Kegiatan Lab Mandiri

Pendalaman Database 1

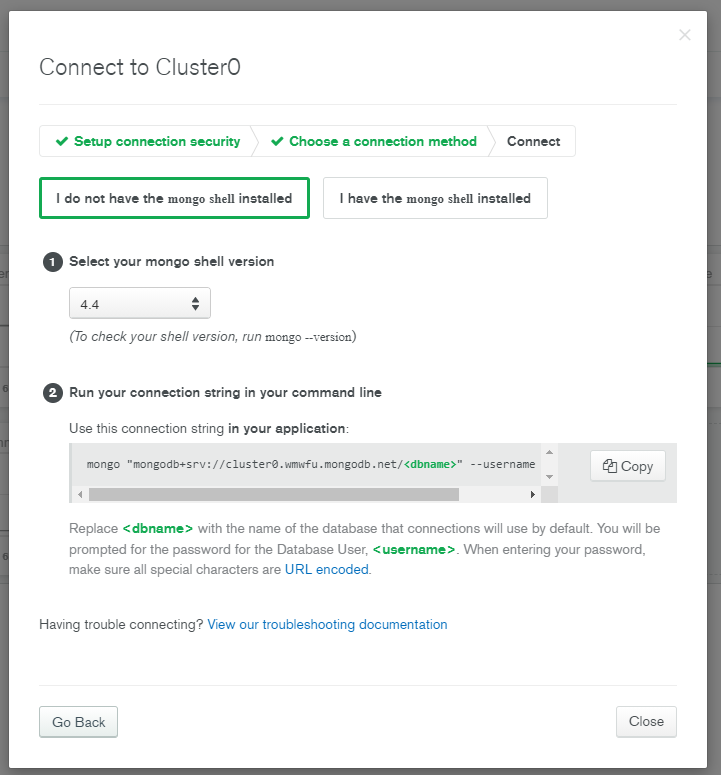
Prerequisite:

* Memiliki akun pada mongodb Atlas, <https://cloud.mongodb.com/>
* Melakukan instalasi MongoDB Shell

# Bagian 1: MongoDB

## Penggunaan index untuk optimalisasi query

1. Melakukan koneksi ke instance database, masuk ke akun atlas dan pilih “connect”, menggunakan MongoShell. Copy connection command, ganti <username> dan <password> sesuai dengan user pada masing-masing instance yang sudah dibuat.



1. Masuk ke database `simpleBank`

|  |
| --- |
| >MongoDB Enterprise Cluster0-shard-0:PRIMARY> show dbs  simpleBank 0.000GB  >MongoDB Enterprise Cluster0-shard-0:PRIMARY> use simpleBank  switched to db simpleBank |

1. Insert sample data:

|  |
| --- |
| db.customers.insertMany([  { first\_name: "Savannah", last\_name: "Bishop", age: 28, customer\_type: "employee", street: "8859 Hamilton Ave", city: "Cambridge", state: "Pennsylvania", zip\_code: "45417",phone\_number: "(676)-847-4094" },  { first\_name: "Lesa", last\_name: "Graham", age: 57, customer\_type: "employee", street: "9179 Oak Lawn Ave", city: "Spokane", state: "Indiana", zip\_code: "65626",phone\_number: "(183)-068-1703" },  { first\_name: "Rick", last\_name: "Harper", age: 22, customer\_type: "student", street: "8732 Robinson Rd", city: "Orange", state: "Pennsylvania", zip\_code: "31168",phone\_number: "(377)-762-7780" },  { first\_name: "Ian", last\_name: "Bennett", age: 71, customer\_type: "employee", street: "2199 Preston Rd", city: "Columbia", state: "Arkansas", zip\_code: "97400",phone\_number: "(050)-371-8203" },  { first\_name: "Mae", last\_name: "Fernandez", age: 25, customer\_type: "student", street: "335 Rolling Green Rd", city: "Wichita Falls", state: "Kansas", zip\_code: "95191", phone\_number: "(339)-055-3014" },  { first\_name: "Gerald", last\_name: "Ray", age: 36, customer\_type: "employee", street: "1738 Blossom Hill Rd", city: "Addiss", state: "Oregon", zip\_code: "61178",phone\_number: "(482)-912-3722" }  ]) |

1. Melakukan query sederhana dan menggunakan explain

|  |
| --- |
| MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.customers.find({age: 36, customer\_type: "employee"}).pretty()  {  "\_id" : ObjectId("5f6e8ac091167bdd0b933030"),  "first\_name" : "Gerald",  "last\_name" : "Ray",  "age" : 36,  "customer\_type" : "employee",  "street" : "1738 Blossom Hill Rd",  "city" : "Addiss",  "state" : "Oregon",  "zip\_code" : "61178",  "phone\_number" : "(482)-912-3722"  } |

Menggunakan explain untuk mengetahui operasi query.

|  |
| --- |
| MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.customers.find({age: 36, customer\_type: "employee"}).explain("executionStats")  {  …,  "executionStats" : {  "executionSuccess" : true,  "nReturned" : 1,  "executionTimeMillis" : 0,  "totalKeysExamined" : 0,  "totalDocsExamined" : 6,  "executionStages" : {  "stage" : "COLLSCAN",  "filter" : {  "$and" : [  {  "age" : {  "$eq" : 36  }  },  {  "customer\_type" : {  "$eq" : "employee"  }  }  ]  },  "nReturned" : 1,  "executionTimeMillisEstimate" : 0,  "works" : 8,  "advanced" : 1,  "needTime" : 6,  "needYield" : 0,  "saveState" : 0,  "restoreState" : 0,  "isEOF" : 1,  "direction" : "forward",  "docsExamined" : 6  }  },  ...  } |

Hasil yang perlu diperhatikan yaitu pada key totalDocsExamined. Dari hasil tersebut menunjukkan, untuk menemukan dokumen yang dicari, sistem perlu melakukan pengecekan pada setiap data yang ada. Hal ini sangat tidak efisien, khususnya jika ukuran data pada collection tersebut besar.

1. Menerapkan basic index

|  |
| --- |
| MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.customers.createIndex({ age: 1, customer\_type: 1 }) |

1. Melakukan query hasil index

|  |
| --- |
| MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.customers.find({age: 36, customer\_type: "employee"}).explain("executionStats")  ...,  "executionStats" : {  "executionSuccess" : true,  "nReturned" : 1,  "executionTimeMillis" : 0,  "totalKeysExamined" : 1,  "totalDocsExamined" : 1,  "executionStages" : {  "stage" : "FETCH",  "nReturned" : 1,  "executionTimeMillisEstimate" : 0,  "works" : 2,  "advanced" : 1,  "needTime" : 0,  "needYield" : 0,  "saveState" : 0,  "restoreState" : 0,  "isEOF" : 1,  "docsExamined" : 1,  "alreadyHasObj" : 0,  "inputStage" : {  "stage" : "IXSCAN",  "nReturned" : 1,  "executionTimeMillisEstimate" : 0,  "works" : 2,  "advanced" : 1,  "needTime" : 0,  "needYield" : 0,  "saveState" : 0,  "restoreState" : 0,  "isEOF" : 1,  "keyPattern" : {  "age" : 1,  "customer\_type" : 1  },  "indexName" : "age\_1\_customer\_type\_1",  "isMultiKey" : false,  "multiKeyPaths" : {  "age" : [ ],  "customer\_type" : [ ]  },  "isUnique" : false,  "isSparse" : false,  "isPartial" : false,  "indexVersion" : 2,  "direction" : "forward",  "indexBounds" : {  "age" : [  "[36.0, 36.0]"  ],  "customer\_type" : [  "[\"employee\", \"employee\"]"  ]  },  "keysExamined" : 1,  "seeks" : 1,  "dupsTested" : 0,  "dupsDropped" : 0  }  }  },  ... |

Dari hasil tersebut totalDocsExamined menunjukkan optimasi yang sangat signifikan.

## Agregasi dasar pada MongoDB

1. Menggunakan agregasi addFields untuk mendapatkan nama lengkap dari data pengguna.

|  |
| --- |
| MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.customers.aggregate([  ... {  ... $addFields: {  ... full\_name: { $concat: ["$first\_name", " ", "$last\_name" ]}  ... }  ... }  ... ]).pretty() |

Query tersebut akan menambahkan field full\_name seperti contoh salah satu hasil dokumen berikut:

|  |
| --- |
| {  "\_id" : ObjectId("5f6e8ac091167bdd0b93302b"),  "first\_name" : "Savannah",  "last\_name" : "Bishop",  "age" : 28,  "customer\_type" : "employee",  "street" : "8859 Hamilton Ave",  "city" : "Cambridge",  "state" : "Pennsylvania",  "zip\_code" : "45417",  "phone\_number" : "(676)-847-4094",  "full\_name" : "Savannah Bishop"  } |

1. Mengambil sampel data menggunakan $sample

|  |
| --- |
| MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.customers.aggregate([  ... {  ... $sample: { size: 2 }  ... }  ... ]).pretty() |

Operasi tersebut akan mengambil sampel data dari collection yang dimaksud secara random, dengan contoh hasil sebagai berikut:

|  |
| --- |
| {  "\_id" : ObjectId("5f6e8ac091167bdd0b93302e"),  "first\_name" : "Ian",  "last\_name" : "Bennett",  "age" : 71,  "customer\_type" : "employee",  "street" : "2199 Preston Rd",  "city" : "Columbia",  "state" : "Arkansas",  "zip\_code" : "97400",  "phone\_number" : "(050)-371-8203"  }  {  "\_id" : ObjectId("5f6e8ac091167bdd0b93302f"),  "first\_name" : "Mae",  "last\_name" : "Fernandez",  "age" : 25,  "customer\_type" : "employee",  "street" : "335 Rolling Green Rd",  "city" : "Wichita Falls",  "state" : "Kansas",  "zip\_code" : "95191",  "phone\_number" : "(339)-055-3014"  } |

## Core Agregation pada MongoDB

1. Menggunakan $group untuk melihat customer\_type pada collection customers

|  |
| --- |
| MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.customers.aggregate([  ... {  ... $group: { \_id: "$customer\_type" }  ... }  ... ]).pretty() |

Dari operasi tersebut kita akan mengetahui value unik apa saja yang terdapat pada field customer\_type dengan hasil sebagai berikut:

|  |
| --- |
| { "\_id" : "student" }  { "\_id" : "employee" } |

Untuk mendapatkan informasi lain misalnya banyak dokumen pada masing-masing value dapat ditambahkan seperti contoh berikut:

|  |
| --- |
| MongoDB Enterprise Cluster0-shard-0:PRIMARY> db.customers.aggregate([  ... {  ... $group: {  ... \_id: "$customer\_type",  ... count: { $sum: 1 }  ... }  ... }  ... ]) |

Operasi tersebut akan memberikan hasil sebagai berikut:

|  |
| --- |
| { "\_id" : "student", "count" : 2 }  { "\_id" : "employee", "count" : 4 } |

## Performa Agregasi

1. Menambahkan parameter explain pada agregasi untuk menampilkan performa.

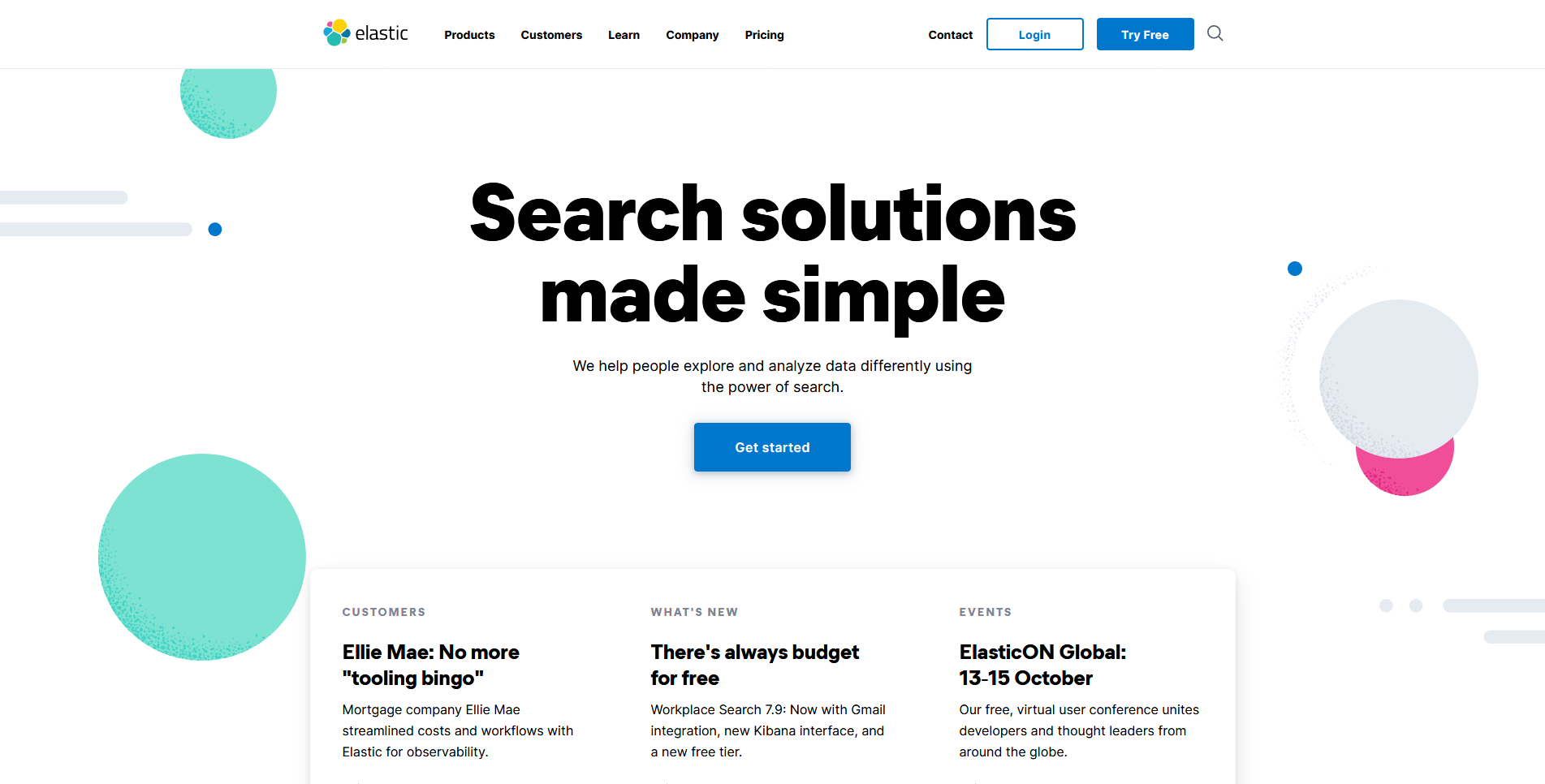
|  |
| --- |
| db.customers.aggregate([  {  $group: {  \_id: "$customer\_type",  count: { $sum: 1 }  }  }  ], { explain: true }) |

# Bagian 2: Elasticsearch

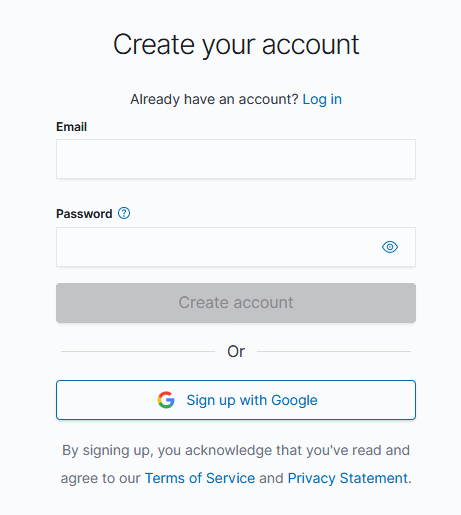
## Setup Elastic pada Cloud

Pada bagian ini akan menggunakan service cloud elasticsearch pada <https://www.elastic.co/> menggunakan akun trial.

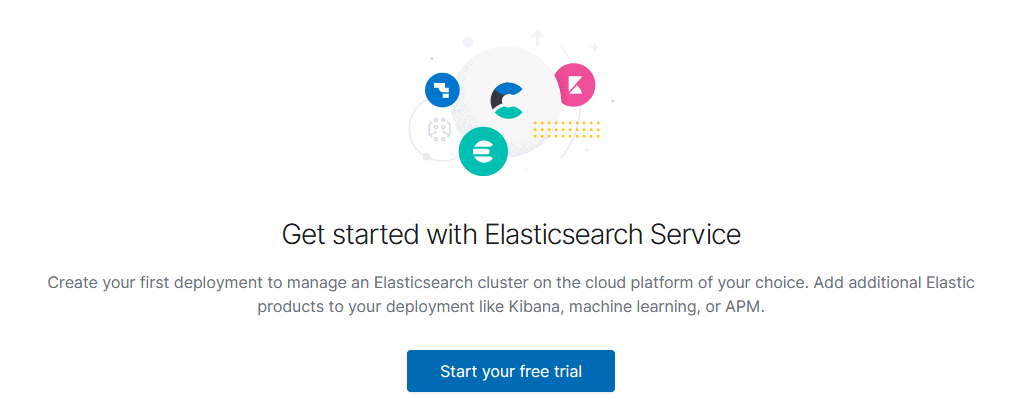
1. Buka https://www.elastic.co/ dan pilih “Get Started”



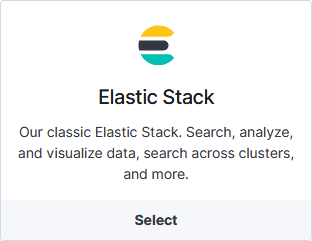
1. Lakukan registerasi menggunakan email atau menggunakan akun google



1. Lakukan verifikasi email
2. Pilih “Start Your Free Trial”



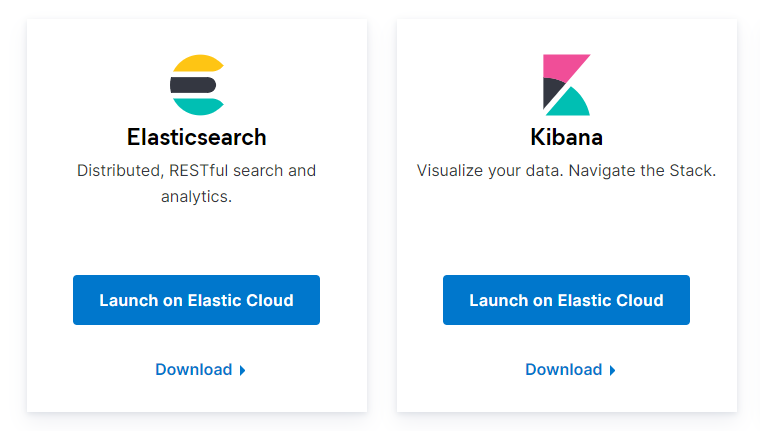
1. Pilih “Elastic Stack” dan “Create Deployment”



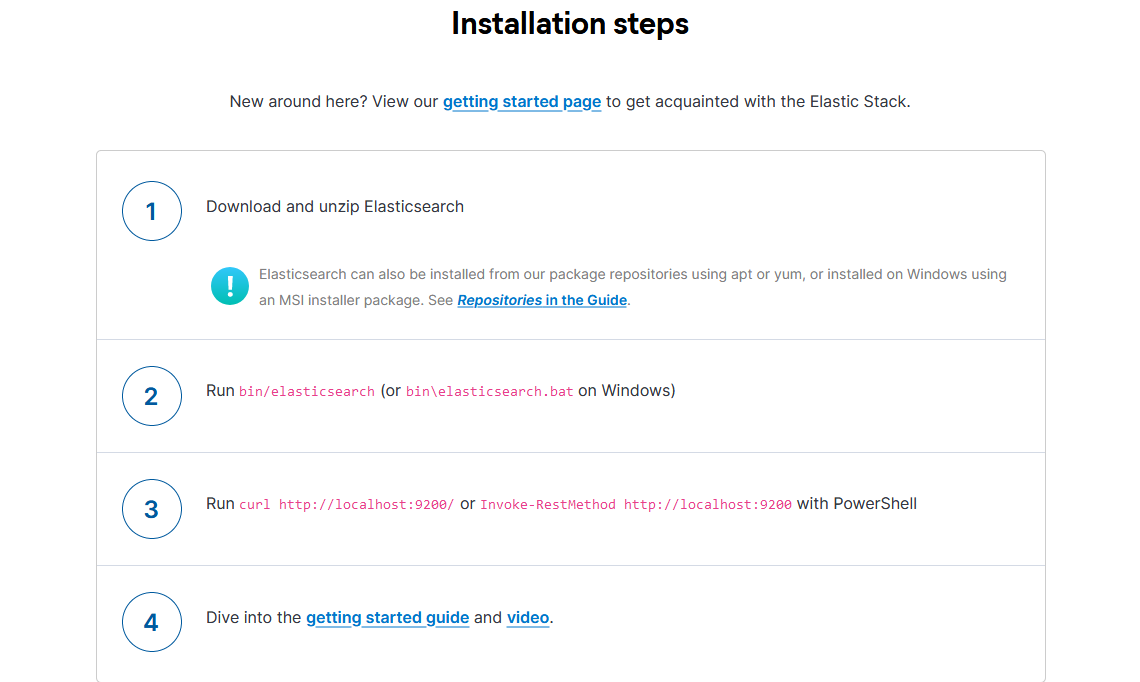
1. Simpan Username dan Password

## Setup Elastic di Local

1. Buka hamalan <https://www.elastic.co/downloads/> dan download pilih download elastic dan kibana

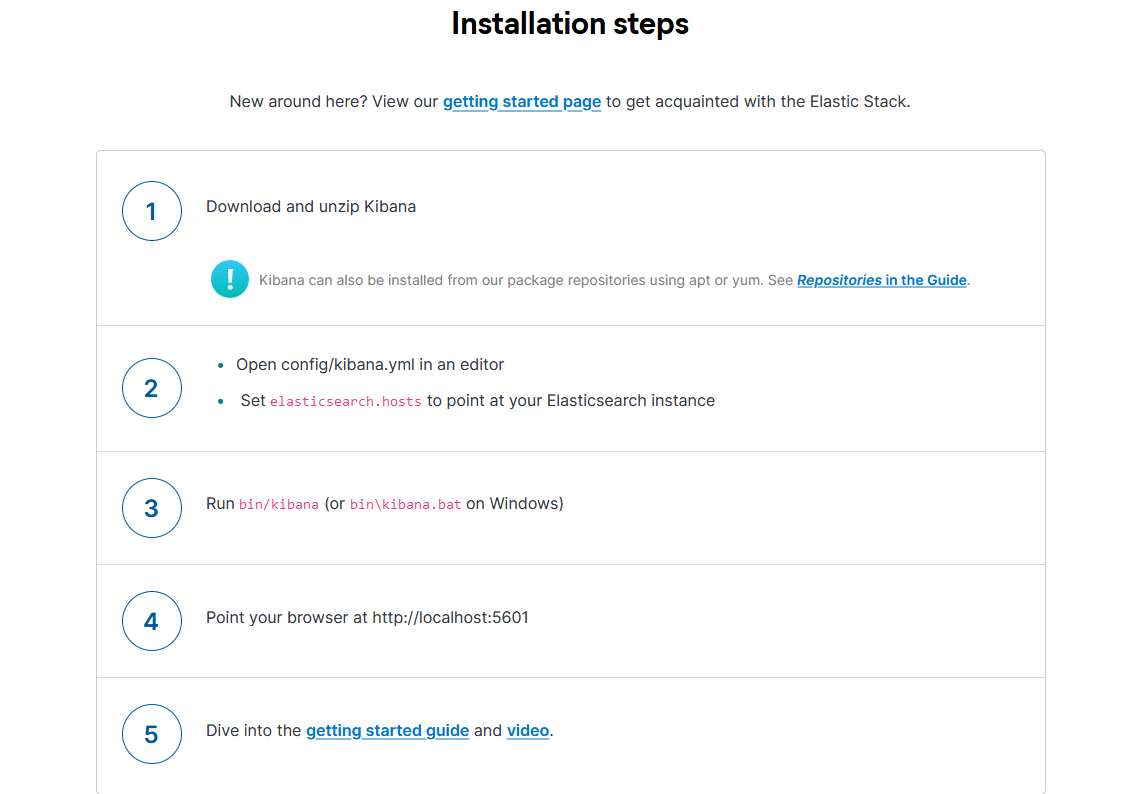


1. Pilih download yang sesuai untuk OS masing-masing, untuk elasticsearch Windows disarankan memilih MSI installer.
2. Untuk windows, jalankan file .msi hasil download dan selesaikan proses instalasi.
3. Installation steps elastic (muncul pada bagian bawah page download)



Buka <http://localhost:9200/> pada browser untuk memastikan elasticsearch berhasil berjalan.

1. Installation steps kibana (muncul pada bagian bawah page download)



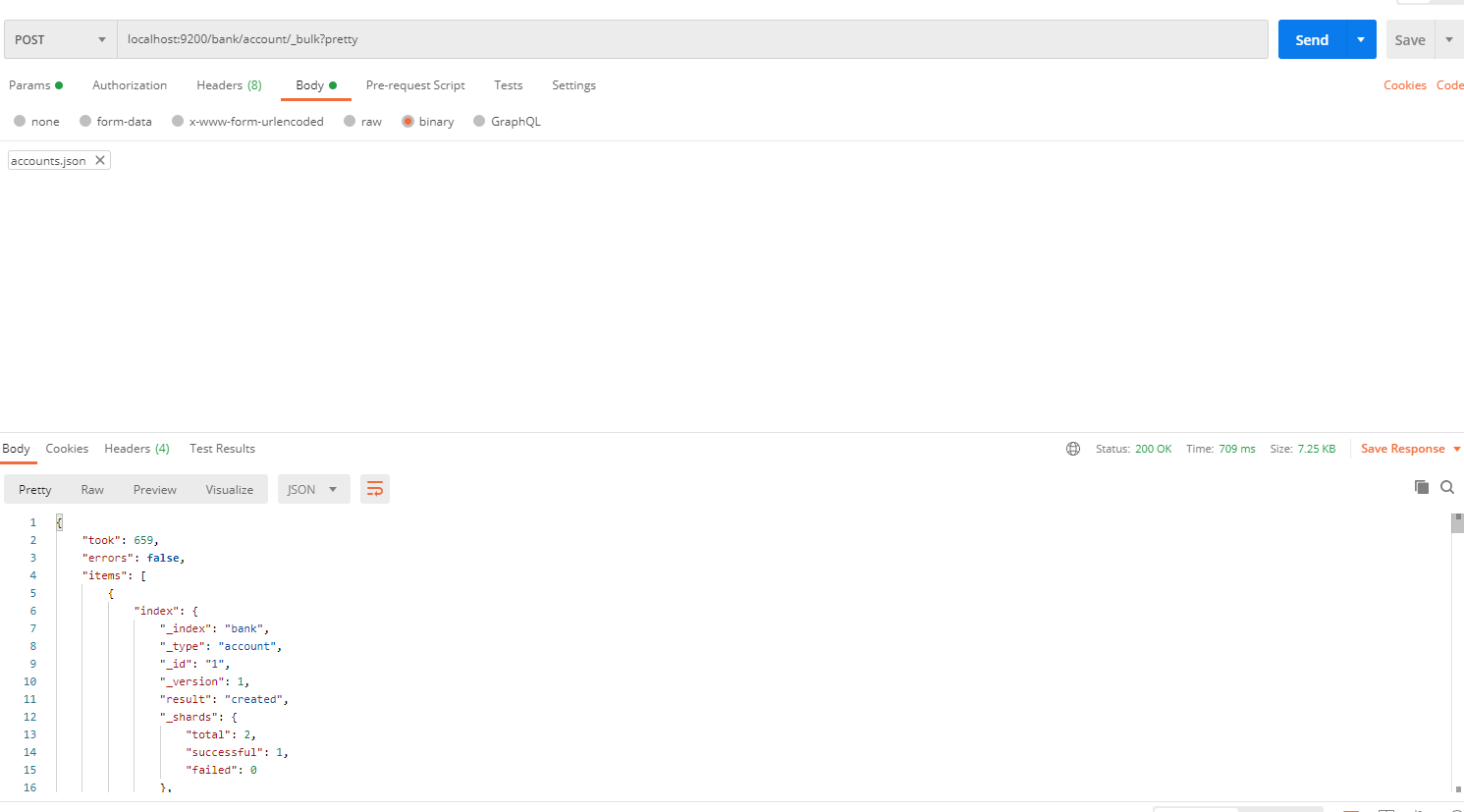
Untuk membuka bat file pada windows perlu menggunakan akses Administrator.

## Membuat index baru pada Elasticsearch

1. Index baru akan otomatis dibuat ketika kita melakukan load bulk data melalui POST request ke kibana.
2. Pastikan elasticsearch maupun kibana berjalan
3. Download data sample pada link <https://drive.google.com/file/d/1SdTsTwqcuK19_8xXDfiCchLkqz6i_lU3/view?usp=sharing>
4. Jika menggunakan terminal, CD ke directory hasil download file
5. Lakukan request POST ke kibana menggunakan command curl ataupun postman seperti berikut:

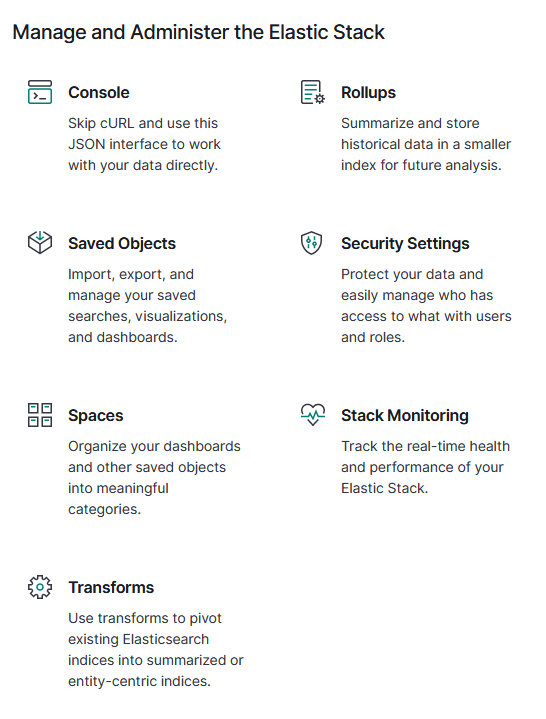
|  |
| --- |
| curl -H 'Content-Type: application/x-ndjson' -XPOST 'localhost:9200/bank/account/\_bulk?pretty' --data-binary @accounts.json |

Postman:



Index bank akan dibuat secara otomatis.

1. Kita akan melihat index yang telah dibuat pada kibana dengan membuka <http://localhost:5601/app/home#/>
2. Buka “console”



1. Masukkan command berikut

|  |
| --- |
| GET \_cat/indices |

Jalankan dengan menekan tombol play pada ujung kanan. Akan tampil index yang ada, termasuk index yang baru dibuat.

|  |
| --- |
| green open .kibana-event-log-7.9.2-000001 UesNOS\_USsqYJ7oM44zCXA 1 0 1 0 5.5kb 5.5kb  yellow open bank Ex1ie\_xQTrGh2p0GKp7JWg 1 1 1000 0 382.2kb 382.2kb  green open .apm-custom-link kBc7D9RuT2SvDEDFyJqZ5g 1 0 0 0 208b 208b  green open .kibana\_task\_manager\_1 PlKhAnX2StCweAcPd5x2WQ 1 0 6 42 106.7kb 106.7kb  green open .apm-agent-configuration NHHr8yiATcWfmXvrVeB5lQ 1 0 0 0 208b 208b  green open .kibana\_1 df56ZITITVCRuDREL07RPg 1 0 24 9 10.4mb 10.4mb |