

Data Benchmark pada Google BigQuery dan Elasticsearch

(Data Benchmark for Google BigQuery and Elasticsearch)

Nisrina Akbar Rizky Putri¹, Widyawan², Teguh Bharata Adji³

Abstract—Nowadays, the cloud is not only a data storage medium but can be used as a medium for managing or analyzing data. Google offers Google BigQuery as a platform capable of managing and analyzing data, while Elasticsearch itself is a search and analysis engine that can be used to analyze data using Kibana. Using a dataset in the form of tweets crawled through <http://netlytic.org/>, containing the hashtags #COVID19 and #coronavirus, the data will be analyzed and used to compare its performance with benchmarks. Benchmark is a process used to measure and compare performance against an activity so that the desired level of performance is achieved. Data benchmark is performed on both platforms to generate or determine the workload of the platforms. The result obtained in this study is that Google BigQuery has superior results, both from the upload container for larger datasets than Elasticsearch and with two query testing models. The query management time on Google BigQuery is also shorter and faster than Elasticsearch. Meanwhile, the visualization results from these two platforms have the same percentage amount.

Intisari—Cloud di masa kini tidak hanya berfungsi sebagai media penyimpanan data, tetapi dapat digunakan juga sebagai media untuk mengelola ataupun menganalisis suatu data. Google menawarkan Google BigQuery sebagai platform yang mampu mengelola dan menganalisis data, sedangkan Elasticsearch merupakan pencari dan analisis yang dapat digunakan untuk menganalisis data dengan menggunakan Kibana. Dengan menggunakan dataset berupa cuitan hasil proses crawling melalui <http://netlytic.org/> yang mengandung tagar #COVID19 dan #coronavirus, data tersebut dianalisis dan digunakan untuk membandingkan kinerjanya dengan benchmark. Benchmark merupakan proses yang digunakan untuk mengukur dan membandingkan kinerja terhadap sebuah aktivitas, sehingga tercapai tingkat kinerja yang diinginkan. Data benchmark dilakukan pada kedua platform untuk menghasilkan atau mengetahui beban kerja dari platform. Hasil akhir yang didapatkan menunjukkan bahwa Google BigQuery memiliki hasil yang lebih unggul, baik dari wadah upload untuk dataset yang lebih besar dibandingkan Elasticsearch dan dengan dua model pengujian kueri. Waktu pengelolaan kueri pada Google BigQuery juga lebih singkat dan cepat dibandingkan dengan Elasticsearch. Selain itu, hasil visualisasi dari kedua platform ini memiliki jumlah persentase yang sama.

Kata Kunci—Google BigQuery, Elasticsearch, Benchmarking, Tweets.

^{1,2,3} Departemen Teknik Elektro dan Teknologi Informasi, Universitas Gadjah Mada, Jl. Grafika No.2, Yogyakarta 55281 Indonesia (telp: 0274-552-301; fax: 0274-547-506; e-mail: ¹nisrinaakbar95@mail.ugm.ac.id, ²widyawan@ugm.ac.id, ³adji@ugm.ac.id)

I. PENDAHULUAN

Dengan segala perkembangan teknologi yang semakin pesat, internet kini menjadi salah satu sumber dari banyaknya data dan informasi. Data merupakan berkas digital yang dihasilkan baik oleh komputer (*raw*), acak, belum terorganisasi, dan belum memiliki nilai, fungsi maupun arti [1].

Setiap harinya terbuat 2,5 Qi-byte data, sehingga 90% data di dunia saat ini telah dibuat hanya dalam dua tahun terakhir [2]. Jumlah ini terus bertambah dikarenakan adanya kebiasaan sosial yang baru, transformasi sosial, serta peningkatan sistem perangkat lunak yang makin luas [3]. Salah satu penyumbang data dengan jumlah cukup banyak adalah media sosial, seperti Twitter dengan cuitannya yang menjadi sumber data. Twitter dipilih karena menjadi salah satu media sosial yang populer dengan lebih dari 353 juta pengguna aktif dalam setiap bulannya [4]. Cuitan ini nantinya dapat disimpan sebagai salah satu media untuk menyampaikan informasi.

Penyimpanan cuitan tidak hanya dilakukan secara lokal, tetapi dapat juga menggunakan *cloud*. Pengguna dapat dengan mudah menggunakan data tersebut di mana pun dan membayar layanan sesuai dengan yang digunakan [5]. Selain media penyimpanan, *cloud* dapat digunakan sebagai media untuk mengelola ataupun menganalisis suatu data. Google menawarkan platform yang mampu mengelola dan menganalisis suatu data mentah (*raw*), yaitu Google BigQuery, sedangkan Elasticsearch merupakan mesin pencari dan analisis yang bersifat *open source* sehingga dapat digunakan untuk menganalisis data dengan menggunakan Kibana.

BigQuery adalah sebuah *enterprise data warehouse* yang dapat menyelesaikan suatu masalah dengan menyanggupi *super-fast SQL queries* menggunakan kekuatan pemrosesan dari infrastruktur Google [6]. BigQuery digambarkan sebagai alat dari Google Developers yang memungkinkan dijalankannya kueri dengan cepat dari kumpulan data dalam jumlah besar. Untuk merespons suatu tuntutan pada *cloud* berdasarkan *big data* dan *analytical*, BigQuery menyediakan kemampuan untuk memproses kueri yang besar dari *dataset*. Google BigQuery digambarkan seperti menyediakan pengelolaan sepenuhnya, *NoOps*, dan murah biaya dalam menganalisis [7]. BigQuery dapat melakukan pengelolaan data, seperti membuat atau menghapus tabel berdasarkan skema dengan kode JSON, serta mengimpor data dengan jenis CSV atau JSON. BigQuery juga dapat digunakan dari bahasa apa pun yang dapat bekerja dengan REST API atau sesuai dengan yang dibutuhkan. BigQuery memberikan kredit sebesar \$300 dan dalam waktu satu tahun untuk mengeksplor.

Elasticsearch merupakan *tool* yang dapat menangani analisis media sosial atau sebagai media penyimpanan dan mesin pencari serta analisis yang mampu mengelola data secara *real-*

time dan bersifat *scalable*, yang berarti dapat melakukan penyesuaian tergantung pada kebutuhan pengguna dan sistemnya [8]. Elasticsearch memiliki kueri dengan level rendah yang berbasis *RESTful*. Konsep yang dimiliki adalah dapat mengasumsikan *index* sebagai *database* dengan tipe dan dokumen menjadi *record* atau *row*, sedangkan *mapping* diasumsikan seperti table skema. Elastic.co dapat mengimpor data dengan jenis CSV ataupun JSON dan biasanya Elasticsearch akan langsung melakukan konversi teks menjadi suatu istilah yang digunakan untuk menyaring karakter-karakter yang tidak diperlukan, sehingga data secara langsung disaring. Dengan menggunakan Kibana sebagai alat visualisasi, Elasticsearch dapat diandalkan sebagai alat yang mampu menganalisis suatu data. Baik Elasticsearch diinstal secara lokal ataupun dengan masa empat belas hari percobaan dari Elastic.co, disediakan beberapa fitur yang dapat dicoba dalam menganalisis dan memvisualisasi data.

Kedua *platform* di atas mampu menganalisis dan memvisualisasikan data cuitan. Dengan melakukan suatu perbandingan kinerja di antara Google BigQuery dan Elasticsearch dengan menggunakan data cuitan tersebut, dilakukan *benchmark* pada kedua *platform*. *Benchmark* merupakan suatu proses yang digunakan untuk mengukur dan membandingkan kinerja terhadap sebuah aktivitas, sehingga suatu organisasi atau perusahaan mencapai tingkat kinerja yang diinginkan. *Benchmark* sebenarnya dapat diklasifikasikan berdasarkan hal yang sedang atau akan diklasifikasikan. Secara konseptual, *benchmark* bertujuan untuk menghasilkan beban kerja suatu aplikasi dan uji kemampuan dari suatu proses *dataset* untuk menampilkan atau menunjukkan hasil evaluasi [9]. *Benchmark* sendiri dapat diklasifikasikan berdasarkan hal yang sedang atau akan diklasifikasikan. Pengelompokan ini dibagi menjadi *benchmark* pada perangkat keras atau perangkat lunak. Pada makalah ini disinggung mengenai *benchmark* pada perangkat lunak, yang di dalamnya terdapat subkategori, yaitu *application software*, yang membandingkan suatu kinerja dan fungsi spesifik suatu aplikasi; sistem perangkat lunak, yaitu membandingkan kinerja dan fungsi dari perangkat lunak; dan *service applications* dan *micro-benchmark*, yang mengukur kinerja paling kecil dan spesifik [10].

Berdasarkan latar belakang tersebut, dengan melakukan *crawling* data pada Twitter yang didapatkan dari <http://netlytic.org/>, yang memiliki jumlah sekitar 5.000 cuitan mengandung tagar #COVID19 dan #coronavirus, makalah ini membandingkan kinerja *platform* Google BigQuery dan Elasticsearch. *Benchmarking* data dilakukan untuk menghasilkan atau mengetahui beban kerja suatu *platform* dan uji kemampuan dalam memproses suatu *dataset* untuk menampilkan atau menunjukkan hasil evaluasi yang memiliki guna [9].

Beberapa penelitian yang berkaitan dengan topik mengenai *benchmark* telah dilakukan. Salah satunya adalah pengujian pada *dataset* yang berisikan status Twitter dalam jumlah 100.000 dan 3.000.000 baris status dengan menggunakan Hadoop 2, Spark, dan Presto [11]. Pengujian dilakukan dengan suatu proses dari kueri dari model yang sudah ditentukan untuk mengetahui waktu responsnya. Pengujian dilakukan dalam

lima aktivitas berbeda, yaitu pengujian dengan kueri tingkat rendah, yaitu “*select*”; pengujian dengan kueri tingkat menengah, yaitu “*group*”; lalu pengujian dengan kueri tingkat tinggi, yaitu “*join*”. Pada pengujian keempat dan kelima digunakan kueri tingkat rendah, tetapi dilakukan peningkatan sebanyak dua dan empat kali lipat pada *core processor*-nya.

Penelitian lainnya melakukan evaluasi dan perbandingan kinerja pada Twitter River dan Logstash. Data yang digunakan dalam penelitian ini merupakan data cuitan yang didapatkan dengan *crawling* dengan jumlah berkisar 60.000 cuitan. Parameter yang diamati adalah beban proses CPU, penggunaan RAM, penggunaan *disk*, jumlah masukan data Twitter ke Elasticsearch, dan jumlah *input field* Twitter ke Elasticsearch [12].

Telah dilakukan juga penelitian yang membandingkan kinerja pada basis data NoSQL, yaitu Elasticsearch dan CouchDB, dengan menggunakan sampel *dataset* yang terdiri atas 20.000 dokumen [13]. Pengujian dilakukan dengan cara *insertion*, *selection*, *updatation*, dan *deletion* pada data sampelnya. Hasil yang didapatkan setelah dilakukan observasi adalah Elasticsearch membutuhkan lebih banyak waktu dibandingkan CouchDB pada perintah *insert*, *update*, dan *delete* data. Akan tetapi, dalam melakukan *selection* data, kinerja dari Elasticsearch jauh lebih efisien.

Penelitian lain mengenai *benchmark* dilakukan pada data mengenai meteorologi dan oseanografi (MODData) dengan cara mengetahui waktu respons suatu indeks dan nonindeks dengan menggunakan tiga model kueri yang diujikan dengan menggunakan Hive. Tipe kueri yang diujikan pertama adalah “SELECT WHERE” serta tipe kedua “SELECT JOIN” dan “SELECT GROUP BY” [14].

Penelitian selanjutnya melakukan perbandingan pada Amazon Web Service dan Microsoft Azure Cloud Platform untuk *High Performance Computing* (HPS) [15]. Kedua *platform* tersebut dibandingkan untuk mengetahui kinerja pada kedua HPC *benchmark*, yaitu *HPC Challenge* (HPPC) dan *High-Performance Conjugate Gradient* (HPCG). Kedua *benchmark* merupakan tes dari beberapa aspek dari sistem kinerja pada komputer, termasuk pada kecepatan komputasi, *memory bandwidth*, dan *network bandwidth*.

II. METODOLOGI

A. *Crawling* Data

Dalam makalah ini, data didapatkan dari <http://netlytic.org/>. *Website* tersebut mampu melakukan *crawling* data pada beberapa media sosial dan juga dapat melakukan pengelolaan menggunakan *dataset* pribadi. Pada makalah ini dibutuhkan data berupa cuitan dari Twitter. *Crawling* dilakukan dengan menggunakan akun pribadi yang akan mencari berdasarkan *keyword* sesuai dengan kebutuhan, lalu dapat dilakukan filter berdasarkan bahasa atau sumber dan target dari cuitan dengan batas maksimal 2.500 cuitan yang dapat dikenai proses *crawling*.

Data yang didapatkan pada makalah ini merupakan data yang dicuitkan pada tanggal 23 Maret 2021, dengan dilakukan *crawling* sebanyak dua kali; dalam satu kali *crawling* didapatkan 2.500 cuitan. Data didapatkan dengan pencarian

berupa dua tagar yang berbeda, yaitu #COVID19 dan #coronavirus. Data cuitan tersebut didapatkan dalam bentuk dua buah *file* CSV. Untuk kebutuhan dalam makalah ini, kedua *file* CSV digabungkan (*merged*) menjadi satu *file* CSV dengan total data menjadi 5.000 cuitan.

B. Merged Dataset

Kedua *dataset* yang didapatkan melalui <http://netlytic.org/> disimpan di Google Drive. Kemudian, dengan Google Colab, kedua *file* tersebut digabungkan. Sebelumnya, dilakukan beberapa impor modul pada Google Colab, salah satunya digunakan untuk memanggil data CSV yang disimpan pada Google Drive yang akan diinisialisasikan menggunakan *path*. Kemudian, dilakukan langkah dalam penggabungan *dataset*. Kedua *file* CSV yang memiliki jenis tagar berbeda digabungkan menjadi satu *file* CSV sehingga memudahkan pengelolaan data menggunakan Google BigQuery dan Elasticsearch. Berikut ini merupakan *syntax* yang digunakan dalam penggabungan *dataset* menggunakan Google Colab.

```
all_files = glob.glob(os.path.join(path, "CoronavirusTweets_*.csv"))

all_df = []
for f in all_file:
    df = pd.read_csv(f, sep=',')
    all_df.append(df)

merged_df = pd.concat(all_df, ignore_index=True, sort=True)

merged_df.to_csv("TD_Merged.csv")
```

C. Pengujian Model Pertama

Setelah data digabungkan menggunakan Google Colab, selanjutnya data diujikan menggunakan Google BigQuery dan Elasticsearch. Pada pengujian model pertama, dengan data sebanyak 5.000 cuitan, dilakukan *selecting* data dari masing-masing *platform*. Dari kedua *platform*, terdapat perbedaan dalam cara memasukkan perintah kueri. Berikut ini ditunjukkan kueri pengujian pada Google BigQuery,

```
SELECT * FROM 'benchmarkdata.tweets_merged.Tweets_corona'
```

sedangkan di bawah ini merupakan bagian kueri pengujian menggunakan Elasticsearch.

```
{
  "range": {
    "@timestamp": {
      "gte": "2021-03-23T05:46:19.000Z",
      "lte": "2021-03-23T13:53:04.000Z",
      "format": "strict_date_optional_time"
    }
  }
}
```

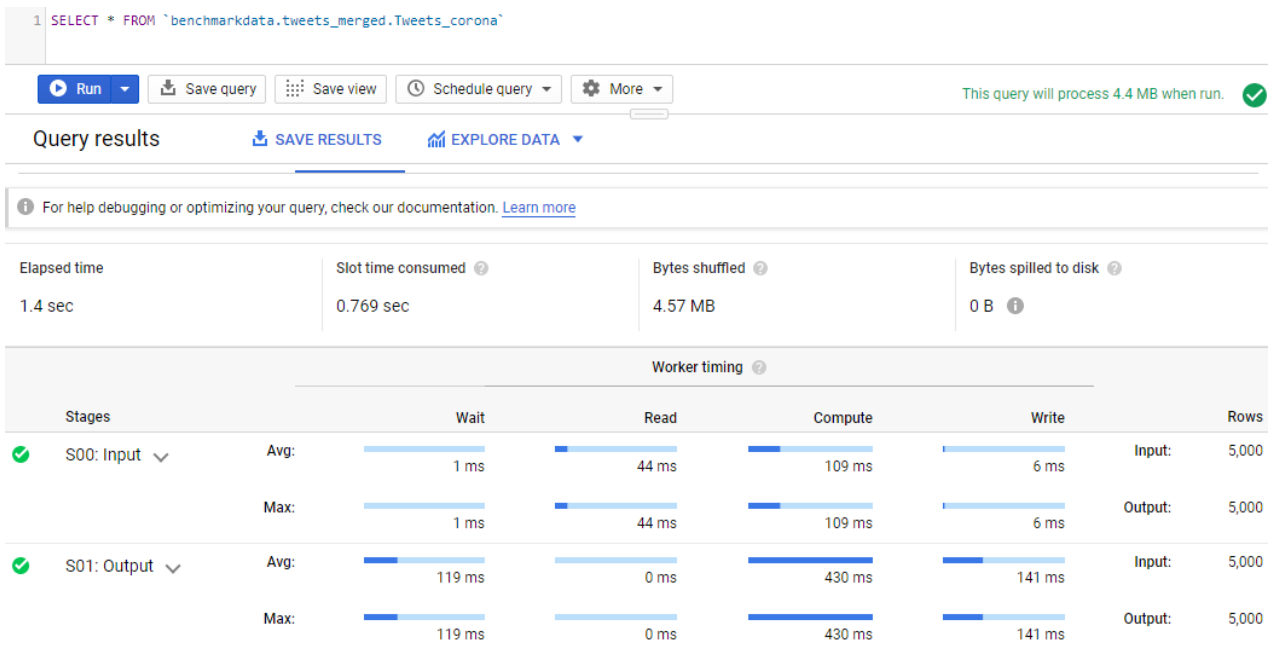
D. Pengujian Model Kedua

Pada pengujian model kedua dibuat sebuah kueri dengan kondisi yang memilih beberapa kolom dengan syarat *user_location* dan *user_bio* "IS NOT NULL" serta *user_follower_count* dan *user_friends_count* ≥ 1 , yang diurutkan berdasarkan *user_created_at* dari yang terendah. Berikut ini merupakan kueri pengujian pada Google BigQuery,

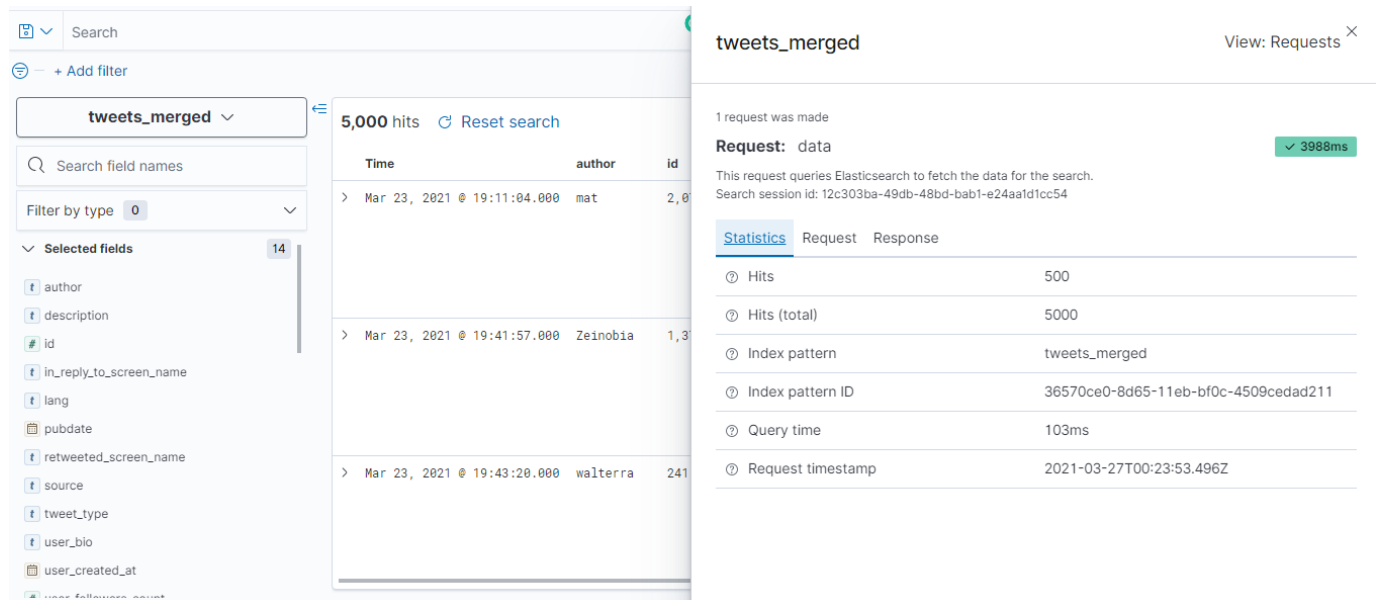
```
SELECT
author as screen_name, id, user_created_at, user_location_lang,
user_bio as Description, description as Text,
source, user_followers_count, user_friends_count,
retweeted_screen_name, tweet_type, in_reply_to_screen_name,
pubdate as Tweet_Create,
FROM 'benchmarkdata.tweets_merged.Tweets_corona'
WHERE user_location IS NOT NULL AND user_bio IS NOT NULL
AND user_followers_count  $\geq 1$  AND user_friends_count  $\geq 1$ 
ORDER BY user_created asc
```

Sementara itu, di bawah ini merupakan bagian kueri yang digunakan pada pengujian data di Elasticsearch, dengan diberikan kondisi yang memiliki kolom dengan beberapa syarat.

```
{
  "range": {
    "user_friends_count": {
      "gte": "1"
    }
  }
},
"minimum_should_match": 1
},
{
  "bool": {
    "should": [
      {
        "range": {
          "user_followers_count": {
            "gte": "1"
          }
        }
      }
    ],
    "minimum_should_match": 1
  }
}
},
{
  "range": {
    "@timestamp": {
      "gte": "2021-03-23T05:46:19.000Z",
      "lte": "2021-03-23T13:53:04.000Z",
      "format": "strict_date_optional_time"
    }
  }
},
"should": [],
"must_not": [
  {
    "match_phrase": {
      "user_location": "NULL"
    }
  },
  {
    "match_phrase": {
      "user_bio": "NULL"
    }
  }
}
```



Gbr. 1 Log model pertama pada Google BigQuery.



Gbr. 2 Log model pertama pada Elasticsearch.

E. Alat dan Bahan

Spesifikasi laptop yang digunakan dalam pengujian ini adalah sebagai berikut.

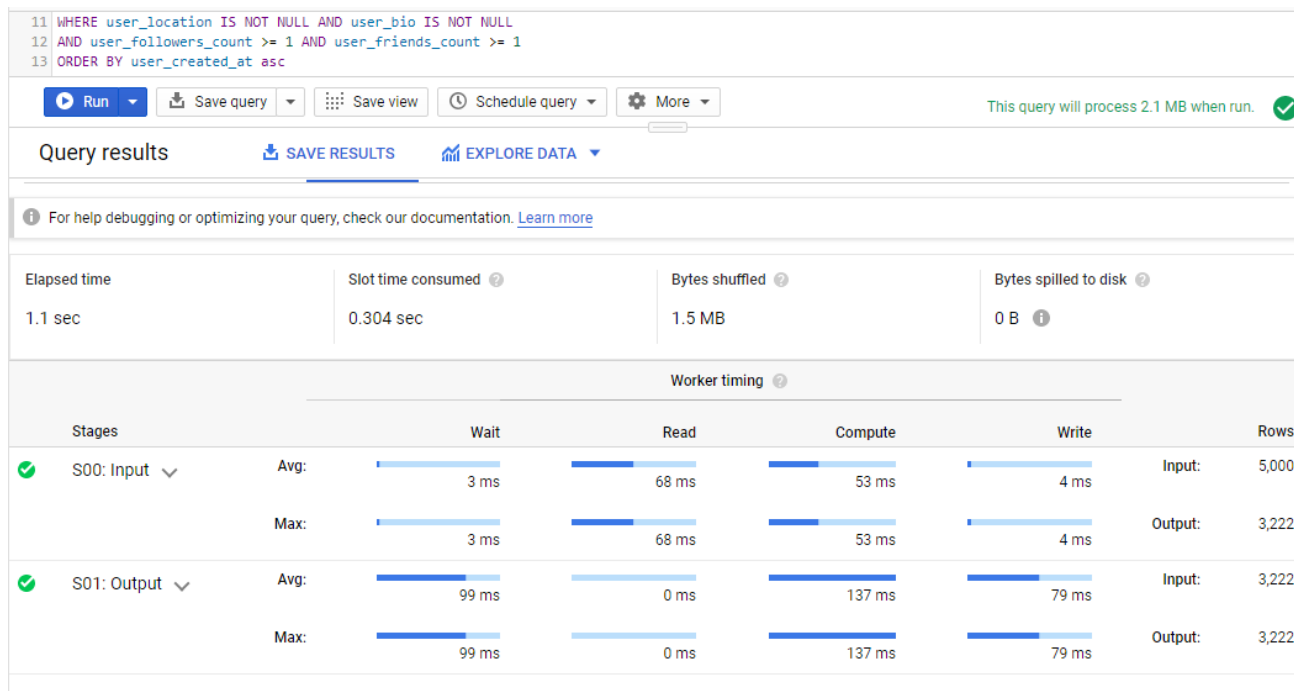
- Intel(R) Core (TM) i3-2310M CPU @ 2.10GHz 2.10GHz
- Sistem operasi 64-bit
- Memori 4GB(RAM)
- 5.000 cuitan dari Twitter yang mengandung tagar #COVID19 dan #coronavirus.

III. HASIL DAN PEMBAHASAN

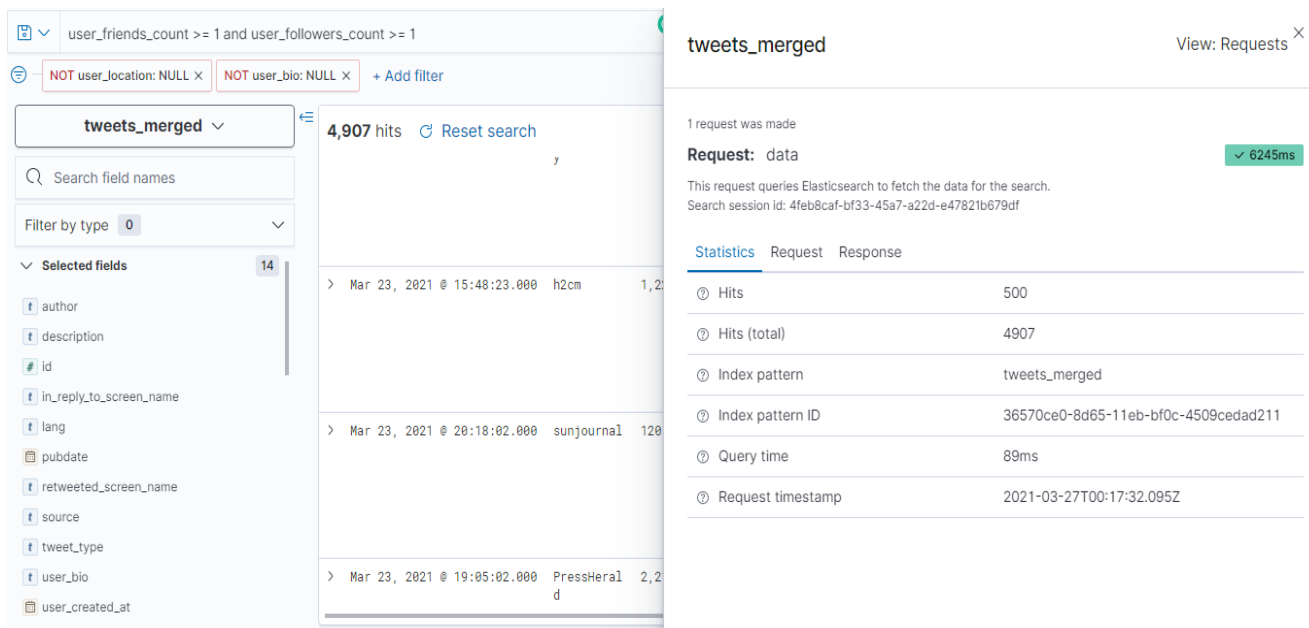
Pada tahapan *crawling* data, hasil yang didapatkan adalah 5.000 cuitan yang memiliki tagar #COVID19 dan #coronavirus

dengan total tiga puluh kolom. Pengujian pertama dilakukan untuk membandingkan Google BigQuery dan Elasticsearch dalam mengelola suatu kueri. Sebelumnya, pada Google BigQuery dibuat tabel dengan cara mengimpor data CSV secara lokal dengan besar file maksimal 10 MB. Jika ukuran melebihi 10 MB, dilakukan impor melalui GCS yang tidak memiliki batas maksimal dalam ekspor data. Elasticsearch mengimpor data secara lokal dengan batas maksimal tepatnya 1.024 MB.

Gbr. 1 dan Gbr. 2 merupakan hasil pengolahan kueri pada kedua *platform*. Hasil yang diambil adalah *query time*, yang merupakan waktu yang dibutuhkan dalam pengelolaan kueri, dan *total time*, yang merupakan waktu keseluruhan permintaan



Gbr. 3 Log model kedua pada Google BigQuery.



Gbr. 4 Log model kedua pada Elasticsearch.

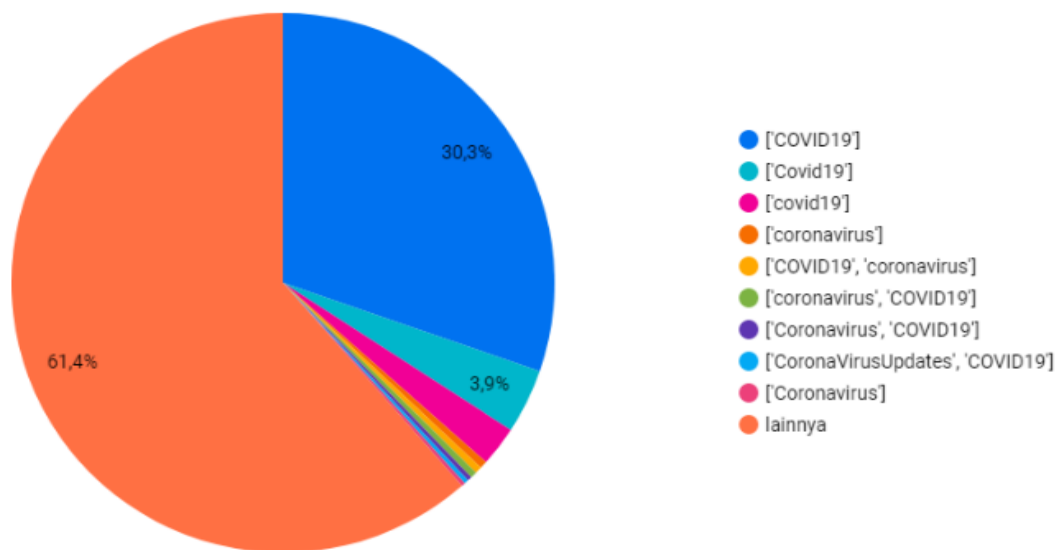
TABEL I
SELECTION TIME

	Google BigQuery (ms)	Elasticsearch (ms)
Query Time	141	103
Total Time	1400	3988
Outcome Row	5.000	5.000

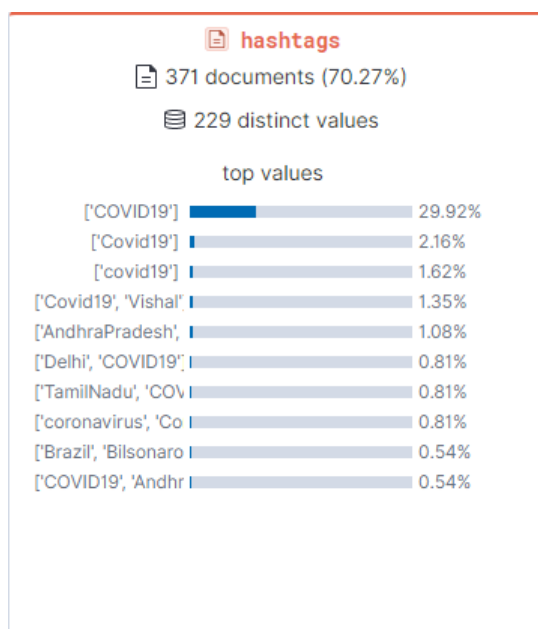
pada website. Hasil keluaran pada pengujian model pertama dari kedua platform yang didapatkan disajikan pada Tabel I. Hasil query time pada Google BigQuery adalah 141 ms, sedangkan pada Elasticsearch membutuhkan waktu 103 ms.

Lalu, untuk total waktu keseluruhan, Google BigQuery membutuhkan 1.400 ms dan Elasticsearch membutuhkan waktu 3.988 ms. Kedua platform ini menghasilkan outcome row yang sama, yaitu sejumlah 5.000 baris.

Gbr. 3 dan Gbr. 4 merupakan hasil kueri pengujian kedua. Hasil yang didapatkan pada pengujian model kedua ditunjukkan pada Tabel II. Hasil query time dari Google BigQuery adalah sebesar 79 ms, sedangkan pada Elasticsearch dibutuhkan waktu 89 ms. Total time yang dibutuhkan pada Google BigQuery sebesar 1.100 ms, sedangkan Elasticsearch membutuhkan waktu yang lebih lama, yaitu 6.245 ms.



Gbr. 5 TOP hashtag pada Google BigQuery.



Gbr. 6 TOP hashtag pada Elasticsearch.

TABEL II
CONDITION TIME

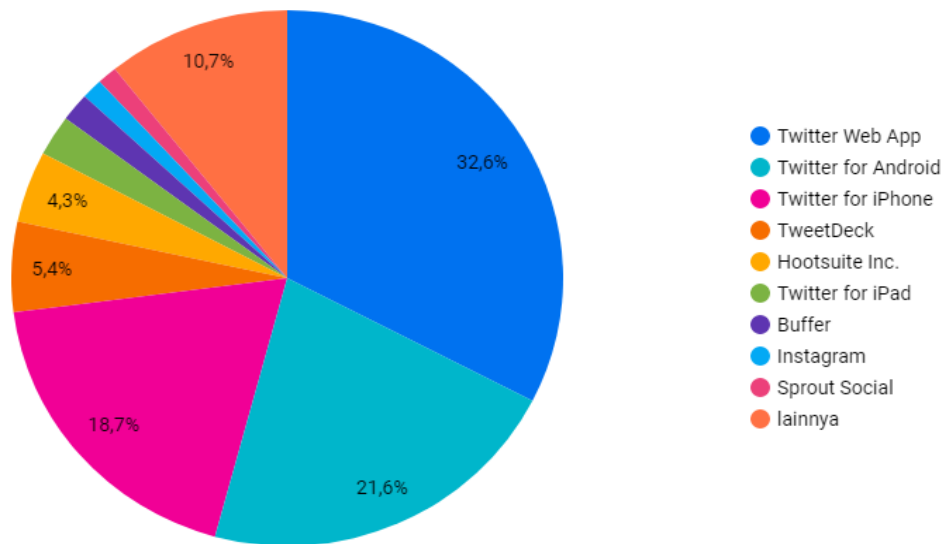
	Google BigQuery (ms)	Elasticsearch (ms)
Query Time	79	89
Total Time	1.100	6.245
Outcome Row	3.222	4.907

Sementara itu, pada *outcome row* yang didapatkan Google BigQuery sejumlah 3.222 baris, lebih sedikit dibandingkan Elasticsearch yang berjumlah 4.907 baris.

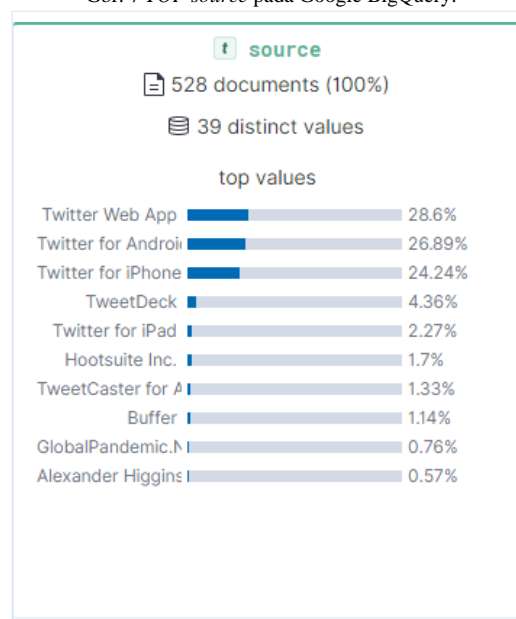
Pada visualisasi data, Google BigQuery memiliki Data Studio, sedangkan pada Elasticsearch sudah menjadi satu dengan Kibana, seperti pada Gbr. 5 dan Gbr. 6 yang menunjukkan *TOP hashtag* dari *dataset*. Kedua *platform* tersebut menunjukkan jumlah persentase yang cukup berbeda.

Google BigQuery memvisualisasikan 61,4% tagar yang digunakan dalam cuitan adalah 'lainnya' atau dapat dikatakan sebagai cuitan dengan tagar yang beragam dan 30,3% merupakan cuitan yang membahas COVID19. Sementara itu, pada Elasticsearch, persentase tertinggi terdapat pada tagar #COVID19, dengan jumlah persentase 29,92%.

Selanjutnya, visualisasi pada Gbr. 7 dan Gbr. 8 menunjukkan *TOP source* dari *dataset* pada kedua *platform* yang memiliki hasil sedikit berbeda. Pengguna yang menggunakan Twitter Web App pada Google BigQuery memiliki persentase 32,6%, sedangkan pada Elasticsearch memiliki persentase 28,6%. Urutan kedua dan ketiga diduduki *source* yang sama pada kedua *platform*, tetapi nilai persentase pada Elasticsearch lebih tinggi dibandingkan pada Google BigQuery.



Gbr. 7 TOP source pada Google BigQuery.



Gbr. 8 TOP source pada Elasticsearch.

IV. KESIMPULAN

Penelitian ini dilakukan untuk mengetahui perbandingan kinerja dari *platform* Google BigQuery dan Elasticsearch *Kibana*. Beberapa perbedaan bisa terlihat dari kedua *platform* ini, dari besar *input* data yang dapat dilakukan masing-masing *platform*, Google BigQuery lebih unggul karena dapat mengimpor data jauh lebih besar dari Elasticsearch. Walau disaat memvisualisasikan jumlah tagar dan *source* keduanya memiliki persentase yang hampir sama. Pada pengujian kueri pada kedua *platform* menunjukkan hasil dalam pemrosesan data dengan menggunakan Google BigQuery lebih cepat dan stabil dibandingkan dengan Elasticsearch. Pada pengujian model kedua memiliki pengaruh pada kinerja *platform* terutama pada Elasticsearch yang membutuhkan waktu pengelolaan cukup lama kurang lebih dua kali lipat dibandingkan pengujian model

pertama, sedangkan pada Google BigQuery cenderung lebih cepat dan stabil.

REFERENSI

- [1] E. Pratama dan I.P. Agus, *Handbook Data Warehouse*. Bandung, Indonesia: Informatika Bandung, 2018.
- [2] C. Dobre dan F. Xhafa, "Intelligent Services for Big Data Science," *Futur. Gener. Comput. Syst.*, Vol. 37, hal. 267–281, 2014.
- [3] V.K. Jain dan S. Kumar, "Big Data Analytic Using Cloud Computing," *Proc. - 2015 2nd IEEE Int. Conf. Adv. Comput. Commun. Eng. (ICACCE 2015)*, 2015, hal. 667–672.
- [4] H. Tankovska (2021) "Global Social Networks Ranked by Number of Users 2021," [Online], <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>, tanggal akses: 31-Mar-2021).
- [5] P. Srivastava dan R. Khan, "A Review Paper on Cloud Computing," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, Vol. 8, No. 6, hal. 17-20, 2018.
- [6] J. Tigani dan S. Naidu, *Google BigQuery Analytics*, Hoboken, USA:

- Wiley, 2014.
- [7] O. Dawelbeit dan R. McCrindle, "Efficient Dictionary Compression for Processing RDF Big Data Using Google BigQuery," *Proc. 2016 IEEE Glob. Commun. Conf. (GLOBECOM 2016)*, 2016, hal. 1–6.
- [8] V.A. Zamfir, M. Carabas, C. Carabas, dan N. Tapus, "Systems Monitoring and Big Data Analysis Using the Elasticsearch System," *Proc. - 2019 22nd Int. Conf. Control Syst. Comput. Sci. (CSCS 2019)*, 2019, hal. 188–193.
- [9] Y.C. Tay, "Data Generation for Application-Specific Benchmarking," *Proc. VLDB Endow.*, Vol. 4, No. 12, hal. 1470–1473, 2011.
- [10] A. Bog, *Benchmarking Transaction and Analytical Processing Systems*. Cham, Switzerland: Springer, 2014.
- [11] D.O. Baskoro, *Big Data Benchmark pada Hadoop 2, Spark, dan Presto Menggunakan Metode Perbandingan Waktu Respon Query*. Skripsi, Universitas Gadjah Mada, Yogyakarta, Indonesia, 2015.
- [12] P.P.I. Langi, Widyawan, W. Najib, dan T.B. Aji, "An Evaluation of Twitter River and Logstash Performances as Elasticsearch Inputs for Social Media Analysis of Twitter," *Proc. 2015 Int. Conf. Inf. Commun. Technol. Syst. ICTS 2015*, 2016, hal. 181–186.
- [13] S. Gupta dan R. Rani, "A Comparative Study of Elasticsearch and CouchDB Document Oriented Databases," *Proc. Int. Conf. Inven. Comput. Technol. (ICICT 2016)*, 2016, hal. 1–4.
- [14] A.U. Abdullahi, R. Ahmad, dan N.M. Zakaria, "Big data: Performance Profiling of Meteorological and Oceanographic Data on Hive," *Proc. 2016 3rd Int. Conf. Comput. Inf. Sci. (ICCOINS 2016)*, 2016, hal. 203–208.
- [15] C. Kotas, T. Naughton, dan N. Imam, "A Comparison of Amazon Web Services and Microsoft Azure Cloud Platforms for High Performance Computing," *2018 IEEE Int. Conf. Consum. Electron. (ICCE 2018)*, 2018, hal. 1–4.