

BIG DATA : Elasticsearch

Ahmad Rio Adriansyah S.Si. M.Si.

Elasticsearch

- Server mesin pencari berbasiskan Apache Lucent
- Dikembangkan oleh Shay Banon dan dipublikasikan tahun 2010
- Real-time-distributed, open source (Apache licence ver. 2.0), berfungsi sebagai mesin pencari dan analitik
- Scalable hingga beberapa petabyte, data baik terstruktur ataupun tidak
- Digunakan oleh organisasi besar seperti Wikipedia, The Guardian, Stackoverflow, GitHub, dll.

Konsep

- Node
- Cluster
- Index
- Type/Mapping (deprecated pada versi 7 ke atas)
- Document
- Shard
- Replica



Keunggulan

- Dikembangkan dalam bahasa java, kompatibel dengan hampir semua platform
- Real time, data yang baru dimasukkan langsung masuk ke dalam pencariannya
- Distributed, mudah diskalakan, dan diintegrasikan
- Berkomunikasi menggunakan RESTful-API
- Menggunakan objek JSON sebagai responnya, sehingga memungkinkan servernya bekerja sama dengan berbagai bahasa pemrograman
- Mampu memproses hampir semua tipe dokumen (selain yang tidak mensupport text rendering)



Kelemahan

- Tidak mensupport banyak cara untuk menangani request dan response (hanya JSON). Apache Solr memungkinkan format CSV, XML, dan JSON, tetapi Elasticsearch lebih mudah menangani multi-tenant dibanding Apache Solr
- Split brain situation (jarang terjadi)



Elasticsearch vs RDBMS

— — —

Elasticsearch	RDBMS
Index	Database
Shard	Shard
Type/Mapping	Table
Field	Field
JSON Object	Tuple

Instalasi

- Download Java (Elasticsearch versi yang baru membutuhkan minimal Java 11)

<https://www.java.com/en/>

- Download Elasticsearch dari link berikut

versi terbaru 7.4.2 (31 Oktober 2019)

<https://www.elastic.co/downloads/elasticsearch>

- Download Postman, Fiddler, Sense (atau client webservice lain, boleh cli seperti curl), atau gunakan library elasticsearch di python

\$ pip install elasticsearch

Instalasi



Setelah diekstrak, elasticsearch sudah siap dijalankan

Jalankan menggunakan

```
$ cd elasticsearch-<version>
```

```
$ ./bin/elasticsearch
```

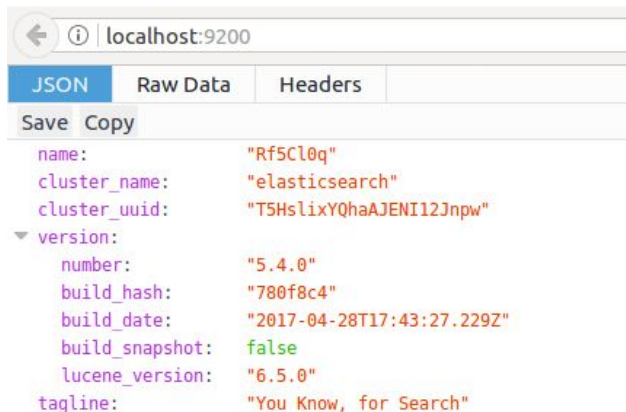
Lakukan pengujian dengan membuka localhost di port 9200 (bisa lewat browser atau curl)

```
$ curl 'http://localhost:9200/?pretty'
```



Instalasi

Jika muncul respon berikut, berarti sebuah node elasticsearch sudah berjalan dan siap digunakan



```
{
  "name": "Rf5Cl0q",
  "cluster_name": "elasticsearch",
  "cluster_uuid": "T5HslixYQhaAJENI12Jnpw",
  "version": {
    "number": "5.4.0",
    "build_hash": "780f8c4",
    "build_date": "2017-04-28T17:43:27.229Z",
    "build_snapshot": false,
    "lucene_version": "6.5.0"
  },
  "tagline": "You Know, for Search"
}
```



Change Port Configuration

Elasticsearch secara default berjalan pada port 9200

Untuk mengubahnya, edit file konfigurasi elasticsearchnya (/config/elasticsearch.yml) pada bagian http.port



Elasticsearch Client

- [Java REST Client \[7.4\]](#) — other versions
- [Java API \[7.4\]](#) — other versions
- [JavaScript API \[7.x\]](#) — other versions
- [Ruby API \[7.x\]](#) — other versions
- [Go API](#)
- [.NET API \[7.x\]](#) — other versions
- [PHP API \[7.2\]](#) — other versions
- [Perl API](#)
- [Python API](#)
- [Community Contributed Clients](#)



Elasticsearch Python Client

— — —

<http://elasticsearch-py.rtfld.org/>

Instalasi :

```
$ pip install elasticsearch
```



Data

- Data dituliskan dalam elasticsearch dengan hirarki berikut

`/<index>/<type>/<id>`

- Misalkan kita mau memasukkan dokumen dalam index twitter dengan type tweet dan id 1

```
PUT twitter/tweet/1
{
  "user" : "kimchy",
  "post_date" : "2009-11-15T14:12:12",
  "message" : "trying out Elasticsearch"
}
```



Connect ES from Python Client

```
» from elasticsearch import Elasticsearch
```

```
» es = Elasticsearch([{"host": "localhost", "port": 9200}])
```

```
» es.ping()
```

```
True
```



Add Data from Python Client

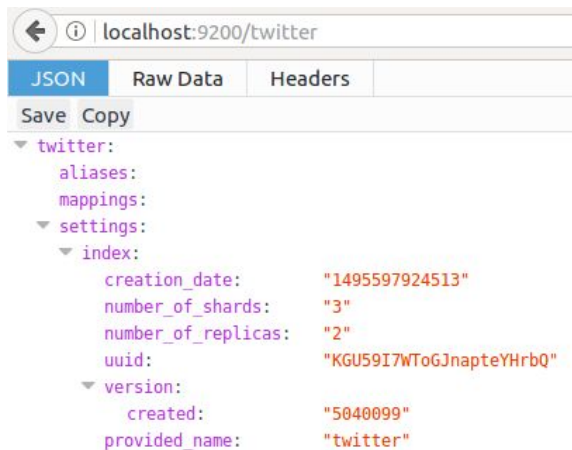
```
data = {"user": "kimchy",  
        "post_date": "2009-11-15T14:12:12",  
        "message": "trying out Elasticsearch"  
}
```

```
es.index(index="twitter",  
        doc_type="tweet",  
        id=1,  
        body=data)
```



Create Index

- Index pada elasticsearch setara dengan database yang kita gunakan pada SQL
- Index akan secara otomatis terbuat saat memasukkan data dengan fungsi `es.index()`
- Nama index harus huruf kecil semua (lowercase) atau angka
- Contoh : index bernama twitter



Read Data

- Memanggil data dapat menggunakan browser ke alamat

<http://localhost:9200/twitter/tweet/1>

atau melalui python client dengan fungsi **get**

```
➤ es.get(index="twitter",  
        doc_type="tweet",  
        id=1  
        )
```

- Datanya akan muncul pada field **_source**

Delete Data

- Menghapus dengan fungsi **delete**

```
es.delete(index="twitter",  
          doc_type="tweet",  
          id=2)
```

```
{'_index': 'twitter',  
  '_type': 'tweet',  
  '_id': '2',  
  '_version': 2,  
  'result': 'deleted',  
  '_shards': {'total': 2, 'successful': 1, 'failed': 0},  
  '_seq_no': 3,  
  '_primary_term': 1}
```

Search

```
► es.search(index="twitter")
```

```
{'took': 1202,
  'timed_out': False,
  '_shards': {'total': 1, 'successful': 1, 'skipped': 0, 'failed': 0},
  'hits': {'total': {'value': 2, 'relation': 'eq'},
    'max_score': 1.0,
    'hits': [{'_index': 'twitter',
      '_type': 'tweet',
      '_id': '1',
      '_score': 1.0,
      '_source': {'user': 'kimchy',
        'post_date': '2009-11-15T14:12:12',
        'message': 'trying out Elasticsearch'}}],
    {'_index': 'twitter',
      '_type': 'tweet',
      '_id': '3',
      '_score': 1.0,
      '_source': {'user': 'kimchy',
        'post_date': '2009-11-15T15:00:00',
        'message': 'my next post about Elasticsearch'}}]]}}
```

Mengupdate Data

- Misalnya ada data yang salah, mau diperbaiki, atau mau menambahkan data baru (misalnya tanggal lahir), kita bisa menggunakan fungsi update.
- Informasi yang diperbaiki diletakkan pada field **doc** dalam badan pesan

```
es.update(index="twitter",  
          doc_type="tweet",  
          id=3,  
          body={"doc":{"message":"edited"}})
```

- Elasticsearch secara otomatis akan meng-increment versi dari dokumen tersebut

Mengupdate Data

- Perhatikan bahwa update yang dilakukan diletakkan pada **“doc”** dalam jsonnya, tidak langsung.
- Jika kita panggil menggunakan perintah GET, versi dokumen tersebut terlihat sudah diupdate, tetapi bukan berarti dokumen versi sebelumnya tersimpan otomatis.
- Kita **tidak bisa** secara langsung mengambil versi sebelumnya dari dokumen yang sudah diupdate. (bisa diakali dari cara penyimpanan dokumennya)

Menampilkan Semua Index

Daftar semua index dapat dimunculkan dengan memanggil object `indices` pada `elasticsearch` object. Dari object tersebut, dipanggil fungsi `get_alias()` untuk semua indexnya.

Elasticsearch akan memunculkan nama index dan aliasnya. Terapkan fungsi `keys()` untuk mendapatkan daftar indexnya saja.

```
es.indices.get_alias("*").keys()
```

Menggunakan data yang lebih besar

- Download dataset di

<https://www.elastic.co/guide/en/kibana/7.1/tutorial-load-dataset.html>

- Ada 3 dataset yang tersedia, download **accounts.zip** untuk digunakan sebagai contoh. **Shakespeare** dan **logs** dapat digunakan untuk latihan dan eksplorasi



Import dari JSON

Pada folder Dataset telah tersedia file `accounts.json`

Masukkan menggunakan `curl`

```
$ curl -H 'Content-Type:application/x-ndjson' -XPOST  
'http://localhost:9200/bank/account/_bulk' --data-binary  
@accounts.json
```



Mencari Data

- Selama ini kita memanggil data menggunakan id dari dokumennya (link lengkap). Kita bisa mencari data yang mengandung kata tertentu dari seluruh dokumen yang ada di **index** atau **type** tertentu dengan menggunakan fungsi **_search**
- Formatnya **_search?q=katayangdicari**

```
$ curl -XGET 'http://localhost:9200/bank/account/_search?q=john'
```



Mencari Data

— — —

- Pencarian dengan `_search?q=katayangdicari` mengembalikan semua nilai pada semua key.
- Jika ingin mencari kata pada key tertentu, maka kita gunakan filter tambahan
- Formatnya `_search?q=keyfilter:value`

```
$ curl -XGET 'http://localhost:9200/bank/account/_search?q=nama:ahmad'
```



Mencari Data dengan Python Client

```
▶ es.search(index="bank",  
            body={  
                "query":{  
                    "match":{"firstname":"Johnson"}  
                }  
            })
```



Query DSL

<https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html>

Query Domain Specific Language

Digunakan untuk mendefinisikan query pada elasticsearch.

Ada 2 tipe klausa :

- leaf query : untuk mencari nilai tertentu di field tertentu (contoh : match, term, range)
- compound query : mengkombinasikan beberapa query secara logical