

Implementasi Sistem Distribusi Pesan dan Proses Data Secara Real Time dengan Apache Kafka

Fezan Nabawi

Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nahdlatul Ulama Indonesia

Correspondence author: vzan.smile@gmail.com

DOI : <https://doi.org/10.37012/jtik.v8i1.836>

Abstrak

Didunia sekarang ini, informasi *real time* sangat dibutuhkan, dan informasi *real time* dihasilkan oleh aplikasi diantaranya dengan aplikasi kafka, Kafka adalah aplikasi untuk *publish* dan *subscribe*. Ini adalah sistem perpesanan yang awalnya dikembangkan di LinkedIn untuk mengumpulkan dan mengirimkan pesan. Kafka sendiri merupakan aplikasi pendistribusi dan perantara pesan. Dalam penelitian ini, penulis mengambil tempat penelitian di lokasi PT.Adira Finance Jakarta, dan sistem operasi yang digunakan pada server kafka yaitu server linux centos 7.0. Tahapan dalam implementasi penelitian secara umumnya yaitu meng-instal dan mengkonfigurasi kafka, mengaktifkan server kafka, membuat topic, dan testing aplikasi producer dan consumernya. Dari tahapan tersebut berjalan dengan baik sehingga bisa mendistribusikan pesan atau data secara realtime.

Kata Kunci: Kafka, *publish subscribe*, *realtime*

Abstract

In today's world, real time information is needed, and real time information generated by applications includes the kafka application, Kafka is an application to publish and subscribe. It is a messaging system originally developed on LinkedIn to collect and deliver messages. Kafka itself is a message distribution and intermediary application. In this study, the authors took the place of research at the location of PT. Adira Finance Jakarta, and the operating system used on the kafka server was the Centos 7.0 linux server. The stages in the general implementation are installing and configuring kafka, activating the kafka server, creating topics, and testing producer and consumer applications. From these stages it goes well so that it can distribute messages or data in real time.

Keywords: Kafka, Publish Subscribe, Real Time

PENDAHULUAN

Perkembangan teknologi informasi di era digital saat ini sangat cepat dan pesat, sehingga mempengaruhi segala aspek dibidang kehidupan, terutama pertukaran sebuah informasi dan komunikasi menjadi lebih cepat, mudah dan real time.

Didunia sekarang ini, informasi real time sangat dibutuhkan, dan informasi real time dihasilkan oleh aplikasi, kemudian untuk mengolah informasi ini perlu cara yang andal dan cepat untuk bisa dialihkan ke berbagai jenis penerima. Aplikasi yang mengkonsumsi informasi ini terpisah sehingga tidak dapat diakses satu sama lain, jadi perlu pembangunan untuk menyediakan titik integrasi di antara aplikasi producer yang disini bertindak sebagai pengirim pesan atau data ke consumer, dimana consumer yang *subscribe* inilah yang akan menerima pesan secara *uptodate* dan *realtime*. *Real time* adalah pesan yang dihasilkan oleh produser harus segera terlihat oleh consumer, fitur ini sangat penting untuk sistem berbasis peristiwa seperti sistem pemrosesan peristiwa

Producers mempublikasikan data ke topik pilihan mereka. Producer bertanggung jawab untuk memilih pesan mana yang akan ditetapkan ke partisi mana dalam topik. Sedangkan kumpulan consumer dapat membaca dari *server* dan setiap pesan masuk ke salah satu dari mereka, di *publish-subscribe* pesan di *broadcast* ke consumer. Setiap pesan yang dipublikasikan ke suatu topik dikirimkan ke satu consumer dalam setiap kelompok consumer yang berlangganan [1].

Apache kafka merupakan salah satu aplikasi *message service/broker* atau aplikasi untuk *publish* dan *subscribe*. *Message broker* bertujuan untuk mengirimkan pesan dari sebuah producer yang akan di terima oleh sebuah consumer. Kafka ini adalah sistem perpesanan yang awalnya dikembangkan di LinkedIn untuk mengumpulkan dan mengirimkan pesan. Kafka sendiri merupakan aplikasi perantara pesan.

Apache kafka juga merupakan proyek *open-source* yang dikembangkan oleh Apache Software Foundation yang ditulis menggunakan bahasa pemrograman Scala dan Java. yang membuatnya sangat berharga untuk infrastruktur perusahaan untuk memproses data secara streaming. Kafka terhubung dengan sistem eksternal untuk mengimpor atau mengeksport data melalui kafka connect. Kafka connect adalah *Application Programming Interface* (API) untuk integrasi data antara Kafka dengan sistem data lainnya. Saat ini *server* kafka masih bergantung dengan zookeeper. Zookeeper disini bertugas untuk mengatur dan mengkoordinasikan *server* kafka.

Layanan zookeeper terutama digunakan untuk memberi tahu producer dan consumer tentang kehadiran *broker* baru di sistem kafka atau untuk mengetahui kegagalan broker di sistem kafka [2].

METODE PENELITIAN

Menurut Sugiyono Pengertian metode penelitian adalah cara ilmiah untuk mendapatkan data dengan tujuan dapat dideskripsikan, dibuktikan, dikembangkan dan ditemukan pengetahuan, teori, untuk memahami, memecahkan, dan mengantisipasi masalah dalam kehidupan manusia (Sugiyono: 2012) [3].

A. Teknik pengumpulan data

Teknik pengumpulan data atau metode pengumpulan data yang digunakan dalam penelitian ini adalah :

1. Observasi, dilakukan dengan melakukan pengamatan secara langsung di objek penelitian.
2. Studi Pustaka, dilakukan untuk mencari dan mengumpulkan data yang dapat mendukung penulis dalam menyelesaikan penelitian ini. Data ini dapat berupa bahan-bahan pendukung seperti teori-teori, konsep-konsep yang berasal dari literatur-literatur.

B. Tempat dan Waktu Penelitian

Dalam penelitian ini, penulis mengambil tempat penelitian di lokasi PT.Adira Finance Jakarta

C. Bahan dan Peralatan

Dalam mengerjakan penelitian ini, penulis menggunakan peralatan laptop Lenovo L530 dengan spesifikasi Intel processor Core i3, Memory 10 GB DDR3, HDD 320 GB, dan sistem operasi desktop menggunakan Windows 10.

Sedangkan komputer yang berfungsi sebagai server kafka, dan sekaligus server lense untuk monitoring ini menggunakan virtualisasi yakni dengan virtualbox dimana spesifikasinya RAM 6 GB, HDD 30 GB, 2 buah ethernet card virtual (Bridge & NAT).

D. Prosedur Penelitian

Terdapat beberapa prosedur yang dilakukan dalam perancangan dan implementasi aplikasi ini. Pertama dengan studi literatur, dimana merupakan prosedur untuk mendapatkan artikel tentang

kafka mulai dari pengenalan kafka, instalasi, dan konfigurasi. Kedua mempelajari sistem jaringan untuk digambarkan topologi dan alurnya agar bisa komunikasi antar aplikasi, dan bisa mendistribusikan pesan dari producer ke consumer serta membuka port-port yang dibutuhkan dalam hal ini akan berhubungan dengan firewall untuk meng-allow port tersebut. Ketiga implementasi sistem, dengan menganalisis kebutuhan perangkat lunak dari sistem yang dirancang dan menginstalasi. Keempat yaitu melakukan monitoring kafka untuk melihat bahwa distribusi sudah berjalan dengan baik

E. Perancangan Sistem

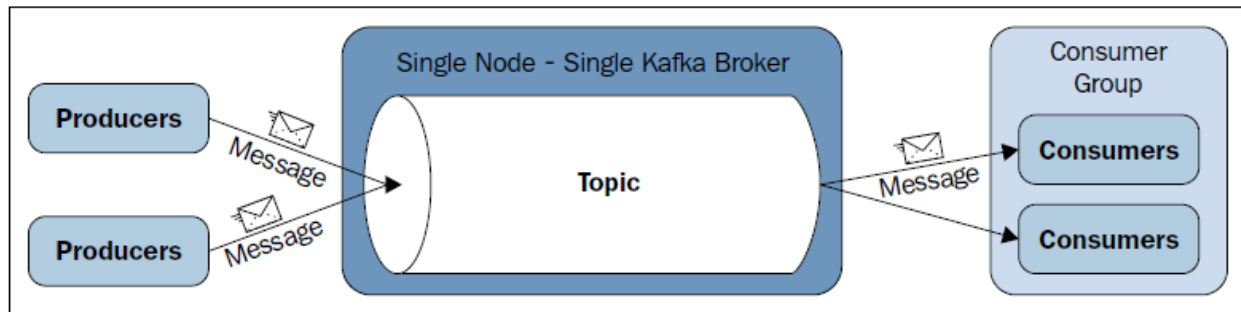
Dalam pembuatan system kafka ini atau dalam penelitian ini penulis merancang sebuah laptop menjadi server Kafka sekaligus sebagai server monitoring kafka dengan menggunakan virtualbox dimana virtual mesinnya menggunakan sistem operasi Linux Centos 7.0. berikut ini adalah tahapan secara umumnya:

1. meng-instal sistem operasi linux centos 7.0 di sebuah virtual machine,
2. menseting IP address dari server kafka untuk hubungan komunikasi keluar dengan perangkat lain,
3. meng-instal aplikasi kafka,
4. mengkonfigurasi kafka dengan menseting zookeeper.properties dan server.properties,
5. mengaktifkan zookeeper, dan kafka,
6. membuat topic,
7. testing producer dan consumer
8. install lense untuk monitoring server kafka

HASIL DAN PEMBAHASAN

A. Dasar-dasar desain Kafka

Dalam struktur yang sangat dasar, produser menerbitkan pesan ke topik kafka, yaitu dibuat di broker kafka yang bertindak sebagai server kafka. Consumer kemudian berlangganan topik kafka untuk mendapatkan pesan. Hal ini dijelaskan dalam diagram berikut:



Arsitektur ini menganggap bahwa ketiga pihak tersebut yakni : producer, broker kafka, dan consumer berjalan di mesin yang berbeda. Di sini, setiap konsumen direpresentasikan sebagai sebuah proses dan proses-proses ini diatur dalam kelompok yang disebut kelompok consumer. Sebuah pesan dikonsumsi oleh satu proses consumer dalam kelompok consumer.

Fakta penting dalam *desain* kafka adalah sebagai berikut:

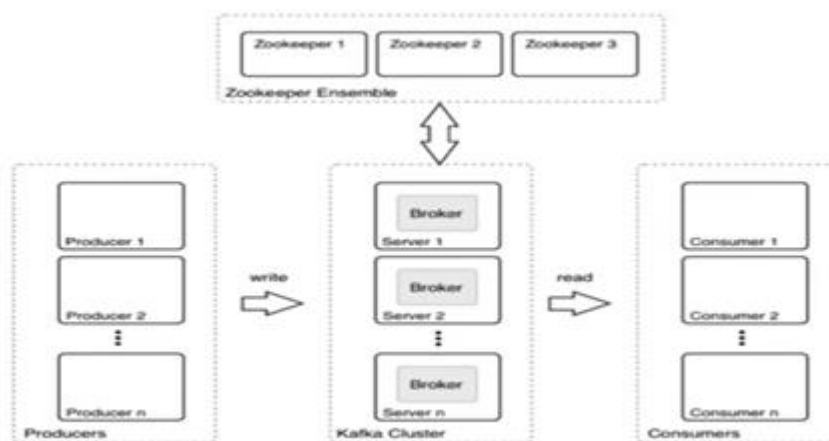
- Tulang punggung dasar kafka adalah caching pesan dan menyimpannya di sistem file. Di kafka, data segera ditulis ke halaman kernel OS.
- Kafka memberikan retensi pesan yang lebih lama setelah dikonsumsi, memungkinkan consumer untuk mengkonsumsi kembali, jika diperlukan.
- Kafka menggunakan set pesan untuk mengelompokkan pesan untuk memungkinkan overhead jaringan yang lebih rendah.
- Tidak seperti kebanyakan sistem pesan, di mana metadata dikonsumsi pesan disimpan di tingkat server, di kafka status dikonsumsi pesan dipertahankan pada tingkat consumer. Ini juga membahas masalah-masalah seperti:
 - Kehilangan pesan karena kegagalan
 - Beberapa pengiriman pesan yang sama
- Secara default, konsumen menyimpan status di zookeeper, tetapi kafka juga mengizinkan menyimpannya dalam sistem penyimpanan lain yang digunakan untuk transaksi pemrosesan online.
- Konfigurasi durasi waktu data disimpan, defaultnya adalah 7 hari
- Di kafka, producer dan consumer bekerja dengan cara tradisional push-and-pull model, di mana producer mendorong pesan ke broker kafka dan consumer tarik pesan dari broker.
- Kafka tidak memiliki konsep master dan memperlakukan semua broker sebagai rekan-rekan. Pendekatan ini memfasilitasi penambahan dan penghapusan broker kafka di poin

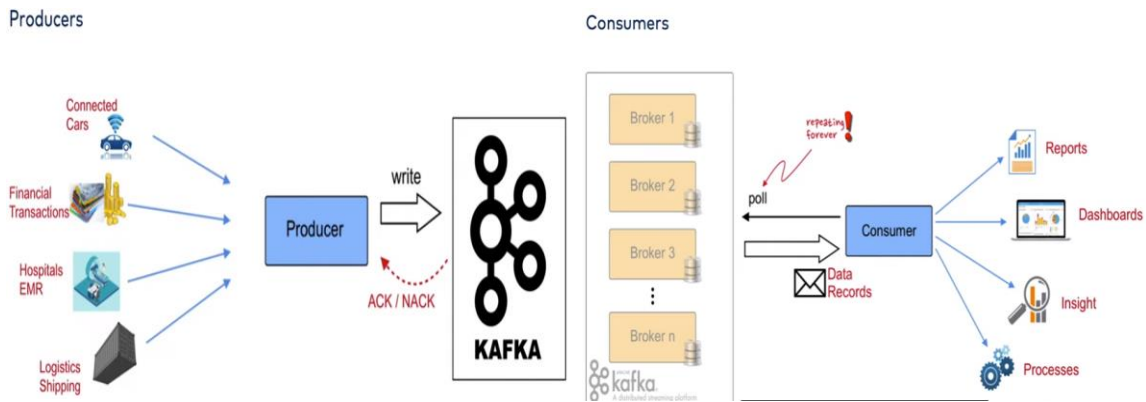
apa pun, karena metadata broker dipertahankan di zookeeper dan dibagikan kepada producer dan consumer.

- Producer juga memiliki opsi untuk memilih antara asinkron atau mode sinkron untuk mengirim pesan ke broker [4].

B. Arsitektur dan Desain Kafka

Producer adalah yang mempublikasikan pesan ke topik kafka, Suatu server kafka dapat menerima pesan dari satu atau lebih producer. Consumer yang berlangganan topik dan menerima umpan pesan yang diterbitkan. Cluster adalah terdiri dari dua atau lebih server atau dimana sekumpulan brokers dikelompokkan, cluster untuk mendukung scalability, high availability dan fault-tolerant, sedangkan server-server disebut broker. Topics adalah tempat dimana pesan-pesan atau data dikelompokkan. Dalam satu broker bisa terdapat banyak topic. Partitions adalah pembagian penyimpanan untuk pesan dalam suatu topics. Offset adalah posisi terakhir dari suatu urutan pesan yang masuk ke dalam suatu partisi di dalam topic.





C. Kelebihan utama dari Apache Kafka adalah:

- Distributed

Apache Kafka dapat menyimpan, menerima dan mengirim data atau pesan dari berbagai node.

- Horizontally-scalable

Apache Kafka dapat bekerja dengan baik dalam suatu kelompok atau cluster, sehingga dengan bertambahnya volume dan kecepatan aliran data yang harus diproses, maka kita hanya perlu menambah mesin baru saja pada cluster tanpa harus melakukan vertical-scaling.

- Fault-tolerant

Salah satu kelemahan dari sistem sentralisasi (tidak terdistribusi) adalah Single Point of Failure (SPoF). Sedangkan sistem terdistribusi dirancang untuk mengakomodasi jika terjadi masalah pada satu atau lebih node, maka node yang lain dapat menggantikan posisi node yang bermasalah tersebut, sehingga keseluruhan sistem tidak terganggu. Pada Apache Kafka, terdapat satu partition leader yang berperan sebagai node utama, dimana partition leader mereplikasi data yang diterimanya ke node-node lainnya, yang disebut sebagai followers [5].

D. Bagaimana Kafka Bekerja

Producer menulis data ke topik, kemudian kafka menyimpan pesan dalam topik dan consumer membaca dari topik. Karena kafka adalah sistem terdistribusi, topik dipartisi, setiap consumer dalam kelompok consumer akan membaca pesan dari subset unik dari partisi di setiap topik

yang mereka berlangganan, dan semua pesan dengan kunci yang sama tiba di consumer yang sama.

Kafka menyimpan semua pesan untuk jangka waktu tertentu, dan consumer bertanggung jawab untuk melacak lokasi mereka di setiap log. Akibatnya, kafka menyimpan sejumlah besar data consumers [6].

E. Kafka Connect

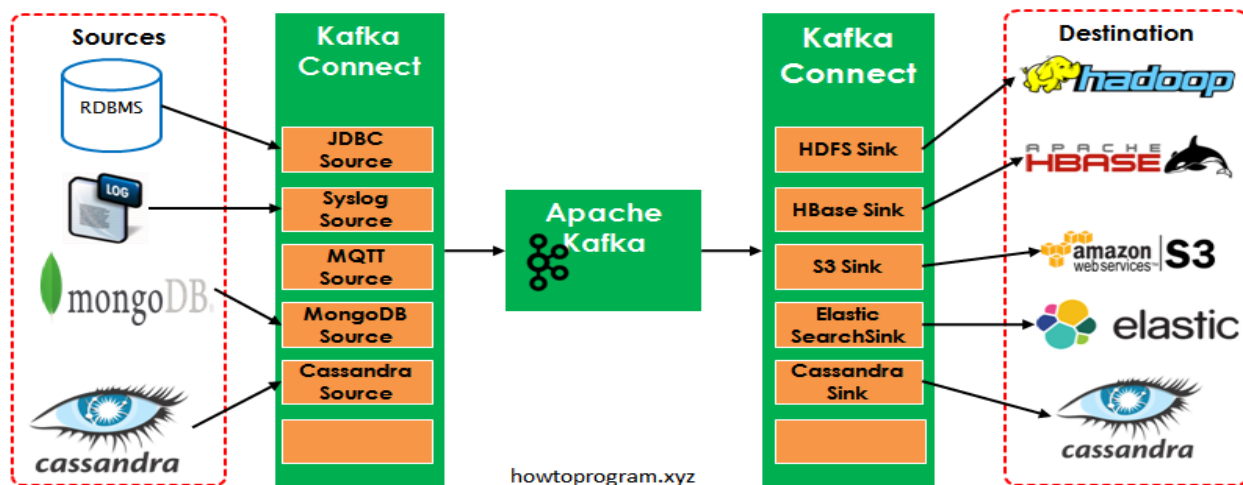
Kafka connect adalah API untuk integrasi data antara kafka dengan sistem data lainnya. Contoh sistem data yang dimaksud adalah database mysql, nosql mongodb, elasticsearch, big data, hadoop, dan lain sebagainya. Kafka connect source API adalah keseluruhan kerangka kerja yang dibangun di atas producer API, dibangun agar pengembang mendapatkan API yang lebih bagus yang dibuat untuk berbagai konektor yang tersedia yang dapat di manfaatkan hari ini untuk memasukkan data dari sebagian besar sumber Anda, tanpa menulis satu baris kode pun.

Dari sumber website resmi apache kafka, apache kafka memiliki 5 API inti yaitu:

1. Producer API memungkinkan aplikasi mengirim aliran data ke topik di kluster Kafka.
2. Consumer API memungkinkan aplikasi membaca aliran data dari topik di kluster Kafka.
3. Streams API memungkinkan transformasi aliran data dari topik masukan menjadi topik keluaran.
4. Connect API memungkinkan penerapan konektor yang secara terus-menerus menarik dari beberapa sistem sumber atau aplikasi ke Kafka atau mendorong dari Kafka ke beberapa sistem atau aplikasi .
5. Admin API memungkinkan pengelolaan dan pemeriksaan topik, broker, dan objek Kafka lainnya.

Apa itu API ?

API adalah sebuah interface yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya. Jadi, API berperan sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu platform yang sama atau lintas platform [7].



F. Implementasi Sistem

Ada empat tahapan dalam implementasi sistem untuk bisa mendistribusikan pesan/data secara real time, yaitu

1. instalasi kafka, dan konfigurasinya
2. membuat topic
3. producer kirim pesan
4. consumer menerima pesan



Untuk instalasi kafka digunakan spesifikasi yang telah di paparkan diatas. Laptop dengan virtual machine centoskafka akan berfungsi sebagai server kafka sekaligus server lense, sedangkan linux ubuntu berisi aplikasi producer dan consumer, yang dibuat dengan pemrograman python. Kemudian ethernet card virtual yang merupakan media untuk menghubungkan dua platform yang berbeda yakni platform linux dan windows

Prasyarat

- system operasi server Centos 7.0 dengan akses root
- server kafka membutuhkan minimum RAM 4GB untuk dijalankan.
- Install Java di server kafka

Langkah Instal Kafka

1. Instal java 8 atau diatasnya

 Java 8 Update 311 (64-bit)	Oracle Corporation	1/15/2022	127 MB	8.0.3110.11
 Java SE Development Kit 8 Update 311 (64-bit)	Oracle Corporation	1/15/2022	314 MB	8.0.3110.11

Perintahnya:

```
# yum install java-11-openjdk -y
```

2. Download kafka di <https://kafka.apache.org/downloads>,



```
#wget https://dlcdn.apache.org/kafka/3.1.0/kafka\_2.13-3.1.0.tgz
```

3. Setelah download kafka selesai, extract file tersebut, kemudian letakan filenya di folder yg diinginkan misal di path home/kafka

4. Ekstrak

```
#tar -xzf kafka_2.13-3.0.0.tgz
```

5. Kemudian masuk ke folder kafka hasil ekstrak

```
#cd kafka_2.13-3.0.0
```

6. Buka folder config dan edit file Zookeeper.properties dan Server.properties

```
#vi zookeeper.properties
```

```
dataDir= /home/kafka/zookeeper-data
```

```
#vi server.properties
```

```
log.dirs=/home/kafka/kafka-logs
```

```
listeners=PLAINTEXT://192.168.43.123:9092
```

```
zookeeper.connect=192.168.43.123:2181
```

7. Disable Firewall

```
# systemctl stop firewalld  
  
# systemctl disable firewalld
```

Pastikan firewalld inactive

```
# systemctl status firewalld
```

8. Start Zookeeper di folder kafka_2.13-3.0.0

```
#bin/zookeeper-server-start.sh config/zookeeper.properties
```

9. Start Kafka

```
# bin/kafka-server-start.sh config/server.properties
```

10. Create Topic

```
#bin/kafka-topics.sh --create --partitions 1 --replication-factor 1 --topic tes --bootstrap-server 192.168.43.123:9092
```

11. Menulis pesan dalam topik tes di console producer

```
#bin/kafka-console-producer.sh --topic tes --bootstrap-server localhost:9092
```

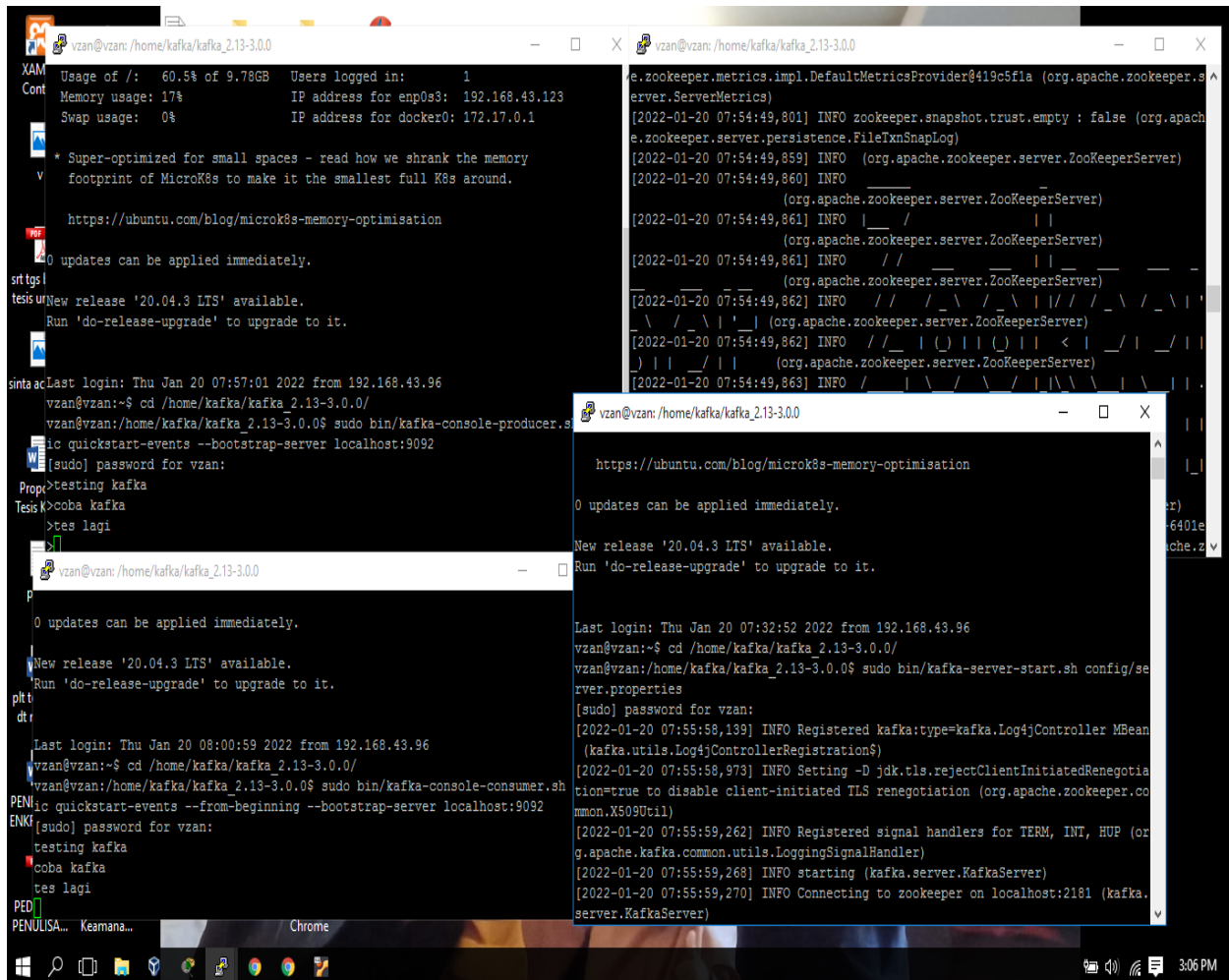
```
This is my first data  
This is my second data
```

12. Membaca pesan masuk dalam topik tes di console consumer

```
#bin/kafka-console-consumer.sh --topic tes --from-beginning --bootstrap-server localhost:9092
```

```
This is my first data  
This is my second data
```

→Terlampir lampiran hasil dari kafka (single) dibawah ini:



Instal python & dependensinya di Ubuntu 18.04:

#add-apt-repository universe

#apt-get update

#apt install python3-pip

#pip install kafka-python

a. Lampiran Kode Producer di Python

```
from kafka import KafkaProducer
```

```
producer = KafkaProducer(bootstrap_servers='192.168.43.123:9092')
```

```
if(producer.bootstrap_connected()):
    print("kafka connected")
else:
    print("failed to connect to kafka")
try:
    print("sending message");
    for i in range(10):
        producer.send('tes', b'raw_bytes')
        producer.flush();
        print("message sent");
except Exception as e:
    print("kafka error");
    print(str(e));
    pass
```

b. Lampiran Kode Consumer di Python

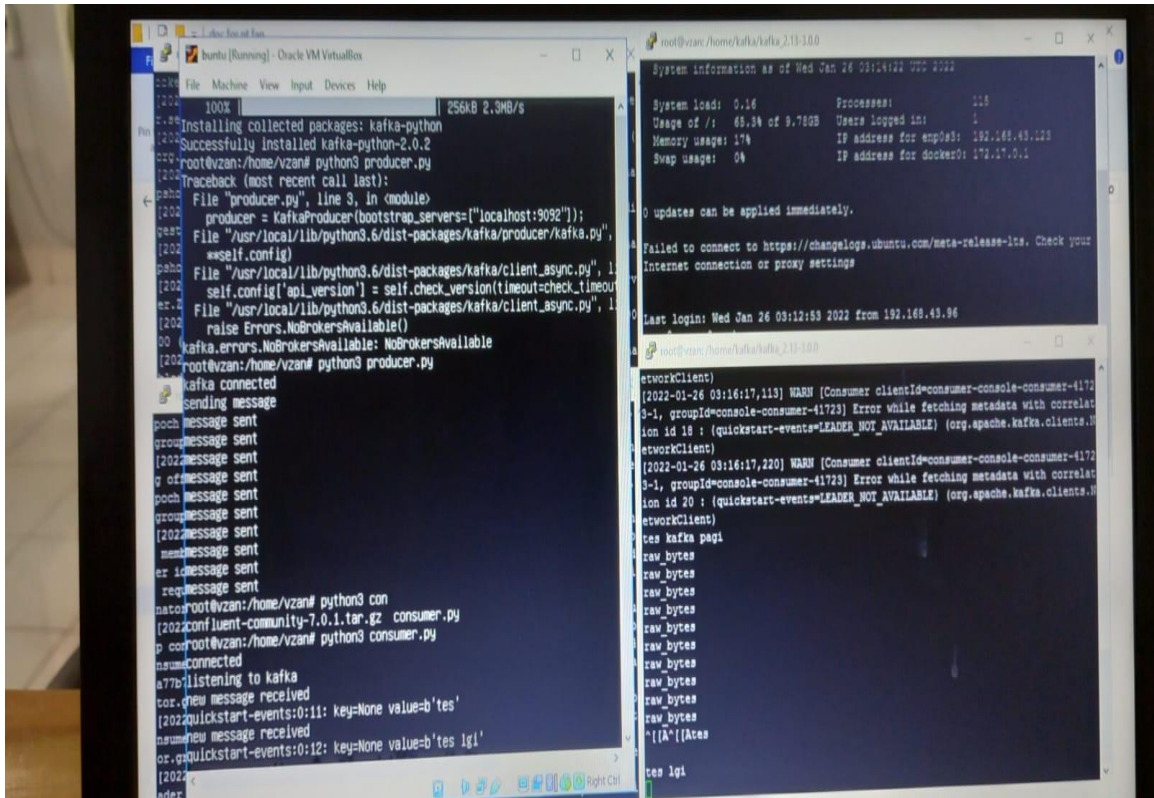
```
from kafka import KafkaConsumer
consumer = KafkaConsumer('tes',
                          bootstrap_servers='192.168.43.123:9092')
if(consumer.bootstrap_connected()):
    print("connected")
else:
    print("failed to connect")
print("listening to kafka")
for message in consumer:
    print("new message received")
    print ("%s:%d:%d: key=%s value=%s" % (message.topic, message.partition,
                                          message.offset, message.key,
                                          message.value))

KafkaConsumer(auto_offset_reset='earliest', enable_auto_commit=False)
KafkaConsumer(value_deserializer=lambda m: json.loads(m.decode('ascii')))
```

KafkaConsumer(value_deserializer=msgpack.unpackb)

KafkaConsumer(consumer_timeout_ms=1000)

c. Lampiran Hasil dengan Python berhasil



Instal Lenses

1. Buat folder

```
#mkdir /apps
```

2. Masuk ke folder

```
#cd /apps
```

3. Download lenses-4.3.6-linux64.tar.gz

```
#wget https://archive.lenses.io/lenses/4.3/lenses-4.3.6-linux64.tar.gz
```

```
--no-check-certificate
```

4. Masuk ke folder

```
#cd /apps/lenses-4.3.5
```

5.Copy

```
#cp lenses.conf.sample lenses.conf
```

```
#cp security.conf.sample ssecurity.conf
```

6.Konfig

```
#vi lenses.conf
```

```
lenses.ip = 0.0.0.0
lenses.port = 9991

# License file
lenses.license.file = "/etc/lenses/license.json"

# Directory for local storage (write access needed) - preserve between upgrades
lenses.storage.directory = "/var/lib/lenses/storage"

# Bootstrap brokers
lenses.kafka.brokers = "PLAINTEXT://host1:9092,PLAINTEXT://host2:9092,PLAINTEXT://host3:9092"
```

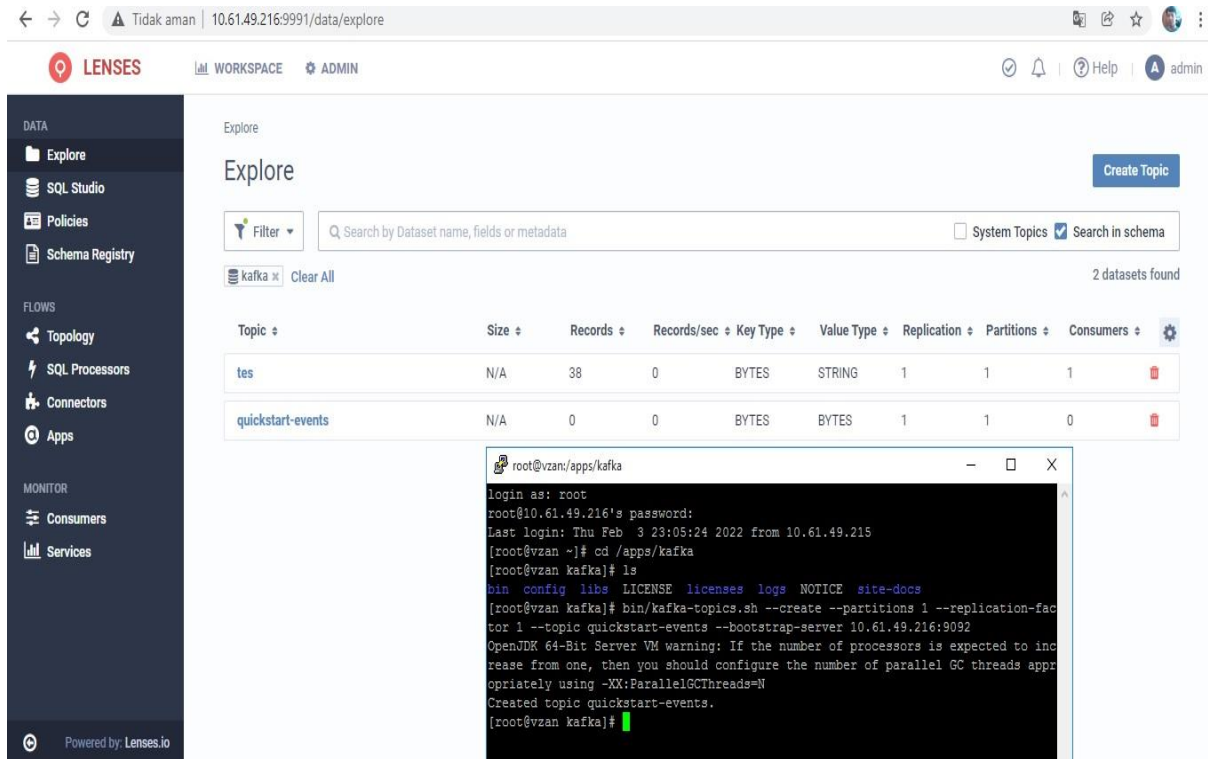
A quick start example `security.conf` looks like:

```
# A single admin user
lenses.security.user = "admin"
lenses.security.password = "admin"
```

7. Jalankan lenses

```
#bin/lenses
```

8. Cek ke browser -> IPAddress:9991 [9].



Diatas kami menjalankan eksperimen pada 2 mesin Linux, masing-masing dengan RAM 6 dan 1 GB. Kedua mesin terhubung dengan jaringan virtual. salah satu dari mesin digunakan sebagai broker dan mesin lainnya digunakan sebagai aplikasi producer atau consumer. Dan hasil implementasi tersebut berhasil berjalan dengan baik

KESIMPULAN

Kafka adalah solusi untuk masalah waktu nyata dari solusi perangkat lunak apa pun, yaitu, untuk menangani volume informasi waktu nyata dan mengarahkannya ke banyak konsumen dengan cepat. Kafka menyediakan integrasi tanpa batas antara informasi produsen dan konsumen tanpa menghalangi produsen informasi dan tanpa memberi tahu produsen siapa konsumen akhir [2].

Kafka menyediakan API yang mirip dengan sistem pesan dan memungkinkan aplikasi untuk menggunakan peristiwa log secara real time. Ini sangat menyederhanakan infrastruktur.

Kami menguji kafka API untuk producer dan consumer dengan menggunakan bahasa pemrograman python. Kafka menangani seluruh publikasinya dan berlangganan sistem pesan dengan bantuan empat API yaitu, API produsen, konsumen API, API streaming, dan API konektor.

REFERENSI

- [1] Thein, K. M. M. (2014). Apache kafka: Next generation distributed messaging system. *International Journal of Scientific Engineering and Technology Research*, 3(47), 9478-9483.
- [2] Shaheen, J. (2017). Apache Kafka: real time implementation with Kafka architecture review. *International Journal of Advanced Science and Technology*, 109, 35-42.
- [3] Salmaa. 2021. Pengertian Metode Penelitian, Tujuan, Macam, dan Contoh Lengkapnya. <https://penerbitdeepublish.com/pengertian-metode-penelitian/> dipublish 11 Februari 2021.
- [4] Garg, N. (2013). Apache kafka (pp. 30-31). Birmingham, UK: Packt Publishing.
- [5] MTI Binus. 2018. Big Data Messaging dengan Apache Kafka <https://mti.binus.ac.id/2018/06/28/big-data-messaging-dengan-apache-kafka/>. di publish 28 Jun 2018.
- [6] Momtselidze, N., & TSITSAGI, A. (2015). Apache kafka-real-time data processing. *Journal of Technical Science and Technologies*, 4(2), 31-34.
- [7] Amaldi, Wahyu. 2021. Apache KAFKA Adalah ? API Streaming data Pipeline. <https://ilmuprogram.com/2021/08/09/apache-kafka-api-streaming-data-pipeline/> dipublish 9 Agustus, 2021.
- [8] <https://kafka.apache.org/quickstart>
- [9] <https://docs.lenses.io/4.2/installation/linux/>