

PENGEMBANGAN SISTEM STREAMING SENSOR DATA TEMPERATUR DAN KELEMBAPAN DI RUANG LABORATORIUM BBTA3 MELALUI TEKNOLOGI OPEN SOURCE

SYSTEM DEVELOPMENT FOR STREAMING DATA SENSOR TEMPERATURE AND HUMIDITY IN BBTA3 LABORATORY USING OPEN SOURCE TECHNOLOGY

Arizal Akbar Zikri¹

¹ Balai Besar Teknologi Aerodinamika, Aeroelastika dan Aeroakustika – BPPT
Kawasan PUSPIPTEK Gedung 240, Setu, Tangerang Selatan – Banten, Indonesia

Abstraks

Aplikasi streaming data sensor berupa temperatur dan kelembapan di ruang server perlu dikembangkan, sehingga memudahkan bagi operator untuk memantau kondisi udara terkini secara dinamis di ruang server tanpa harus masuk kedalam ruang tersebut. Pengembangan aplikasi dilakukan menggunakan teknologi open source agar memudahkan pengembang untuk mandiri dan berinteraksi dalam komunitas tanpa terikat dengan permasalahan lisensi.

Pembacaan data sensor dilakukan oleh Raspberry Pi dan dijadikan sebagai producer untuk mengirimkan data tersebut ke Kafka Cluster. Kafka merupakan teknologi open source yang digunakan sebagai alat untuk streaming data terdistribusi. Satu node dalam cluster digunakan untuk menerima kiriman data atau dikenal sebagai consumer sekaligus menjalankan python servis untuk menangani permintaan dari pengguna aplikasi melalui Server Sent Event (SSE) dalam bentuk REST API.

Aplikasi ini diberi nama TempHum dan dapat dijalankan di Desktop (Windows, Linux, Mac OS), web browser, dan smartphone (Android dan iOS), sehingga aplikasi ini dapat melayani banyak pengguna dalam memantau kondisi ruang server secara dinamis.

Kata Kunci : open source, cluster, raspberry pi, kafka, python.

Abstract

Development of temperature and humidity data streaming Application from sensors in the server room is necessary to enable the operator to monitor the real-time air condition in the server room, without having to enter the room. This application development utilizes open source technology to make developers more independent and able to interact easily within the community without license hassle.

Sensor data reading is done by Raspberry Pi, assigned as a producer in sending data to Kafka Cluster. Kafka is an open source technology used as tools for streaming data distributedly. One node in a cluster is set to receive sensor data, known as consumer, executes python service to handle requests from users through Server Sent Event (SSE) in form of REST API.

This application is called TempHum and can be executed on Desktop (Windows, Linux, Mac OS), web browser, and smartphone (Android and iOS). Hence, the application can serve many clients in monitoring air condition in realtime.

Keywords: open source, cluster, raspberry pi, kafka, python.

I. PENDAHULUAN

Berdasarkan standar TIA-942-A halaman 92 tentang temperatur dan kelembapan udara ruang server, komputer atau server berada dalam ruangan dengan

* Arizal Akbar Zikri. Tel: +62 21 756 090 2
E-mail: arizal.akbar@bppt.go.id

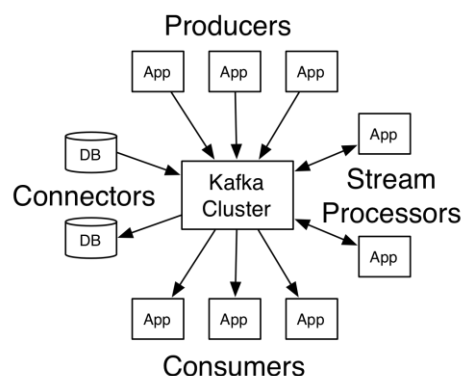
rentang temperatur 20°C hingga 25°C dan kelembapan 40% hingga 55% [1]. Operator atau teknisi mengakses server BBT3 secara tidak langsung atau remote (SSH). Mekanisme ini tidak membantu teknisi untuk memantau kondisi udara ruang server karena sistem operasi server tidak memiliki kemampuan membaca kondisi udara. Untuk itu, dibutuhkan komputer tambahan untuk memantau kondisi udara, yaitu *Raspberry Pi*.

Raspberry Pi merupakan komputer ukuran kecil yang dapat membaca temperatur dan kelembapan sekitar melalui sensor yang terpasang di *GPIO* (General Purpose Input Output). Sensor yang dipilih dan digunakan adalah sensor DHT22 yang dapat membaca temperatur dan kelembapan udara sekitar.

Hasil pembacaan data temperatur dan kelembapan dapat dilakukan setelah teknisi atau pengguna masuk ke sistem operasi raspberry pi melalui SSH. Cara ini tidak praktis dikarenakan pengguna tertentu saja yang diberi hak akses SSH. Untuk itu, diperlukan servis yang berjalan di server dan dapat diakses banyak pengguna aplikasi. Untuk mendukung banyak pengguna, servis dijalankan dalam server tersendiri dan terpisah dari raspberry pi. Dalam kasus ini, pengembangan servis dilakukan menggunakan bahasa *Python*.

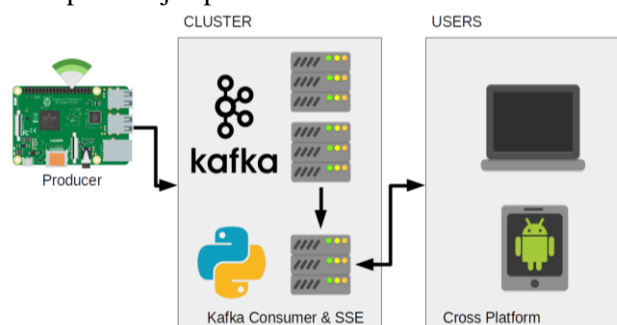
Agar komunikasi antara raspberry pi dengan server terjalin lancar, dibutuhkan suatu mekanisme *producer-consumer*. Raspberry pi berperan sebagai producer yang mengirimkan data sensor temperatur dan kelembapan ke server dan servis python sebagai consumer yang menerima, mengolah, dan mengirimkan data sensor ke pengguna aplikasi. Diantara producer dengan consumer dibutuhkan penghubung atau *broker* yang dijalankan dalam cluster sehingga pesan atau data yang diterima kemudian dikirimkan ke consumer secara terdistribusi dari masing-masing *node* dalam cluster. Teknologi tersebut dikenal dengan nama *Apache Kafka*.

Kafka merupakan teknologi open source yang digunakan untuk menerima sekaligus mengirimkan data atau pesan secara terdistribusi ke consumer [2]. Kafka dapat dijalankan lebih dari satu node dalam cluster sehingga teknologi ini mudah untuk memenuhi kebutuhan *high availability* dan *high durability* [3]. Pengguna tidak disarankan untuk langsung membaca data temperatur dan kelembapan dari raspberry pi dikarenakan spesifikasi yang dimiliki masih minimum untuk taraf server [4]. Sebagai pembanding, raspberry pi dijadikan sebagai server VPN dapat menangani 3 hingga 4 pengguna [5]. Untuk itu dibutuhkan kafka sehingga jumlah pengguna aplikasi dapat ditingkatkan seperti **Gambar 1**.



Gambar 1. Komunikasi producer, consumer, dan kafka cluster

Agar aplikasi ini dapat dijalankan di berbagai media atau *cross platform*, pengguna aplikasi dapat menerima data sensor melalui servis yang dibuat dengan Python dalam bentuk REST API. Servis yang disediakan Python adalah sebagai kafka consumer (menerima data sensor dari producer) dan mengirim data melalui *Server Sent Event* (SSE) ke pengguna secara terus menerus. Dengan demikian, pengguna dapat memantau kondisi udara terkini dalam ruang server secara terus menerus, kapan saja, dan dimana saja. **Gambar 2** memberikan ilustrasi pola kerja aplikasi secara umum.



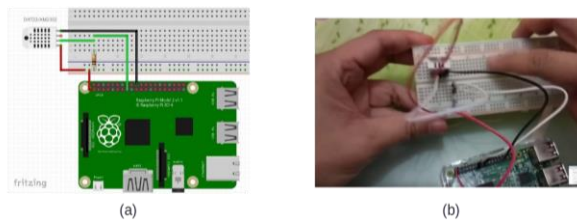
Gambar 2. Pola kerja aplikasi secara umum

Aplikasi ini diberi nama TempHum. Saat penulisan artikel ini, TempHum sudah diuji coba di sistem operasi Windows dan Linux [6]. Untuk menjalankan aplikasi TempHum di Android, dapat diunduh secara bebas di *Google Play* [7].

II. METODOLOGI

A. Raspberry Pi untuk kebutuhan kafka producer

Raspberry pi difungsi sebagai kafka producer dengan membaca data temperature dan kelembapan kemudian dikirimkan ke kafka cluster. Komponen yang dibutuhkan adalah resistor dengan nilai hambatan sebesar 10 k Ω , sensor DHT22, *breadboard*, dan beberapa kabel *jumper*. Sensor DHT22 dihubungkan ke *output pin* 3.3V, *input pin* data 22, dan *pin ground*. **Gambar 3** menunjukkan skematik pemasangan sensor di Raspberry Pi.



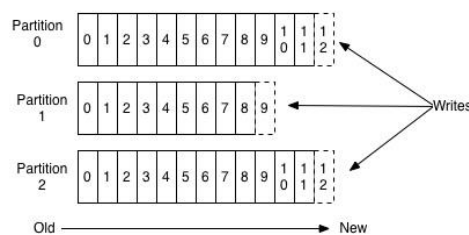
Gambar 3. (a) skematis sirkuit elektrik dan (b) pemasangan sirkuit di breadboard

Pembacaan data sensor berupa temperatur dan kelembapan di Raspberry Pi menggunakan bahasa Python. Data tersebut sudah dalam satuan derajat celcius untuk temperatur dan persentase untuk kelembapan. Data yang terbaca langsung dikirimkan ke kafka cluster dalam bentuk JSON.

B. Kafka Cluster

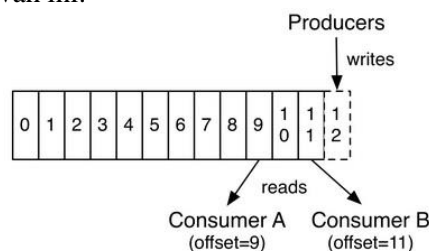
Kafka merupakan bagian dari teknologi open source yang digunakan untuk *streaming* data secara terdistribusi. Data sensor yang diterima dan dikirimkan ke consumer dalam bentuk *binary*.

Pengiriman atau pembacaan data sensor di kafka disimpan kedalam topik tertentu atau dapat dianalogikan kedalam database. Masing-masing topik yang terbentuk disimpan kedalam bentuk partisi. Jumlah minimal partisi yang dimiliki sebuah topik adalah satu. Masing-masing partisi dapat disimpan dan disebar ke banyak node dalam cluster. Data tersebut disimpan sesuai urutan penyimpanan sehingga data tersebut sesuai urutan atau dikenal dengan istilah *offset* oleh kafka. Anatomi sebuah topik di kafka ditunjukkan pada **Gambar 4**.



Gambar 4. Anatomi topik di kafka

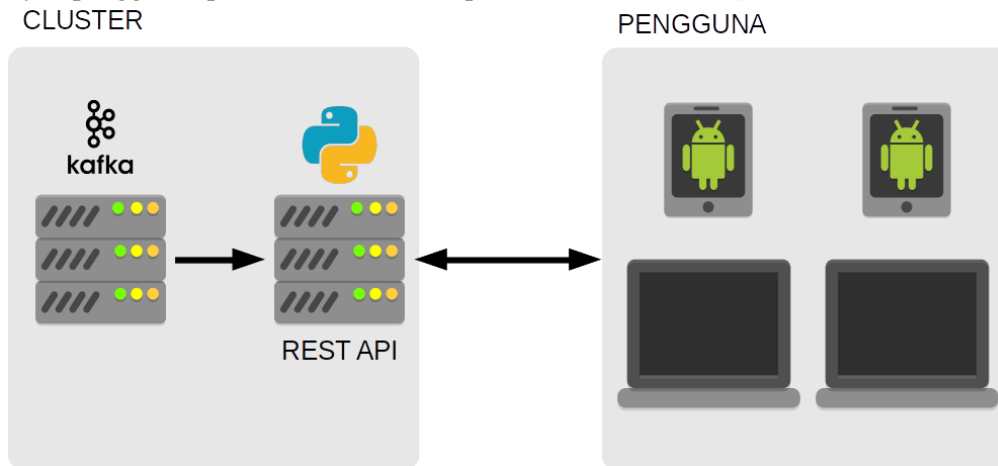
Bagi masing-masing kafka consumer, pembacaan data diatur oleh kafka sesuai *offset*. Hal ini diilustrasikan pada **Gambar 5** dibawah ini.



Gambar 5. Mekanisme pembacaan data oleh kafka consumer

C. Rest API dengan Python

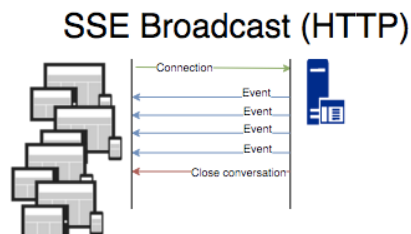
REST (*Representational State Transfer*) merupakan servis yang dijalankan oleh server sebagai penghubung antara pengguna aplikasi dengan streaming data kafka seperti **Gambar 6**. Servis ini dikembangkan dengan bahasa Python yang terdiri dari servis sebagai kafka consumer dan *Server Sent Event*. Servis ini dijalankan di server menggunakan delapan *worker* dengan komunikasi antara *worker* secara *asynchronous* melalui paket atau modul *gevent* di Python [8]. Hal ini dilakukan agar servis dapat melayani banyak pengguna aplikasi karena menerapkan mekanisme *multi processing*.



Gambar 6. Peran REST API sebagai penghubung pengguna aplikasi dengan server

Bagian dari servis Python, yaitu Server Sent Event (SSE). SSE merupakan koneksi yang terhubung satu arah (*unidirectional*) dari server yang mengirim data terus-menerus ke pengguna [9].

Dari sisi kehandalan, SSE lebih handal daripada teknologi *ajax polling* [10]. SSE menerima koneksi melalui *EventSource* untuk satu atau lebih pengguna. **Gambar 7** memberikan ilustrasi koneksi banyak pengguna ke servis yang menjalankan SSE.



Gambar 7. Koneksi SSE untuk satu atau lebih pengguna

III. HASIL DAN PEMBAHASAN

Berdasarkan standar yang disebutkan di bagian pendahuluan [1], mudah bagi operator atau pengguna untuk memantau kondisi udara terkini secara dinamis dikarenakan aplikasi ini memiliki fitur *streaming*. Proses streaming dapat dilakukan dengan bantuan teknologi kafka. Pembacaan data temperatur dan kelembapan dilakukan oleh Raspberry Pi dan sekaligus dijadikan sebagai kafka producer.

Raspberry Pi mengirimkan data temperatur dan kelembapan ke kafka cluster dengan *Content-Type* berjenis JSON dalam format binary seperti yang ditunjukkan dalam **Gambar 8**.

```
bash-4.4# bash kafka-console-consumer.sh --topic temphum --bootstrap-server kafk
a1:9614
{"hum": 44.900001525878906, "wkt": "2018-04-12 17:50:01.152639", "temp": 21.7999
99237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:50:06.685141", "temp": 21.799999237060547}
{"hum": 44.900001525878906, "wkt": "2018-04-12 17:50:12.217587", "temp": 21.7999
99237060547}
{"hum": 44.900001525878906, "wkt": "2018-04-12 17:50:17.750149", "temp": 21.7999
99237060547}
{"hum": 44.900001525878906, "wkt": "2018-04-12 17:50:25.809461", "temp": 21.7999
99237060547}
{"hum": 44.900001525878906, "wkt": "2018-04-12 17:50:31.341986", "temp": 21.7999
99237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:50:36.874441", "temp": 21.799999237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:50:42.406256", "temp": 21.799999237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:50:47.938762", "temp": 21.799999237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:50:53.471334", "temp": 21.799999237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:50:59.003760", "temp": 21.799999237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:51:04.536189", "temp": 21.799999237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:51:10.068669", "temp": 21.799999237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:51:15.601198", "temp": 21.799999237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:51:21.130329", "temp": 21.799999237060547}
{"hum": 45.0, "wkt": "2018-04-12 17:51:26.662908", "temp": 21.799999237060547}
```

Gambar 8. Komposisi data yang dikirimkan oleh Raspberry Pi ke kafka cluster

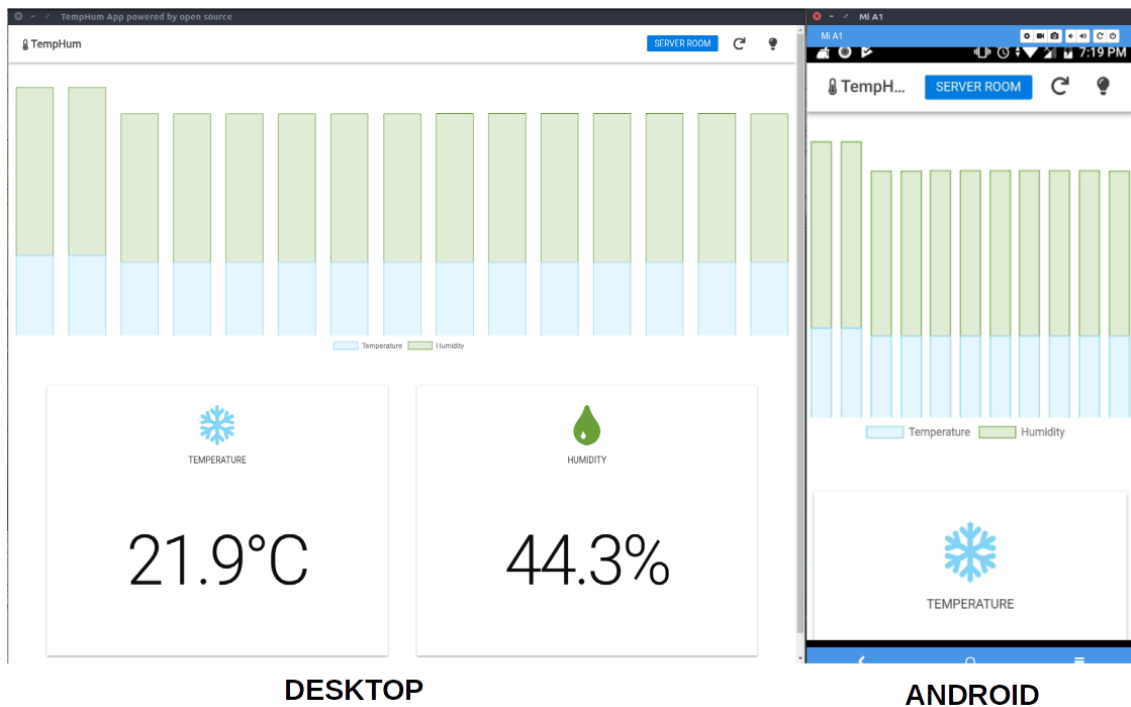
Data yang dikirim Raspberry kedalam kafka cluster menggunakan topik ‘temphum’ dengan konfigurasi satu partisi dan satu replication factor seperti yang ditunjukkan dalam Gambar 9.

```
Topic:temphum PartitionCount:1 ReplicationFactor:1 Configs:
2 Topic: temphum Partition: 0 Leader: 1002 Replicas: 1002 Isr: 100
bash-4.4#
```

Gambar 9. Konfigurasi topik dalam kafka untuk temphum

Data sensor yang diterima oleh kafka consumer (servis python) harus diolah sehingga memenuhi persyaratan streaming data sesuai referensi [9]. Data yang sudah memenuhi persyaratan tersebut diteruskan secara terus-menerus ke pengguna aplikasi.

Dari sisi pengguna (Frontend), data diterima dengan menggunakan antarmuka EventSource di Javascript. Dengan teknologi Javascript modern saat ini, mudah bagi pengembang untuk merilis aplikasi ke banyak media atau cross platform dengan bantuan teknologi PhoneGap atau Cordova. Berikut ini adalah aplikasi TempHum yang sudah dirilis untuk versi desktop dan smartphone seperti Gambar 10.



Gambar 10. Rilis aplikasi TempHum untuk versi desktop dan android

IV. KESIMPULAN

Aplikasi TempHum dikembangkan untuk memenuhi kebutuhan operator atau pengguna lainnya dalam memantau kondisi udara terkini dalam ruang server sehingga dapat menjaga kondisi ruang server sesuai standar [1]. Aplikasi ini memudahkan pengguna dengan fitur streaming data temperatur dan kelembapan untuk kondisi udara terkini secara dinamis. Tidak hanya itu, aplikasi TempHum dapat digunakan di desktop dan smartphone sehingga pengguna dapat lebih mudah memantau kondisi udara ruang server kapan saja dan dimana saja.

DAFTAR PUSTAKA

- [1] Apache Kafka Documentation. (2018). GETTING STARTED WITH KAFKA. Diakses dari <https://kafka.apache.org/documentation/#gettingStarted>. tanggal 11 April 2018.
- [2] Apache Kafka Documentation. (2018). HIGH AVAILABILITY AND DURABILITY GUARANTEES. Diakses dari https://kafka.apache.org/documentation/#design_ha. tanggal 11 April 2018.
- [3] MDN Web Documentation. (2018). USING SERVER SENT EVENTS. Diakses dari https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events/Using_server-sent_events. tanggal 12 April 2018.
- [4] Python Gunicorn Documentation. (2018). GUNICORN SETTINGS ON WORKER CLASS. Diakses dari <http://docs.gunicorn.org/en/stable/settings.html#worker-class>. tanggal 12 April 2018.
- [5] Raspberry Pi Product. (2018). RASPBERRY PI SPECIFICATION. Diakses dari <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>. tanggal 11 April 2018.
- [6] Stack Overflow Q&A. (2012). SERVER SENT EVENTS VS POLLING. Diakses dari <https://stackoverflow.com/questions/9397528/server-sent-events-vs-polling>. tanggal 12 April 2018.
- [7] The Raspberry Pi Zero. (2016). THE RASPBERRY PI ZERO AS A DEDICATED VPN. Diakses dari <http://adamantine.me/2016/08/02/the-raspberry-pi-zero-as-a-dedicated-vpn/>. tanggal 11 April 2018.
- [8] TempHum Application Deployment. (2018). TempHum ON DESKTOP. Diakses dari <http://aerobox.bbta3.bppt.go.id/index.php/s/QLu5b1xeRT0TBNJ>. tanggal 11 April 2018.
- [9] TempHum Application Deployment. (2018). TempHum ON SMARTPHONE. Diakses dari <https://play.google.com/store/apps/details?id=com.wordpress.arizalakbarzikri.temphum>. tanggal 11 April 2018.
- [10] Telecommunications Infrastructure Standard for Data Centers. (2005). TELECOMMUNICATIONS INDUSTRY ASSOCIATION. Diakses dari <https://manuais.iessanclemente.net/images/9/9f/Tia942.pdf>. tanggal 11 April 2018.