

paper_implementasi_elasticsearch_logstash_kibana_stack_pada_elearning

by Elang Putra Sartika

Submission date: 19-Nov-2019 05:24PM (UTC+0700)

Submission ID: 1215989463

File name: Makalah_PALING_FIX_FIX.pdf (875.29K)

Word count: 2487

Character count: 16137

Implementasi Elasticsearch Logstash Kibana Stack pada E-Learning Lembaga Kebijakan Pengadaan Barang/Jasa (LKPP) Bidang Pengembangan dan Pembinaan Sumber Daya Manusia (PPSDM)

(Studi kasus : Bidang PPSDM pada LKPP)

Abstrak—*E-learning* merupakan sebuah sarana pembelajaran yang sangat memudahkan bagi peserta *e-learning*. Tidak hanya praktis, *e-learning* juga dapat membantu pemateri menyelenggarakan sebuah pelatihan ataupun sertifikasi. Namun, *e-learning* yang telah memiliki banyak kelas dan peserta seringkali menghadapi beberapa masalah yang berkaitan dengan pengolahan data. Kasus ini terjadi pada *e-learning* LKPP PPSDM. Hal ini disebabkan oleh volume data yang sangat besar, sehingga menyebabkan pengolahan data memerlukan waktu yang sangat lama. Masalah ini membuat pemateri *e-learning* kesulitan dalam melihat statistik dari pelatihan yang telah diselenggarakan. Solusi dari masalah ini adalah dengan penerapan ELK Stack pada *e-learning*. ELK Stack akan mengurangi latensi waktu yang diperlukan dalam mengolah data, hal ini dapat terjadi karena data yang telah masuk ke dalam ELK Stack telah di-*index*. Sehingga pengambilan dan pengolahan data menjadi lebih cepat. Selain itu penerapan infrastruktur yang tepat juga sangat berpengaruh demi tercapainya optimasi pengambilan dan pengolahan data.

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

I. PENDAHULUAN

E-learning merupakan kepanjangan dari *electronic learning*, yang berarti sebuah aktivitas pembelajaran yang dilakukan dengan memanfaatkan teknologi yang ada. Dalam penerapannya, *e-learning* memerlukan tempat penyimpanan yang sangat besar. Hal ini

dikarenakan banyak sekali data yang diperlukan untuk membuat sebuah *e-learning* berhasil. Data – data yang diperlukan itu seperti : dokumen pembelajaran, video pembelajaran, audio pembelajaran, data peserta, nilai peserta, dst. Seluruh data ini tidak hanya disimpan, namun juga diolah agar menjadi sebuah informasi.

Dalam pengolahannya, keakuratan dan kecepatan pengolahan data merupakan dua hal pokok yang perlu dicapai oleh sebuah *e-learning*. Data yang tidak akurat dan pengolahan data yang lambat akan menimbulkan banyak masalah untuk sebuah *e-learning*. Adapun masalah yang mungkin akan muncul dikarenakan kedua hal ini antara lain : pengambilan keputusan yang salah, informasi yang tidak valid, waktu tunggu yang terlalu lama, dst. Masalah ini telah dialami juga oleh Lembaga Kebijakan Pengadaan Barang/Jasa Pemerintah (LKPP).

LKPP memiliki sebuah bidang yang bertugas untuk melakukan pembinaan dan pengembangan terhadap sumber daya manusia yang mereka miliki. Bidang ini bernama bidang PPSDM. Bidang ini bertugas sebagai penyelenggara pelatihan, ujian, sertifikasi untuk para peserta yang terlibat di dalamnya. Saat ini *e-learning* PPSDM memiliki data yang sangat besar dengan lebih dari 150 ribu pengguna aktif. Hal ini membuat *e-learning* PPSDM mengalami waktu tunggu pengolahan data yang terbilang lambat dan data yang tidak akurat.

1.1 Inti Masalah

Dari kasus di atas didapatkan inti masalah sebagai berikut :

- Data yang banyak membuat waktu tunggu pengolahan data menjadi lambat
- Hasil pengolahan data yang tidak akurat

1.2 Rumusan Masalah

Untuk menjaga fokus dari makalah ini, adapun rumusan masalah yang perlu diselesaikan sebagai berikut :

- Bagaimana cara untuk mempersingkat waktu tunggu pengolahan data dengan data yang sangat besar ?
- Bagaimana cara mendapatkan hasil pengolahan yang akurat dari data yang jumlahnya sangat banyak ?

1.3 Solusi

Terdapat beberapa solusi yang dapat diterapkan pada sistem untuk menyelesaikan masalah yang sedang dihadapi ini, salah satunya adalah implementasi Elasticsearch Logstash Kibana Stack (ELK Stack) pada *e-learning*. ELK Stack merupakan sekumpulan aplikasi *open source* yang digunakan dalam mengolah data yang berjumlah banyak. Adapun ELK Stack terdiri dari tiga aplikasi *open source* yaitu Elasticsearch, Logstash, dan Kibana. ELK Stack digunakan untuk menganalisa data. Menurut Lexy J. [6](#) *elongs*, analisis data terdiri dari dua proses yaitu mengatur urutan data, dan mengorganisasikannya ke dalam suatu pola, kategori, dan satuan uraian dasar [1]. Adapun penjelasan dari masing – masing aplikasi ELK Stack antara lain :

- **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.
- **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.

- **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*.

Adapun ELK Stack menjadi salah satu solusi penyelesaian masalah ini karena beberapa 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*

1.4 Cara Kerja

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

1.5 Sistematika Makalah

Untuk membantu pembaca memahami isi dari tiap bab yang disampaikan, berikut penjelasan dari masing – masing bab :

- **I. Pendahuluan** – berisi mengenai pokok bahasan makalah.
- **II. Landasan Teori** – berisi mengenai teori atau penelitian sebelumnya yang digunakan sebagai acuan.

12

- **III. Analisis dan Perancangan** – berisi mengenai proses analisis dan perancangan yang dilakukan selama membuat sistem ini.
- **IV. Hasil** – berisi mengenai hasil penerapan dari sistem yang telah dibuat.
- **V. Kesimpulan** – berupa penjelasan singkat mengenai seberapa terselesaikannya masalah utama dengan solusi yang diusulkan.
- **VI. Referensi** – berisi mengenai daftar referensi yang diambil guna mendukung argumen dari usulan solusi yang ditawarkan.

11

II. LANDASAN TEORI

2.1 Elasticsearch

Elasticsearch merupakan alat yang digunakan untuk mengoptimasi pencarian dengan data yang banyak. Proses *indexing* data yang dilakukan oleh Elasticsearch membuat pengolahan data menjadi lebih cepat. Elasticsearch merupakan alat yang bersifat *scalable*, yang berarti dapat dilakukan penyesuaian tergantung dari kebutuhan sistem.

Penelitian Ardian Prima Atmaja dan Susilo Veri Yulianto (2018) mengimplementasikan Elasticsearch pada bidang edukasi. Penelitian ini membahas mengenai pemanfaatan Elasticsearch dalam mencari tugas akhir dari para mahasiswa Politeknik Negeri Madiun (PNM). Adapun penelitian ini bertujuan untuk membantu mahasiswa lain yang membutuhkan referensi dalam tugas akhirnya. Dalam penelitiannya Ardian mengkombinasikan *framework* Laravel dengan Elasticsearch. Sistem yang dibangun oleh Ardian memanfaatkan Elasticsearch sebagai sebuah basis data *NoSQL* yang akan meng-*indexing* semua data dari aplikasi yang telah dibuat dengan Laravel. Selain mampu mengurangi latensi waktu, hal ini membuktikan bahwa Elasticsearch dapat diintegrasikan dengan *framework* Laravel dengan struktur MVC [3].

Penelitian Badhane, et. Al (2014) mengoptimalkan penggunaan *query* dengan mengimplementasikan Elasticsearch untuk membantu pengguna dalam mencari obat berdasarkan kata kunci yang dicari. Penelitian ini memanfaatkan API yang telah tersedia di dalam Elasticsearch untuk mengolah data obat yang ada. Selain itu penelitian ini juga menyesuaikan konfigurasi Elasticsearch yang diperlukan, sehingga didapat hasil optimal yang mampu mengurangi latensi waktu dalam pengolahan data [4].

2.2 Logstash

Logstash digunakan untuk menarik data dari basis data, 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 1). 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 1. Struktur konfigurasi Logstash

Penelitian Arifin, et. Al (2018) mengimplementasikan Logstash untuk menarik data log yang ada pada server untuk membantu administrator dalam melakukan monitoring server. 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 [5].

2.3 Kibana

Kibana digunakan untuk memvisualisasikan data yang telah di – *index* oleh Elasticsearch. Data yang telah divisualisasikan ini dikumpulkan ke dalam sebuah *dashboard* yang dapat di *embed* ke dalam tampilan sistem.

Pada penelitian Shah, et. Al (2018) menganalisis social 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. 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 [6].

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 :
 - Apa saja kendala yang dihadapi dalam *e-learning* ?
 - Bagaimana ekspektasi yang diharapkan pada sistem yang akan dibuat ?
 - Fitur apa saja yang dibutuhkan dalam sistem yang akan dikembangkan ?
 - Dst.
- **Analisis dokumen teknis pada pengembangan sebelumnya – e-learning PPSDM** adalah sistem yang telah digunakan oleh peserta LKPP PPSDM. Sehingga untuk mempersingkat waktu analisis yang dibutuhkan dalam mengembangkan *e-learning*, tim analis membutuhkan dokumen teknis sistem sebelumnya untuk membantu proses *mapping* fitur dan membantu analisis.

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*, banyak *pipeline* yang digunakan, dst (lihat **gambar 2**).

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

Gambar 2. 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 3**).

```
input {
  jdbc {
    jdbc_driver_library => ""
    jdbc_driver_class => "com.mysql.jdbc.Driver"
    jdbc_connection_string => "jdbc:mysql://localhost:3306/coba_modul1"
    jdbc_user => "root"
    jdbc_password => "asd1234567"
    jdbc_staging_enabled => true
    jdbc_persist_options => {}
    schedule => "* * * * *"
    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.timestamp,
    mdl_feedback_completed.timestamp,
    mdl_course.timestamp,
    mdl_user.timestamp
  )
) 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_value.feedback
JOIN mdl_user ON mdl_user.id = mdl_feedback_completed.userid
JOIN mdl_course ON mdl_course.id = mdl_feedback_value.course
WHERE FROM_UNIXTIME(
  greatest(
    mdl_feedback.timestamp,
    mdl_feedback_completed.timestamp,
    mdl_course.timestamp,
    mdl_user.timestamp
  )
) > :sql_last_value
order by updated_at"
}
}

output {
  stdout {
    codec => json_lines
  }
  elasticsearch {
    action => "update"
    doc_as_update => true
    index => "activity_evaluations"
    user => elastic
    password => "asd1234567"
    document_id => "%{id}"
    hosts => "http://10.0.0.135:9200"
  }
}
```

Gambar 3. 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 4**). 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*, *master node* yang digunakan, dst.

- **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 4. 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 5).

```

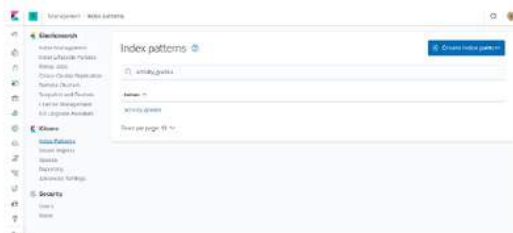
1 server.port: 5601
2 server.host: "10.0.0.139"
3 elasticsearch.hosts: ["http://10.0.0.139:9200"]
4 elasticsearch.username: "username"
5 elasticsearch.password: "password"
6 logging.dest: /var/log/kibana/kibana.log
7 logging.verbose: true
8 xpack.security.encryptionKey: "jsjsj123h3u54uwqwLUYYjsjshakEeRT1-"
9 xpack.security.audit.enabled: true

```

Gambar 5. Dokumen Konfigurasi Kibana.yml

3.5 Indexing Data Sistem

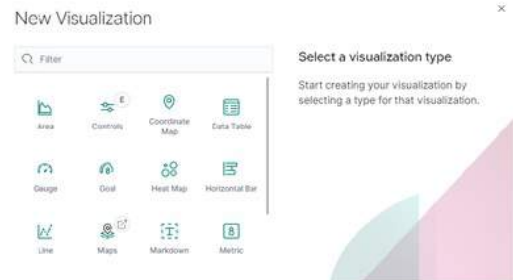
Langkah selanjutnya adalah membuat *indexing* data pada Kibana. Hal ini dapat dilakukan melalui fitur *index patterns* yang terdapat pada menu *management* (lihat gambar 6).



Gambar 6. Menu *management*

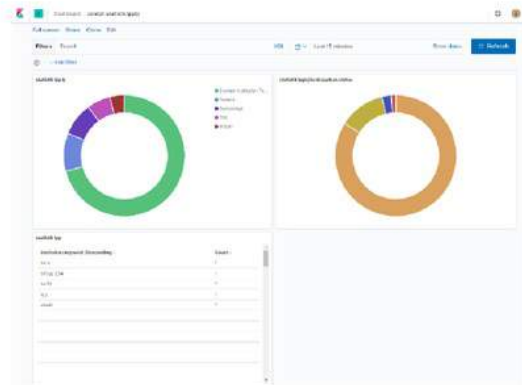
3.6 Perancangan Dashboard

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 7 terlihat beberapa statistik yang dapat digunakan untuk membuat sebuah visualisasi.



Gambar 7. Pilihan Statistik Kibana

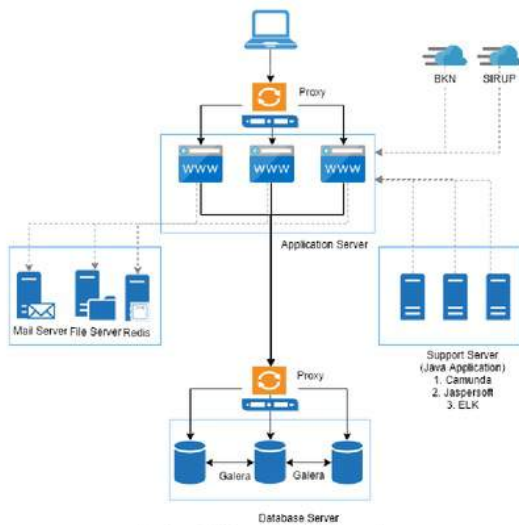
Setelah berhasil membuat visualisasi, langkah selanjutnya adalah membuat *dashboard*. *Dashboard* berguna untuk menyajikan kumpulan visualisasi pada satu halaman (lihat gambar 8).



Gambar 8. Halaman *Dashboard*

3.7 Perancangan Infrastrukturu High Availability

Untuk mencapai sebuah sistem yang *high availability* pada *e-learning* LKPP PPSDM setidaknya diperlukan sebanyak ± 14 *virtual machine* (lihat gambar 9).



Gambar 9. Rancangan Infrastruktur

IV. HASIL

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 10 - 11).

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

Gambar 10. 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 11. Hasil pengambilan dalam format JSON

Selain pengolahan data dengan format JSON, Elasticsearch juga menyediakan penggunaan *query* pada data yang telah di-*index* (lihat gambar 12 – 13). Namun perintah SQL yang dapat dilakukan juga terbatas. Perintah SQL yang dapat digunakan dalam Elasticsearch yaitu :

- **DESCRIBE TABLE**
Berfungsi untuk mendeskripsikan sebuah index.
- **SELECT**
Berfungsi untuk menampilkan data yang tersimpan pada suatu index
- **SHOW TABLES**
Berfungsi untuk menampilkan semua index yang ada.
- **SHOW COLUMNS**
Berfungsi untuk menampilkan semua kolom pada suatu table.
- **SHOW FUNCTIONS**
Berfungsi untuk menampilkan fungsi-fungsi yang tersedia.

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

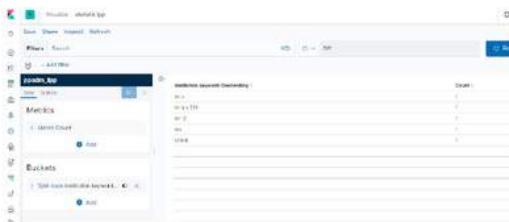
Gambar 12. Perintah pengambilan query dengan format TXT

	nama_pelatihan	nilai
2		
3	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	100
4	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	176
5	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null
6	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null
7	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null
8	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	53
9	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	81
10	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	80
11	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null
12	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	95
13	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	80
14	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null
15	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	80
16	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	100
17	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null
18	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null
19	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	96
20	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	87
21	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	10
22	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null
23	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null
24	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	100
25	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	93
26	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null
27	2/7_12/7_LPKN-BIG Hotel Sahira_Pelatihan PB3 Tingkat Dasar	null

Gambar 13. Hasil pengambilan query dengan format TXT

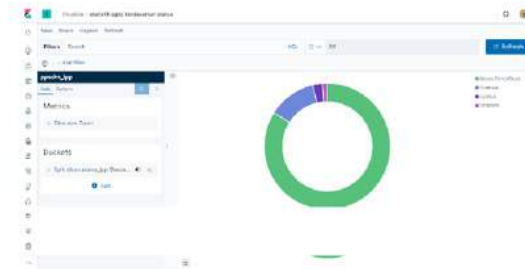
Dengan data yang telah di-index oleh Elasticsearch, kita dapat membuat visualisasi dari data tersebut menggunakan Kibana. Beberapa hasil dari visualisasi data e-learning yang telah dibuat antara lain :

- Statistik LPPBJ berdasarkan instansi (lihat gambar 14)



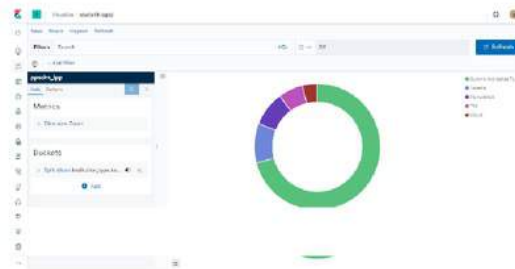
Gambar 14. Data table LPPBJ berdasarkan instansi

- Statistik LPPBJ berdasarkan status (lihat gambar 15)



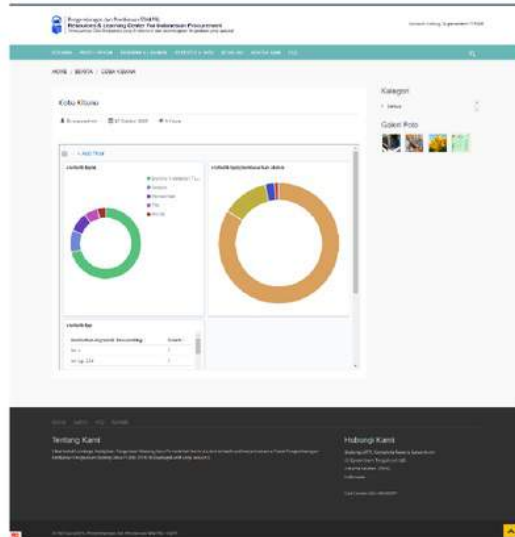
Gambar 15. Statistik LPPBJ berdasarkan status

- Statistik LPPBJ berdasarkan jenis instansi (lihat gambar 16)



Gambar 16. 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 17.



Gambar 17. Hasil penggunaan dashboard Kibana

5.1 Kesimpulan

Implementasi ELK Stack pada *e-learning* untuk membantu pengolahan dan penyajian data mampu mengurangi latensi waktu yang diperlukan dibandingkan dengan menggunakan basis data biasa. Hal ini akan membantu jalannya proses monitoring dan evaluasi yang terjadi selama proses pelatihan dan sertifikasi yang terjadi pada *e-learning* LKPP 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 sistem Universitas untuk membuat *executive summary* dari semua *website* yang dimiliki
- Implementasi ELK Stack sebagai sarana memonitoring *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 Informa*, 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. Arifin, Sugiartowo 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.

ORIGINALITY REPORT

7%

SIMILARITY INDEX

5%

INTERNET SOURCES

2%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

teknosi.fti.unand.ac.id

Internet Source

1%

2

Submitted to Universitas Jenderal Soedirman

Student Paper

1%

3

Sahu, Anoop Kumar, Saurav Datta, and Siba Sankar Mahapatra. "Green supply chain performance benchmarking using integrated IVFN-TOPSIS methodology", International Journal of Process Management and Benchmarking, 2013.

Publication

1%

4

Mannila Sandhu, C. Danielle Vinson, Vijay K. Mago, Philippe J. Giabbanelli. "From associations to sarcasm: Mining the shift of opinions regarding the Supreme Court on twitter", Online Social Networks and Media, 2019

Publication

1%

5

jurnal.umj.ac.id

Internet Source

1%

6	id.scribd.com Internet Source	1%
7	library.binus.ac.id Internet Source	<1%
8	pt.scribd.com Internet Source	<1%
9	id.123dok.com Internet Source	<1%
10	www.scribd.com Internet Source	<1%
11	Submitted to Unika Soegijapranata Student Paper	<1%
12	Submitted to Universitas Muria Kudus Student Paper	<1%
13	clearlakehomebuyer.com Internet Source	<1%
14	Submitted to Universidad Carlos III de Madrid Student Paper	<1%
15	Submitted to Program Pascasarjana Universitas Negeri Yogyakarta Student Paper	<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography On