kembali setelah *server* berhenti mendadak/restart sendiri.
- Memaksimalkan penggunaan fluentbit dengan mengambil beberapa *log* lain dan memaksimalkan fitur filter yang ada di Fluentd.
- Mengelompokkan *log* berdasarkan *server* yang dimiliki dalam konfigurasi Kibana.

### VI. REFERENSI

- [1] M. Arslan, "codepolitan.com," CodePolitan, 23 Desember 2016. [Online]. Available: <https://www.codepolitan.com/pengenalan-singkat-elasticsearch>. [Accessed 20 Mei 2020].
- [2] Elastic, "elastic.co," Elasticsearch, [Online]. Available: <https://www.elastic.co/what-is/elasticsearch>. [Accessed 20 Mei 2020].
- [3] E. P. Sartika, "PEMANFAATAN ELASTICSEARCH LOGSTASH KIBANA UNTUK PENGELOLAAN DAN VISUALISASI DATA PORTAL PENGEMBANGAN DAN PEMBINAAN SUMBER DAYA MANUSIA (PORTAL PPSDM)," *Informati Engineering Universitas Islam Indonesia*, 2020.
- [4] Fluentd, "fluentd.org," Cloud native Computing Foundation, [Online]. Available: <https://www.fluentd.org/>. [Accessed 20 Mei 2020].
- [5] Fluentd, "fluentd.org," Cloud Native Computing Foundation, [Online]. Available: <https://www.fluentd.org/architecture>. [Accessed 20 Mei 2020].
- [6] Elastic, "elastic.co," Elasticsearch, [Online]. Available: <https://www.elastic.co/what-is/kibana>. [Accessed 20 Mei 2020].
- [7] H. I. Wicaksono, "VISUALISASI KUALITAS INTERNET MOBILE BROADBAND MENGGUNAKAN ELK STACK (Studi Kasus di Daerah Jambidan, Banguntapan, Bantul )," *Informati Engineering Universitas Islam Indonesia*, 2020.
- [8] Y. B. Erwinsyah, "KONSOLIDASI DAN VISUALISASI LOG SERVER BSI UII MENGGUNAKAN ELK STACK," *Informati Engineering Universitas Islam Indonesia*, 2020.
- [9] Fluentbit, "fluentbit.io," Treasure Data, [Online]. Available: <https://fluentbit.io/>. [Accessed 20 Mei 2020].
- [10] J. Morgan, "missioncloud.com," Mission Cloud Services Inc, 31 Mei 2016. [Online]. Available: <https://www.missioncloud.com/blog/what-is-centralized-log-management-clm>. [Accessed 20 Mei 2020].

# PENGEMBANGAN DASBOR SISTEM PENCATATAN LOG SERVER MENGGUNAKAN ELASTICSEARCH-FLUENTD-KIBANA (EFK) STACK

---

## ORIGINALITY REPORT

---

8%

SIMILARITY INDEX

5%

INTERNET SOURCES

2%

PUBLICATIONS

6%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

1

[bsi.uui.ac.id](http://bsi.uui.ac.id)

Internet Source

1%

---

2

Panagiotis Trakadas, Nikolaos Nomikos, Emmanouel T. Michailidis, Theodore Zahariadis et al. "Hybrid Clouds for Data-Intensive, 5G-Enabled IoT Applications: An Overview, Key Issues and Relevant Architecture", Sensors, 2019

Publication

1%

---

3

Submitted to STIKOM Surabaya

Student Paper

1%

---

4

Tatsuya SATO, Yosuke HIMURA, Yoshiko YASUDA. "Evidence-Based Context-Aware Log Data Management for Integrated Monitoring System", IEICE Transactions on Communications, 2018

Publication

1%

---

5

[www.kompy.info](http://www.kompy.info)

Internet Source

<1%

6

[www.kaimanakab.go.id](http://www.kaimanakab.go.id)

Internet Source

<1%

7

Submitted to Regis University

Student Paper

<1%

8

[seminar.ilkom.unsri.ac.id](http://seminar.ilkom.unsri.ac.id)

Internet Source

<1%

9

[hdl.handle.net](http://hdl.handle.net)

Internet Source

<1%

10

[eprints.umm.ac.id](http://eprints.umm.ac.id)

Internet Source

<1%

11

[www.lkpp.go.id](http://www.lkpp.go.id)

Internet Source

<1%

12

Submitted to Universitas Islam Indonesia

Student Paper

<1%

13

[es.scribd.com](http://es.scribd.com)

Internet Source

<1%

14

[medicine.uui.ac.id](http://medicine.uui.ac.id)

Internet Source

<1%

15

Submitted to Universitas Jenderal Soedirman

Student Paper

<1%

16

Submitted to Universitas Muhammadiyah

<1%



# Surakarta

Student Paper

---

---

Exclude quotes      Off

Exclude matches      Off

Exclude bibliography      On

# PENGEMBANGAN DASHBOARD SISTEM PENCATATAN LOG SERVER MENGGUNAKAN ELASTICSEARCH-FLUENTD-KIBANA (EFK) STACK

---

PAGE 1

---

PAGE 2

---

PAGE 3

---

PAGE 4

---

PAGE 5

---