

Implementasi Elasticsearch Logstash Kibana Stack pada Sistem Portal Pengembangan dan Pembinaan Sumber Daya Manusia

(Studi kasus : Bidang PPSDM pada Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah)

Elang Putra Sartika¹
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta, Indonesia
16523169@students.uii.ac.id

Andhik Budi Cahyono²
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta, Indonesia
andhik.budi@uii.ac.id

Abstrak — Portal Pengembangan dan Pembinaan Sumber Daya Manusia (PPSDM) adalah sebuah *e-learning* yang digunakan sebagai sarana pelatihan dan sertifikasi peserta Pengadaan Barang/Jasa (PBJ) pada Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah (LKPP). Sistem Portal PPSDM ini sudah memiliki sekitar 150 ribu pengguna aktif. Pengguna sistem ini semakin bertambah begitu juga volume dari basis datanya. Besarnya volume data menyebabkan proses pada basis data semakin lambat dan waktu jeda pada proses bertambah. Waktu jeda ini dapat mempengaruhi keakuratan dari hasil pengolahan data. Solusi dari masalah ini adalah penerapan *Elasticsearch Logstash Kibana Stack* (ELK Stack) pada sistem Portal PPSDM. ELK Stack melakukan indeks terhadap data yang masuk. Data yang terindeks di dalam ELK Stack berguna untuk mempercepat pengolahan data. Hasil akhir dari penerapan ELK Stack adalah dengan berkurangnya latensi waktu yang diperlukan dalam pengolahan data.

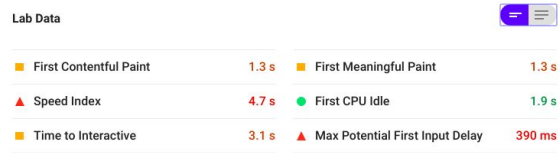
Keywords—*Elasticsearch, Logstash, Kibana, E-Learning, Indexing*

I. PENDAHULUAN

Portal Pengembangan dan Pembinaan Sumber Daya Manusia (PPSDM) merupakan *e-learning* yang digunakan oleh Bidang Pengembangan dan Pembinaan Sumber Daya Manusia (PPSDM) pada Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah (LKPP). Portal PPSDM digunakan sebagai sarana pelatihan dan sertifikasi peserta Pengadaan Barang/Jasa (PBJ). Sistem ini memiliki sekitar 150 ribu pengguna aktif dan mempunyai sekitar 40 aktor dengan hak akses yang bermacam-macam

Portal PPSDM memiliki basis data dengan volume data yang besar. Adapun data – data yang disimpan dalam basis data Portal PPSDM berupa : dokumen pembelajaran, video pembelajaran, audio pembelajaran, data peserta, nilai peserta, dan sertifikat tiap peserta. Seluruh data ini tidak hanya disimpan, namun juga diolah agar menjadi sebuah informasi. Contoh informasi yang didapatkan dari pengolahan data ini antara lain : pelatihan dengan kelulusan terbanyak, sertifikasi dengan kelulusan terbanyak, sepuluh peserta dengan nilai pelatihan terbaik. Dalam Portal PPSDM, kecepatan pengolahan data merupakan hal yang sangat diperlukan. Kecepatan pengolahan data sangat berkaitan erat dengan keakuratan data, karena data yang tidak

akurat dapat disebabkan oleh waktu jeda pengolahan data yang lama. Waktu jeda yang lama dapat menimbulkan informasi yang tidak valid. Informasi yang tidak valid akan berpengaruh pada ketepatan pengambilan keputusan yang dilakukan oleh pengelola Portal PPSDM. Salah satu contoh halaman dengan load yang lama dapat dilihat pada **gambar 1**.



Gambar 1. Hasil pengujian *google page speed*

Solusi untuk menangani waktu jeda yang lama adalah dengan penerapan Elasticsearch Logstash Kibana Stack (ELK Stack) pada Portal PPSDM. ELK Stack merupakan sekumpulan aplikasi *open source* yang digunakan dalam pengolahan data dengan volume yang besar. ELK Stack terdiri dari tiga aplikasi *open source* yaitu : Elasticsearch, Logstash, dan Kibana. ELK Stack digunakan untuk menganalisa data. Menurut Lexy J. Melong [1], analisis data terdiri dari dua proses yaitu mengatur urutan data, dan mengorganisasikannya ke dalam suatu pola, kategori, dan satuan uraian dasar. Tahapan analisis data ini sama seperti yang dilakukan oleh ELK Stack. Pertama, Logstash melakukan penarikan data dari basis data yang dituju. Data yang telah ditarik diuraikan dan diurutkan sedemikian rupa agar dapat disimpan. Setelah itu, Elasticsearch melakukan indeks data. Data yang telah terindeks akan diolah menjadi visualisasi pada Kibana.

Untuk membantu pembaca memahami isi dari tiap bab yang disampaikan, berikut penjelasan dari masing – masing bab : bab I berisi mengenai pokok bahasan makalah, bab II berisi mengenai teori atau penelitian sebelumnya yang digunakan sebagai acuan, bab III berisi mengenai proses analisis dan perancangan yang dilakukan selama membuat sistem ini, bab IV berisi mengenai hasil penerapan dari sistem yang telah dibuat, bab V berupa penjelasan singkat mengenai seberapa terselesaikannya masalah utama dengan solusi yang diusulkan, bab VI berisi mengenai daftar referensi yang diambil guna mendukung argumen dari usulan solusi yang ditawarkan.

II. LANDASAN TEORI

2.1 Elasticsearch

Elasticsearch merupakan sebuah mesin pencari dan analisis untuk semua jenis data [2]. Pada Elasticsearch terjadi suatu proses yang dinamakan *Data Ingestion*. Pada proses ini data mentah diolah melalui tahapan penguraian, normalisasi, dan diperkaya dengan data tambahan sebelum dilakukan proses *indexing*. *Indexing* inilah yang menjadi kunci keberhasilan Elasticsearch dalam melakukan pencarian data dengan cepat. Selain itu, Elasticsearch merupakan alat yang bersifat *scalable*, yang berarti dapat dilakukan penyesuaian tergantung dari kebutuhan sistem.

Adapun contoh penerapan Elasticsearch pada bidang edukasi yang dilakukan oleh [3] membahas mengenai pemanfaatan Elasticsearch dalam mencari tugas akhir dari para mahasiswa Politeknik Negeri Madiun (PNM). Adapun penelitian tersebut bertujuan untuk membantu mahasiswa lain yang membutuhkan referensi dalam tugas akhir. Penelitian tersebut mengkombinasikan *framework* Laravel dengan Elasticsearch. Elasticsearch berfungsi sebagai sebuah basis data *NoSQL* yang akan meng-*indexing* semua data dari aplikasi yang telah dibuat dengan Laravel. Selain mampu mempercepat pemrosesan data penelitian tersebut membuktikan bahwa Elasticsearch dapat diintegrasikan dengan berbagai macam sistem, salah satunya sistem dengan *framework* Laravel dan struktur MVC .

Adapun [4] melakukan penelitian untuk optimalisasi penggunaan *query* dengan menerapkan Elasticsearch guna membantu pengguna dalam mencari obat berdasarkan kata kunci yang dicari. Penelitian tersebut memanfaatkan API yang telah tersedia di dalam Elasticsearch untuk mengolah data obat yang ada, sehingga dapat mengurangi waktu jeda dalam pemrosesan data.

2.2 Logstash

Logstash merupakan salah satu produk utama dari Elastic Stack, Logstash digunakan untuk mengumpulkan dan mengolah data sebelum diteruskan ke Elasticsearch [2]. Data yang diterima Logstash dapat diperoleh melalui basis data, Apache Kafka, Amazon S3, dan Beats. Data – data ini nantinya akan diteruskan ke Elasticsearch untuk dilakukan *indexing*. Untuk memperoleh input data

yang tepat, Logstash memerlukan sebuah dokumen konfigurasi. Dokumen konfigurasi Logstash terdiri dari 3 bagian yaitu *input*, *filter*, dan *output* (lihat **gambar 2**). Tiap bagian konfigurasi memiliki *plugin*-nya masing-masing yang telah disediakan. Plugin ini berguna untuk membantu Logstash mengambil data sesuai kebutuhan, seperti menerapkan pengambilan terjadwal, mengeluarkan keluaran *log* yang disimpan dalam sebuah dokumen, dan sebagainya

```

1 # This is a comment. You should use comments to describe
2 # parts of your configuration.
3 input {
4   ...
5 }
6
7 filter {
8   ...
9 }
10
11 output {
12   ...
13 }

```

Gambar 2. Struktur konfigurasi Logstash

Adapun penelitian Arifin yang melakukan implementasi Logstash untuk menarik data log yang ada pada server untuk membantu administrator dalam melakukan monitoring server [5]. Data yang telah ditarik dari log server menggunakan Logstash kemudian diuraikan dan diteruskan ke Elasticsearch. Setelah itu data ini digunakan untuk membuat *dashboard* administrator yang menampilkan data seperti berapa banyak user yang gagal melakukan *login*, berapa user yang dapat melakukan *login*, dan berapa persentase keseluruhannya. Penarikan *log* server yang dilakukan oleh Logstash ini mampu membuat kegiatan monitoring *log* server menjadi lebih efisien .

2.3 Kibana

Kibana merupakan sebuah alat visualisasi dan manajemen data untuk Elasticsearch [2]. Data yang divisualisasikan dapat berupa grafik, *metric*, tabel, ataupun gambar. Kibana juga menyediakan fitur *dashboard*, fitur ini berfungsi untuk mengumpulkan data yang telah divisualisasikan ke dalam satu halaman *dashboard*. *Dashboard* yang telah dibuat dapat di-*embed* ke dalam tampilan sistem.

Pada penelitian Shah yang melakukan analisis sosial media seperti twitter menggunakan Elasticsearch, Logstash dan Kibana. Sistem yang mereka buat memanfaatkan kemampuan Kibana

untuk membuat visualisasi dari data yang telah di-*index* oleh Elasticsearch [6]. Pada penelitian ini Elasticsearch, Logstash dan Kibana dikonfigurasi untuk mengambil data secara *realtime*. Sehingga hal ini membuat statistik yang dibuat Kibana menjadi lebih efisien dan akurat.

III. METODOLOGI

Dalam makalah ini dilakukan metodologi sebagai berikut :

- Analisis
- Konfigurasi Logstash
- Konfigurasi Elasticsearch
- Konfigurasi Kibana
- *Indexing* data sistem
- Perancangan *dashboard*
- Perancangan Infrastruktur *High Availability*

3.1 Analisis

Analisis yang dilakukan dalam proses pengembangan sistem ini berupa :

- **Wawancara dengan stakeholder** – *stakeholder* dari proyek ini adalah LKPP di bidang PPSDM. Adapun pertanyaan – pertanyaan yang diajukan saat wawancara dengan *stakeholder* diantaranya sebagai berikut :
 - o Apa saja kendala yang dihadapi dalam Portal PPSDM ?
 - o Bagaimana ekspektasi yang diharapkan pada sistem yang akan dibuat ?
 - o Fitur apa saja yang dibutuhkan dalam sistem yang akan dikembangkan ?
- **Analisis dokumen teknis pada pengembangan sebelumnya** – Portal PPSDM adalah sistem yang telah digunakan oleh peserta LKPP PPSDM. Sehingga untuk mempersingkat waktu analisis yang dibutuhkan dalam mengembangkan Portal PPSDM, tim analisis membutuhkan dokumen teknis sistem sebelumnya untuk membantu proses *mapping* fitur dan membantu analisis.
- **Analisis mengenai masalah yang dihadapi dan solusi yang digunakan** – solusi yang diambil dalam menyelesaikan masalah pengolahan data dengan waktu jeda yang lambat

adalah dengan penerapan ELK Stack. Solusi ini telah dipertimbangkan dengan alasan sebagai berikut :

- ELK Stack dapat diatur sesuai kebutuhan
- Waktu tunggu pengolahan data yang cepat
- Dapat divisualisasikan dengan mudah hanya dengan *drag & drop*
- Dapat melakukan denormalisasi data dari database yang berbeda
- Dapat melakukan *full text search*

Adapun implementasi ELK Stack ini dibuat pada spesifikasi sebagai berikut :

- Server Ubuntu 16.04
- Java versi 11.04

Optimasi dengan ELK Stack dapat dilakukan dengan beberapa langkah sebagai berikut :

- Membuat script konfigurasi pada Logstash
- Menjalankan script Logstash untuk menarik data dari basis data
- Data yang telah ditarik akan di *index* oleh Elasticsearch
- Hasil data yang telah di *index* akan digunakan sebagai masukan untuk Kibana dalam membuat visualisasi data

3.2 Konfigurasi Logstash

Sebelum melakukan *indexing* data, Logstash memerlukan konfigurasi yang disimpan dalam dokumen berekstensi *yml*. Dokumen ini berisi konfigurasi dasar seperti *path data*, *path log*, dan banyak *pipeline* yang digunakan (lihat **gambar 3**).

```
1 path.data: /var/lib/logstash
2 pipeline.workers: 4
3 path.logs: /var/log/logstash
```

Gambar 3. Dokumen Konfigurasi Logstash.yml

Untuk mendapatkan dan menstandarkan data dari basis data yang diinginkan, Logstash memerlukan konfigurasi dalam menarik, memfilter, dan membuat keluaran data (lihat **gambar 4**).

```
input {
  jdbc {
    jdbc_driver_library => ""
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/coba_moodle"
    jdbc_user => root
    jdbc_password => "asd12341ag1"
    jdbc_pooling_enabled => true
    jdbc_page_size => 50000
    schedule => "0 23 * * *"
    use_column_value => true
    tracking_column => "updated_at"
    tracking_column_type => "timestamp"
    statement => "SELECT mdl_feedback_value.id as id,
    mdl_course.fullname as course_name,
    mdl_course.shortname as course_shortname,
    mdl_course.idnumber as course_id_number,
    mdl_user.username,
    mdl_user.firstname,
    mdl_user.lastname,
    mdl_feedback.name as feedback_name,
    mdl_feedback_item.name as item_name,
    mdl_feedback_value.value as feedback_value,
    FROM_UNIXTIME(
      greatest(
        mdl_feedback.timemodified,
        mdl_feedback_completed.timemodified,
        mdl_course.timemodified,
        mdl_user.timemodified
      )
    ) as updated_at
    FROM mdl_feedback_value
    JOIN mdl_feedback_item ON mdl_feedback_item.id = mdl_feedback_value.item
    JOIN mdl_feedback_completed ON mdl_feedback_completed.id = mdl_feedback_value.completed
    JOIN mdl_feedback ON mdl_feedback.id = mdl_feedback_item.feedback
    JOIN mdl_user ON mdl_user.id = mdl_feedback_completed.userid
    JOIN mdl_course ON mdl_course.id = mdl_feedback.course
    where FROM_UNIXTIME(
      greatest(
        mdl_feedback.timemodified,
        mdl_feedback_completed.timemodified,
        mdl_course.timemodified,
        mdl_user.timemodified
      )
    ) > :sql_last_value
    order by updated_at"
  }
}

output {
  stdout {
    codec => json_lines
  }
  elasticsearch {
    action => "update"
    doc_as_upsert => true
    index => "activity_evaluations"
    user => elastic
    password => asd12341ag1
    document_id => "%{id}"
    hosts => "http://10.0.0.139:9200"
  }
}
```

Gambar 4. Dokumen Konfigurasi Logstash.conf

3.3 Konfigurasi Elasticsearch

Setelah melakukan konfigurasi Logstash, langkah selanjutnya adalah melakukan konfigurasi terhadap Elasticsearch. Konfigurasi ini mengatur mengenai *cluster*, *node*, *path data*, dan *x-pack* (lihat **gambar 5**). Adapun kegunaan dari masing – masing konfigurasi tersebut yaitu :

- **Cluster** – berisi pengaturan mengenai manajemen sekumpulan *nodes*.
- **Node** – berisi pengaturan mengenai jalur lalu lintas data yang digunakan, mulai dari nama *node*, dan *master node* yang digunakan.
- **Path Data** – berisi *path* yang digunakan untuk menyimpan data.
- **X-pack** – berisi pengaturan mengenai manajemen pengguna pada Elasticsearch.

```

1 cluster.name: ppsdm-elasticsearch
2 node.name: ppsdm-node
3 #node.attr.rack: r1
4 path.logs: /var/log/elasticsearch
5 network.host: 0.0.0.0
6 transport.port: 9300
7 client.transport.ignore_cluster_name: true
8 cluster.initial_master_nodes: ["ppsdm-node"]
9 xpack.security.enabled: true
10 xpack.security.transport.ssl.enabled: true
11 xpack.security.audit.enabled: true
12 xpack.security.transport.ssl.verification_mode: certificate
13 xpack.security.transport.ssl.keystore.path: certs/elastic-certificates.p12
14 xpack.security.transport.ssl.truststore.path: certs/elastic-certificates.p12
15 path.repo: ["/home/vagrant/elastic"]

```

Gambar 5. Dokumen Konfigurasi Elasticsearch.yml

Adapun untuk mengkonfigurasi *x-pack*, diperlukan membuat sertifikat yang mempunyai format p12. *X-pack* dapat digunakan untuk melakukan manajemen otoritas dari tiap pengguna.

3.4 Konfigurasi Kibana

Konfigurasi terakhir yang perlu dilakukan adalah konfigurasi Kibana. Sama halnya dengan Elasticsearch dan Logstash, Kibana juga memerlukan konfigurasi yang disimpan ke dalam dokumen berekstensi yml. Adapun konfigurasi ini berisi mengenai Elasticsearch *username* dan *password*, detail dari server yang digunakan, *x-pack*, *logging* (lihat gambar 6).

```

1 server.port: 5601
2 server.host: "10.0.0.139"
3 elasticsearch.hosts: ["http://10.0.0.139:9200"]
4 elasticsearch.username: "username"
5 elasticsearch.password: "password"
6 logging.dest: /var/log/kibana/kibana.log
7 logging.verbose: true
8 xpack.security.encryptionKey: "Jsjjsj123h3u54uwqwLUYyjsjshawEeRT1="
9 xpack.security.audit.enabled: true

```

Gambar 6. Dokumen Konfigurasi Kibana.yml

3.5 Indexing Data Sistem

Langkah selanjutnya adalah membuat *indexing* data pada Kibana. Hal ini dapat dilakukan melalui fitur *index patterns* yang terdapat pada menu *management* (lihat gambar 7).

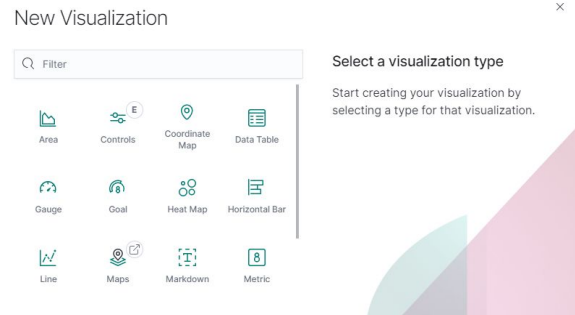


Gambar 7. Menu management

3.6 Perancangan Dashboard

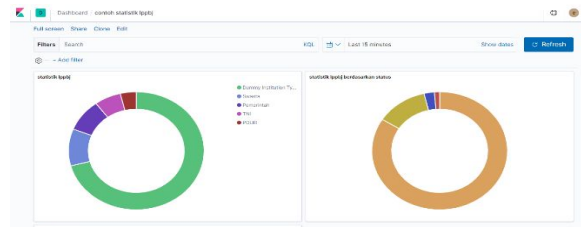
Langkah terakhir adalah dengan membuat visualisasi yang nantinya akan diletakkan ke dalam

dashboard. Statistik yang ingin dibuat juga dapat disesuaikan dengan kebutuhan pengguna. Pada gambar 8 terlihat beberapa statistik yang dapat digunakan untuk membuat sebuah visualisasi.



Gambar 8. Pilihan Statistik Kibana

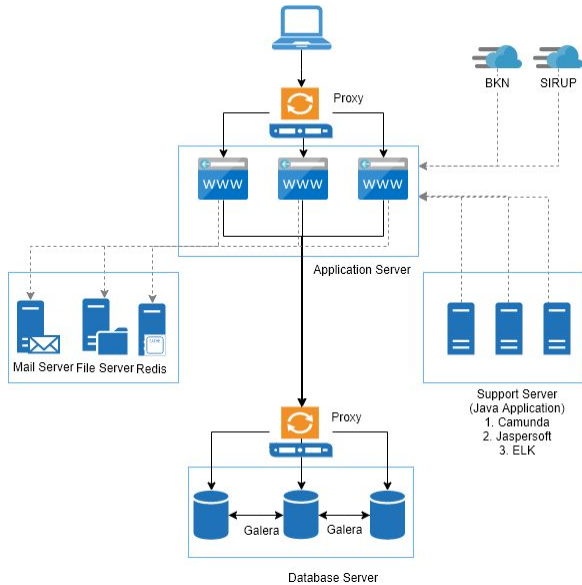
Setelah berhasil membuat visualisasi, langkah selanjutnya adalah membuat *dashboard*. *Dashboard* berguna untuk menyajikan kumpulan visualisasi pada satu halaman (lihat gambar 9).



Gambar 9. Halaman Dashboard

3.7 Perancangan Infrastrukturu High Availability

Untuk mencapai sebuah sistem yang *high availability* pada Portal PPSDM setidaknya diperlukan sebanyak ± 14 *virtual machine* (lihat gambar 10). Selain untuk membuat server dengan pelayanan yang stabil, hal ini dilakukan agar beban kerja ELK Stack dapat terbagi sehingga tidak mengganggu proses pada VM yang lain. Dengan adanya pembagian server menjadi 14 VM, akan lebih mudah melakukan konfigurasi pada ELK Stack.



Gambar 10. Rancangan Infrastruktur

Adapun penjelasan dari infrastruktur di atas (lihat **gambar 10**) antara lain :

- **Application Server** - digunakan agar ketika user mengakses Portal PPSDM diarahkan ke server yang loadnya sedikit
- **Database Server** - database server ini harus ganjil minimal 3. Karena akan implementasi high-availability databases menggunakan *galera cluster*, dimana semua database server akan menjadi server aktif yang dapat melayani *read/write* ke database. Database menggunakan mariaDB
- **Support Server** - server ini menyimpan aplikasi java pendukung, yakni
 - Camunda - workflow application
 - Jasper report - pembuatan laporan
 - ELK Stack - Elasticsearch, Logstash, Kibana untuk *indexing* dan visualisasi data
- **Mail Server** - server yang digunakan untuk melayani pengiriman email

- **File Server** - server yang digunakan untuk melayani *read/write* dokumen dari dan ke Portal PPSDM
- **Redis** - server yang digunakan untuk menyimpan *cache* server
- **Proxy** - untuk mengatur user masuk ke server aplikasi dan basis data yang memiliki antrian tercepat

IV. HASIL DAN PENGUJIAN

Ada beberapa cara untuk memanfaatkan kemampuan dari ELK Stack. Beberapa pemanfaatan ELK Stack antara lain :

- Pemanfaatan API ELK Stack untuk mencari sebuah data
- Pemanfaatan Kibana untuk membuat *dashboard* yang diperlukan untuk melakukan monitoring data

API yang disediakan ELK Stack dapat diolah menjadi berbagai macam format. Format *default* yang akan ditampilkan saat mengambil data adalah format JSON (lihat **gambar 11 - 12**).

```
GET /activity_grades/_search?q=first_name:"Mochamad Noer Isnin "
```

Gambar 11. Perintah untuk mengambil data

Hasil JSON dari ELasticsearch mempunyai beberapa parameter antara lain :

- **took** – waktu yang diperlukan untuk mengeksekusi perintah yang dijalankan (millisecond).
- **timed_out** – parameter yang menunjukkan apakah perintah yang dijalankan mengalami *timed out* atau tidak.
- **_shards** – parameter yang menunjukkan berapa banyak parameter pencarian yang berhasil dan tidak.
- **hits** – hasil dari pencarian.
- **hits.total** – total data yang cocok dengan pencarian yang dijalankan.
- **hits.hits** – semua data hasil dari pencarian dengan paginasi *default* 10 data/halaman.

```

1  {
2    "took" : 90,
3    "timed_out" : false,
4    "_shards" : {
5      "total" : 1,
6      "successful" : 1,
7      "skipped" : 0,
8      "failed" : 0
9    },
10   "hits" : {
11     "total" : {
12       "value" : 39,
13       "relation" : "eq"
14     },
15     "max_score" : 17.638153,
16     "hits" : [
17       {
18         ...
19       }
20     ]
21   }
22 }

```

Gambar 12. Hasil pengambilan dalam format JSON

Selain pengolahan data dengan format JSON, Elasticsearch juga menyediakan penggunaan *query* pada data yang telah di-*index* (lihat gambar 13 – 14).

```

POST _sql?format=txt
{
  "query" : "select course_name as nama_pelatihan,
            raw_grade as nilai from activity_grades where
            first_name='Mochamad Noer Isnin'"
}

```

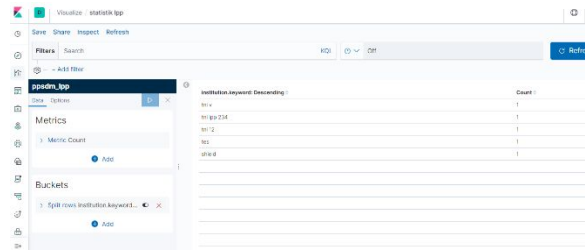
Gambar 13. Perintah pengambilan *query* dengan format TXT

1						nama_pelatihan	nilai
2							
3	2/7_12/7_LPKN-BIG	Hotel	Sahira_Pelatihan	PBJ	Tingkat Dasar		100
4	2/7_12/7_LPKN-BIG	Hotel	Sahira_Pelatihan	PBJ	Tingkat Dasar		176
5	2/7_12/7_LPKN-BIG	Hotel	Sahira_Pelatihan	PBJ	Tingkat Dasar		null
6	2/7_12/7_LPKN-BIG	Hotel	Sahira_Pelatihan	PBJ	Tingkat Dasar		null
7	2/7_12/7_LPKN-BIG	Hotel	Sahira_Pelatihan	PBJ	Tingkat Dasar		null
8	2/7_12/7_LPKN-BIG	Hotel	Sahira_Pelatihan	PBJ	Tingkat Dasar		53
9	2/7_12/7_LPKN-BIG	Hotel	Sahira_Pelatihan	PBJ	Tingkat Dasar		81
10	2/7_12/7_LPKN-BIG	Hotel	Sahira_Pelatihan	PBJ	Tingkat Dasar		80
11	2/7_12/7_LPKN-BIG	Hotel	Sahira_Pelatihan	PBJ	Tingkat Dasar		null

Gambar 14. Hasil pengambilan *query* dengan format TXT

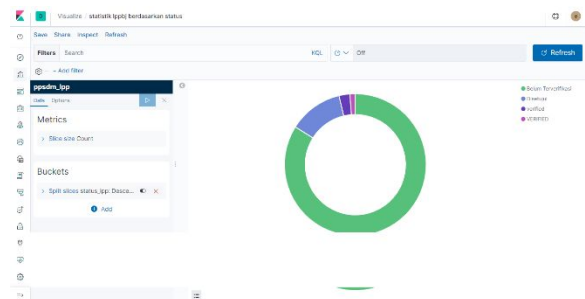
Dengan data yang telah di-*index* oleh Elasticsearch, kita dapat membuat visualisasi dari data tersebut menggunakan Kibana. Beberapa hasil dari visualisasi data Portal PPSDM yang telah dibuat antara lain :

- Statistik LPPBJ berdasarkan instansi (lihat gambar 15)



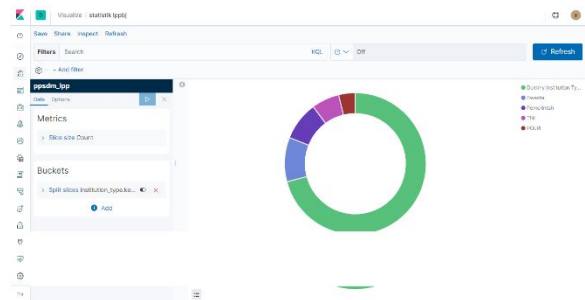
Gambar 15. Data table LPPBJ berdasarkan instansi

- Statistik LPPBJ berdasarkan status (lihat gambar 16)



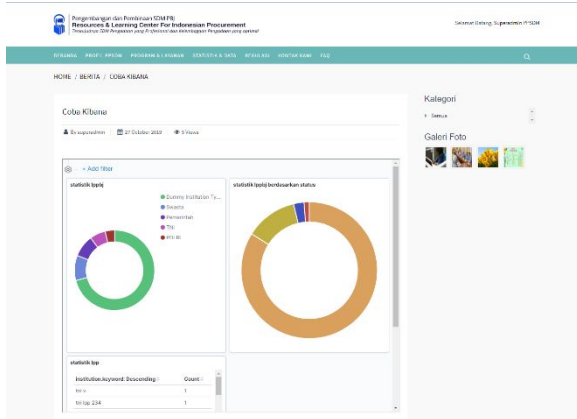
Gambar 16. Statistik LPPBJ berdasarkan status

- Statistik LPPBJ berdasarkan jenis instansi (lihat gambar 17)



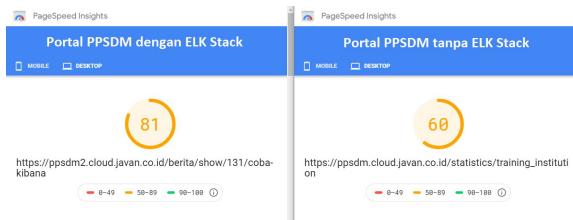
Gambar 17. Statistik LPPBJ berdasarkan jenis instansi

Adapun cara penggunaan statistik yang ingin ditampilkan hanya dengan melakukan *embed* melalui text area yang telah disediakan. Hasil dari dashboard kibana yang di-*embed* dapat dilihat pada gambar 18.

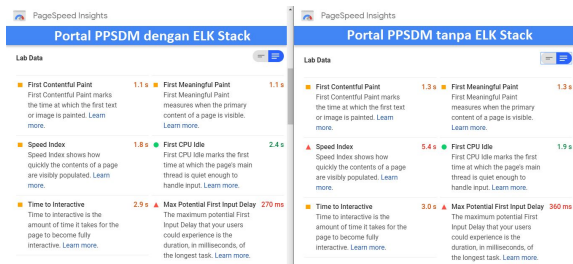


Gambar 18. Hasil penggunaan dashboard Kibana

Setelah melakukan penerapan ELK Stack, sistem dibandingkan dengan menggunakan *google page speed*. Hasil dari pengujian ini dapat dilihat pada gambar 19 - 20. Pengujian ini membuktikan bahwa ELK Stack telah berhasil mengurangi latensi waktu yang diperlukan dalam pengolahan data sekitar 2.5 detik.



Gambar 19. Hasil pengujian dengan *google page speed*



Gambar 20. Hasil pengujian dengan *google page speed*

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Implementasi ELK Stack pada Portal PPSDM untuk membantu pengolahan dan penyajian data mampu mengurangi latensi waktu yang diperlukan sekitar 2,5 detik. Hal ini akan membantu jalannya proses monitoring dan evaluasi yang terjadi selama proses pelatihan dan sertifikasi yang terjadi pada

Portal PPSDM. Selain itu konfigurasi dan juga infrastruktur yang tepat sangat diperlukan untuk membuat sistem dapat bersifat *high availability*.

5.2 Saran

Implementasi ELK Stack yang dilakukan dalam makalah ini masih dapat dikembangkan lebih lanjut guna mengoptimasi penggunaan *query* pada sistem. Beberapa saran pengembangan dari implementasi ELK Stack sebagai berikut :

- Implementasi ELK Stack pada *website* jual beli *online* yang mempunyai data yang sangat besar sebagai sarana monitoring dan pengganti *query* konvensional
- Implementasi ELK Stack sebagai sarana monitoring *log server* yang digunakan
- Implementasi ELK Stack pada sistem FaQ atau forum sebagai pengganti *query* konvensional

VI. REFERENSI.

- [1] L. J. Moleong, Metodologi Penelitian Kualitatif, Bandung: Rosda, 2007.
- [2] Elastic, "What is Elasticsearch ?," 2019. [Online]. Available: <https://www.elastic.co/what-is/elasticsearch> . [Accessed 2019].
- [3] A. P. Atmajaa and S. V. Yulianto, "Pemanfaatan Elasticsearch untuk Temu Kembali Informasi Tugas Akhir," *Jurnal Nasional Teknologi dan Sistem Informasi*, pp. 160-167, 2018.
- [4] H. A. C. Bhadane, D. U. Mody and P. R. S. Shah, "Use of Elastic Search for Intelligent," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. III, no. 6, pp. 2231-2307, 2014.
- [5] M. N. Arifin, Sugartowo and E. Susilowati, "Desain dan Implementasi Log Event Management Server Menggunakan Elasticsearch Logstash Kibana (ELK STACK)," *Seminar Nasional Sains dan Teknologi 2018*, 2018.
- [6] N. Shah, D. Willick and V. Mago, "A framework for social media data analytics using Elasticsearch and Kibana," *Springer Nature 2018*, 2018.