

Microservice Based Software Architecture on Canteen Information Management System

[Arsitektur Perangkat Lunak Berbasis Layanan Mikro Pada Sistem Manajemen Informasi Kantin]

Mukhammad Dayu Anwar ^{*.1)}, Irwan A. Kautsar ²⁾

¹⁾Program Studi Teknik Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

²⁾Program Studi Teknik Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

*Email Penulis Korespondensi: irwan@umsida.ac.id

Abstract. *Microservices is one way to maximize the performance of the data service that will be used to create the system. In addition, it is easy to develop, because each service is structured and separate for data that will be consumed by other developers. The case used in this research is the E-Canteen Application, where the circulation of data from partners, customers and transactions that will be consumed by users must run smoothly. By implementing microservices in this E-Canteen Application, the circulation of data, the speed of the data generated will be very helpful to maximize each service used. The results of testing the microservices that have been created, show that the circulation of data and services that have been used become very efficient and stable when there are many requests or accesses from users. With these results, the effect of microservices for data circulation used becomes very efficient and stable.*

Keywords – *Microservices, E-Kantin, API*

Abstrak. *Microservices menjadi salah satu cara untuk memaksimalkan performa dari service data yang akan digunakan untuk membuat sistem. Selain itu mudah untuk dikembangkan, karena setiap service nya sudah terstruktur dan terpisah untuk data yang akan dikonsumsi oleh developer lain. Kasus yang digunakan dalam penelitian ini adalah aplikasi E-Kantin, yang dimana sirkulasi data dari mitra, customer dan transaksi yang akan dikonsumsi oleh user harus berjalan dengan lancar. Dengan diimplementasikan nya microservices pada Aplikasi E-Kantin ini, maka sirkulasi data, kecepatan data yang dihasilkan akan sangat membantu untuk memaksimalkan setiap service yang dipakai. Hasil dari pengujian microservices yang telah dibuat, menunjukkan bahwa sirkulasi data dan service yang telah digunakan menjadi sangat efisien dan stabil pada saat banyaknya request atau akses dari user pemakainya. Dengan hasil tersebut, maka pengaruh dari microservices untuk sirkulasi data yang digunakan menjadi sangat efisien dan stabil.*

Kata Kunci - *Microservices, E-Kantin, API*

I. PENDAHULUAN

Pada perkembangan teknologi saat ini sebuah Arsitektur Microservice sangat dibutuhkan untuk memudahkan pengembangan sebuah aplikasi / sistem. Pada Microservice komunikasi antar setiap service atau layanan nya menggunakan API (Application programming Interface), karena cocok sebagai perantara service yang saling berhubungan. API sendiri adalah sebuah perantara yang bisa digunakan untuk menyatukan banyak service atau menghubungkan antara tampilan dan data di sebuah sistem. Keunggulan dari Microservices adalah leluasa saat ada pengembangan sistem yang rumit, data saat terdistribusi menjadi lebih lancar, akses untuk setiap service nya lebih mudah dan terstruktur.

Kasus yang digunakan untuk implementasi microservice ini adalah aplikasi E-Kantin. Yang dimana untuk pengembangan fiturnya akan memudahkan developer, karena servicenya sudah terstruktur sesuai dengan yang di rencanakan. Jika dibandingkan dengan struktur monolith yang dimana tampilan, logic dan data acces / querynya dijadikan 1 pada halaman tersebut, akan mempersulit pengembangan pada sistem. Selain itu jika fitur yang akan dikembangkan memiliki alur yang sangat rumit akan mempersulit developer untuk menambah atau mengubahnya, karena harus tau bagaimana tampilan, logic dan query pada fitur tersebut agar tidak merusak alur yang sudah berjalan.

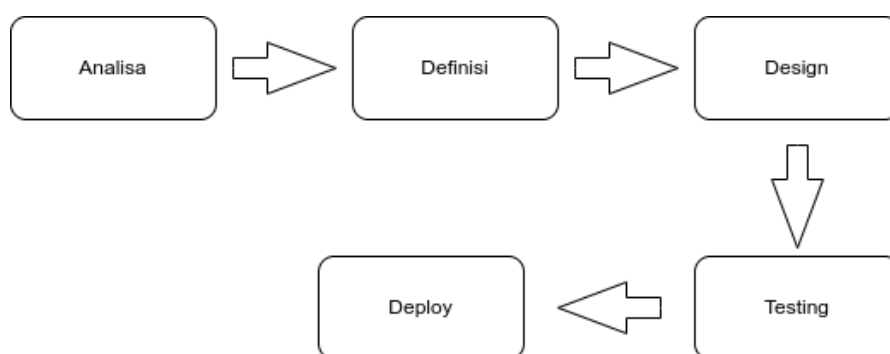
Untuk mengatasi masalah ini, peneliti menerapkan sebuah Architecture Microservices agar pengembangan pada sistem menjadi mudah dan data yang dikonsumsi akan menjadi lebih lancar. Dengan menggunakan Microservice ini diharapkan agar data yang dikonsumsi akan semakin maksimal performanya, karena setiap layanan datanya terstruktur dan terdistribusi dengan baik.

Dengan adanya permasalahan di atas maka peneliti membuat solusi dengan judul “Arsitektur Perangkat Lunak Berbasis Layanan Mikro Pada Sistem Manajemen Informasi Kantin”. Agar distribusi data dari aplikasi tersebut bisa

lancar dan pengembangannya menjadi lebih mudah maka, Architecture Microservices harus diterapkan untuk konsumsi data setiap servicenya.

II. METODE

Untuk pengujian sistem, digunakan metode load-testing. Load testing adalah proses menguji kinerja suatu aplikasi atau sistem dengan memberikan beban yang tinggi pada infra-struktur dan mengamati bagaimana sistem tersebut merespons di bawah beban tersebut. Pengujian beban dilakukan pada sebuah sistem (baik prototipe atau sistem yang berfungsi penuh) daripada desain atau model arsitektur[20]. Penulis menggunakan Locust, salah satu library python yang mempermudah developer untuk melakukan uji load-test pada suatu sistem dengan melakukan simulasi swarm spawning.



Gambar 1. Metode Agile

Pada gambar 1 diatas, menjelaskan tentang metode yang digunakan yaitu metode agile. Ditahap pertama adalah analisa, yang dimana menggunakan referensi jurnal terdahulu. Tahap kedua definisi, menjelaskan tentang teknologi, bahasa pemrograman dan framework yang digunakan. Tahap selanjutnya desain dari microservice yang akan dibuat, setelah desain dibuat selanjutnya adalah testing. Yang dimana menggunakan aplikasi yang bernama postman. Tahap terakhir adalah deploying yang dimana microservice akan disebar dan diakses oleh frond end developer.

2.1 Analisa

Pada penelitian ini, penelitian terdahulu dijadikan sebagai masukan dan referensi untuk membantu penelitian sehingga dapat menambah teori dan wawasan. Jurnal penelitian yang digunakan untuk referensi adalah penelitian [1] menjelaskan proses migrasi sistem alat pendukung rapid prototyping dari monolit ke arsitektur microservice yang akan digunakan sebagai implementasi Project Based Learning. Penelitian yang dilakukan oleh [2] yang berjudul “Arsitektur Microservice untuk Resiliensi Sistem Informasi”. Didalam penelitian tersebut dijelaskan bahwa dengan menggunakan Microservices menunjukkan peningkatan kualitas pada aspek resiliensi, misalkan ketika beberapa service mengalami gangguan, sistem dapat tetap berjalan sebagaimana mestinya. Penelitian yang dilakukan oleh [3] yang berjudul “Implementasi Arsitektur Microservices pada Rancang Bangun Aplikasi Marketplace Berbasis Web”. Aplikasi tersebut menggunakan arsitektur microservices telah berhasil dibangun dengan hasil persentase pengujian black box 100% dari 25 aktivitas berhasil dan hasil persentase pengujian endpoint api 100% dari 29 aktivitas berhasil.

2.2 Definisi

Paragraf ini akan menjelaskan tentang definisi apa saja yang digunakan untuk membuat sebuah microservices.

2.2.1 Microservices

Microservices adalah kumpulan layanan yang saling berkaitan untuk membangun sebuah sistem yang terstruktur, agar sistem tersebut berjalan dengan maksimal. Komunikasi antar setiap service atau layanan nya menggunakan API (Application programming Interface), karena cocok sebagai perantara service yang saling berhubungan. Microservices berguna untuk memaksimalkan layanan dari setiap data yang digunakan dan mempermudah developer mengembangkan sistem tersebut.

2.2.2 Python

Python merupakan salah satu bahasa pemrograman yang banyak digunakan oleh perusahaan besar maupun para developer untuk mengembangkan berbagai macam aplikasi berbasis desktop, web dan mobile [4]. Bahasa pemrograman Python menggunakan struktur orientasi objek yang bertujuan untuk memudahkan seorang programmer menuliskan kode yang jelas, bagus dan logis untuk project skala kecil maupun besar.

2.2.3 Framework Flask

Framework Flask merupakan micro web framework yang menggunakan bahasa python [5]. Micro framework ini berguna untuk mempercepat pembangunan suatu aplikasi karena sudah ada struktur, library dan komponen lain nya untuk menunjang developer tanpa harus membuat aplikasi dari awal atau nol.

2.2.4 API

API merupakan sebuah sistem penyedia layanan yang berjalan pada sisi server [6][7]. Application Programming Interface (API) juga dapat diartikan sebagai perantara atau penghubung antara service database dengan interface aplikasi. API memudahkan developer untuk membangun aplikasi karena service dari data dipisah dan tidak dijadikan satu dengan aplikasi atau monolith, sehingga developer akan mudah untuk mengembangkan aplikasi tersebut meskipun banyak fitur yang sudah berjalan.

2.2.5 Firebase

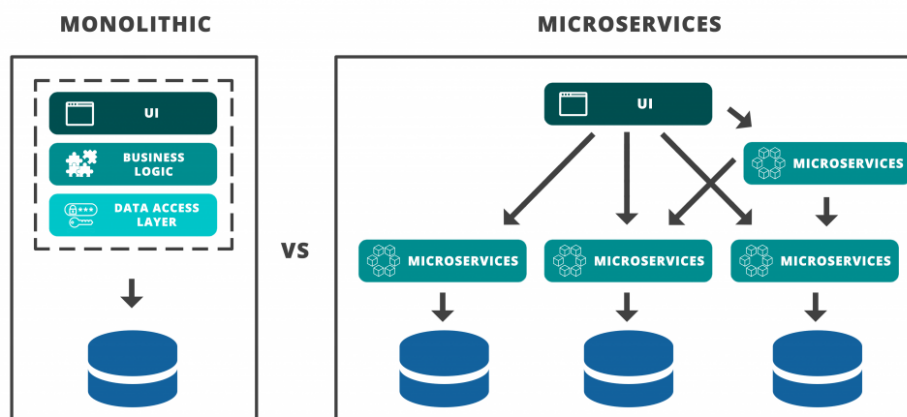
Firebase Database merupakan penyimpanan basis data nonSQL yang memungkinkan untuk menyimpan beberapa tipe data [8]. Firebase memudahkan developer untuk menyimpan data karena berbasis BaaS (Backend as a Service), Sehingga dalam mengembangkan aplikasi developer tidak memberikan effort yang besar untuk urusan backend.

2.2.6 Postman

Postman adalah sebuah aplikasi yang digunakan untuk menguji API yang telah dibuat oleh developer backend untuk menampilkan data yang dibutuhkan. Cara penggunaan dari postman sendiri yaitu request ke url yang akan kita tuju dan isi semua parameter yang sudah dibuat oleh developer, agar data yang muncul sesuai dengan yang anda inginkan.

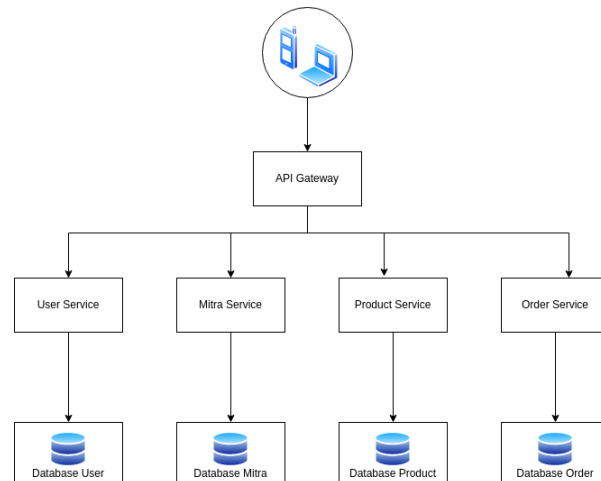
2.3 Design

Setelah definisi dari microservices, tahap selanjutnya adalah desain untuk microservices. Pada gambar 2 dibawah menjelaskan antara monolit dan microservices yang berjalan, monolit merupakan model yang dibangun dalam satu codebase yang sama atau ui dan data nya dibuat pada halaman yang sama. Sedangkan microservices dibangun secara terpisah dari ui atau tampilannya, dan berbagai services nya juga dibuat terpisah agar memaksimalkan performa dari request data yang akan dipanggil atau dipakai.



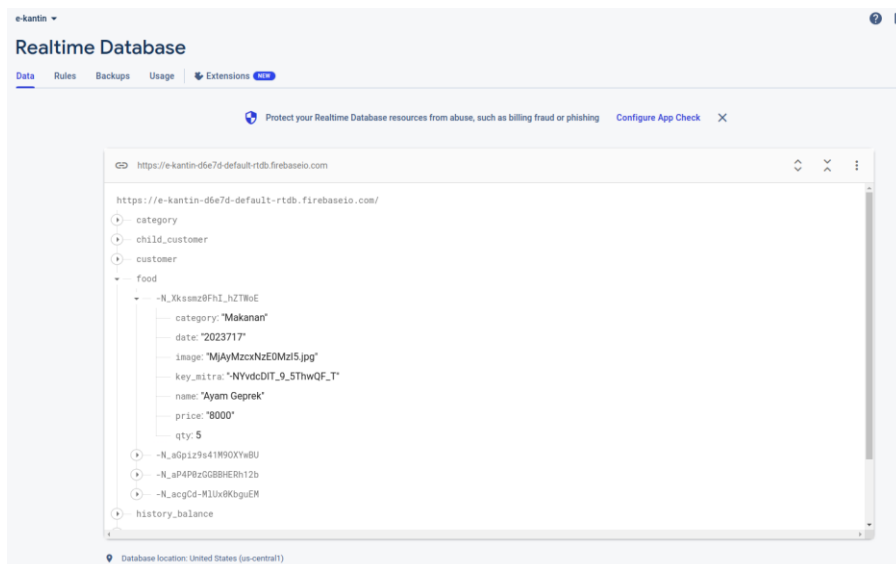
Gambar 2. Monolith dan Microservices

Pada gambar 3 dibawah adalah alur dari setiap services yang akan diakses atau dikonsumsi oleh front end developer dan service akan mengakses setiap database yang sudah ditentukan.



Gambar 3. Alur Service Database

Selanjutnya adalah gambar 4, tampilan database firebase yang digunakan. Pada firebase tampilan data yang masuk akan disimpan seperti JSON dan disinkronkan secara realtime ke setiap user penggunaanya, yang memudahkan mengelola suatu data yang berskala besar.



Gambar 4. Firebase

2.4 Testing

Untuk tahap testing API menggunakan aplikasi postman, karena untuk memastikan seluruh services yang dibuat telah berjalan sesuai yang diharapkan. Pada tahap ini sangat penting mengevaluasi dan menyesuaikan jika ada salah satu services yang tidak sesuai dengan apa yang telah disepakati.

2.5 Deploy

Deploy adalah menyebarkan sistem yang sudah dibuat agar bisa diakses banyak orang. Pada gambar 5 dibawah adalah deploy menggunakan aplikasi ngrok. Aplikasi ini cocok untuk membuat sebuah sistem agar bisa diakses secara online dan gratis, karena dapat membuka jaringan private melalui firewall dan menghubungkan localhost ke internet dengan tunnel yang aman.

```

Session Status      online
Account             Mukhammad Dayu Anwar (Plan: Free)
Update             update available (version 3.3.2, Ctrl-U to update)
Version            3.3.1
Region             Asia Pacific (ap)
Latency            -
Web Interface       http://127.0.0.1:4040
Forwarding          https://55ef-202-67-40-252.ngrok-free.app -> http:

Connections
  ttl    opn    rt1    rt5    p50    p90
  0      0      0.00  0.00  0.00  0.00

```

Gambar 5. Run Ngrok

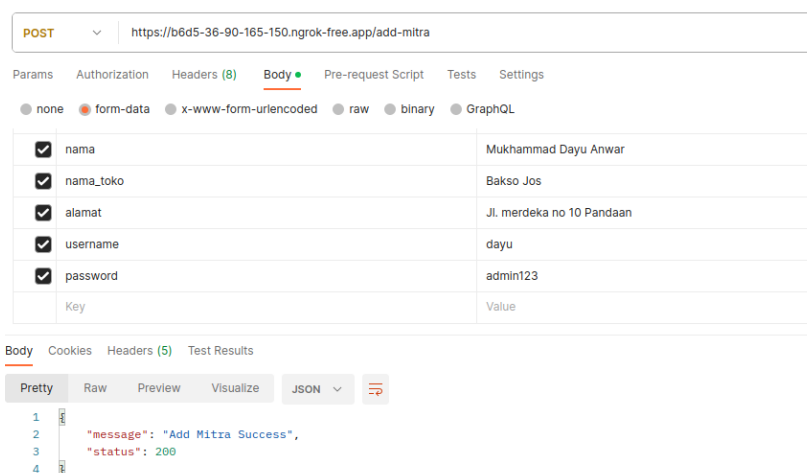
III. HASIL DAN PEMBAHASAN

A. Implementasi

Pada bab ini akan menjelaskan mengenai hasil dan pembahasan dari API yang telah dibuat berdasarkan analisa yang telah dijelaskan diatas. Format data yang akan ditampilkan pada API dibawah adalah json, yang dimana format data ini kompatible dengan banyak bahasa pemrograman, environment dan library. Berikut ini adalah tampilan dari API yang telah dibuat dan diakses menggunakan postman.

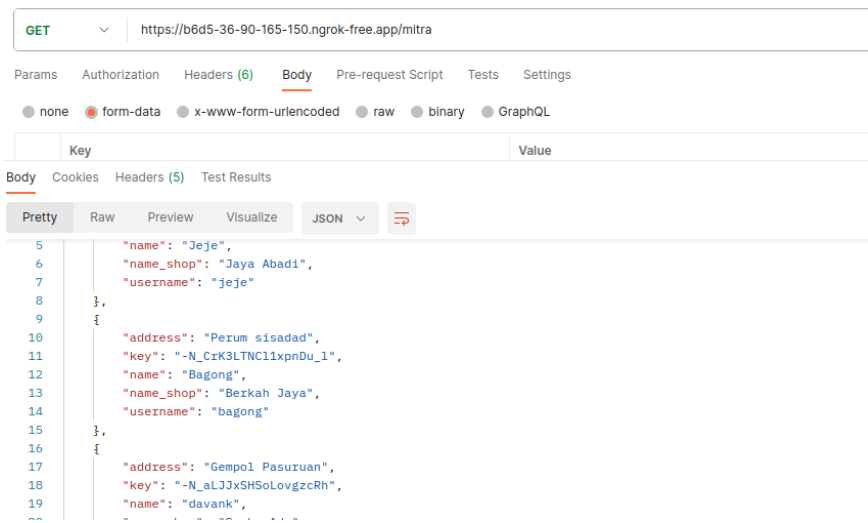
3.1.1 API Mitra

Pada gambar 6 dibawah ini, menjelaskan bagaimana cara untuk menambahkan mitra. Pada tulisan post itu adalah method yang akan direquest, disamping nya itu adalah url yang harus diisi untuk menuju ke service tambah mitra. Untuk body dan form data ialah parameter yang harus dibawa untuk menambahkan mitra, setelah itu akan mengembalikan sebuah json data yang berisi “berhasil menambahkan mitra”.



Gambar 6. API request untuk menambah data mitra

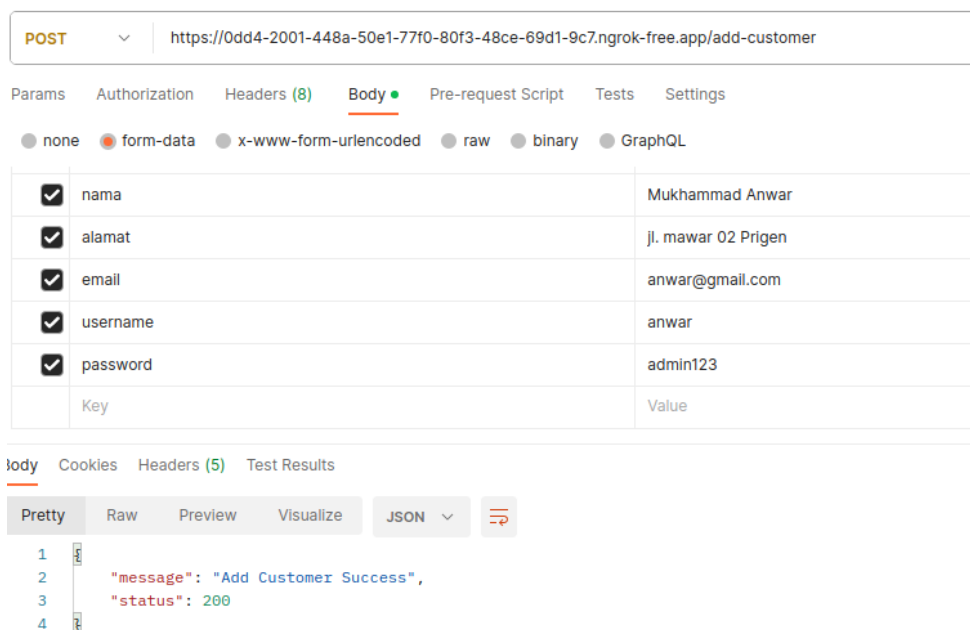
Pada gambar 7 dibawah ini, menjelaskan bagaimana cara untuk menampilkan data semua mitra. Pada tulisan get itu adalah method yang akan direquest, disamping nya itu adalah url yang harus diisi untuk menuju ke service list mitra. Setelah itu akan mengembalikan sebuah json data yang berisi semua data mitra.



Gambar 7. API list semua data mitra

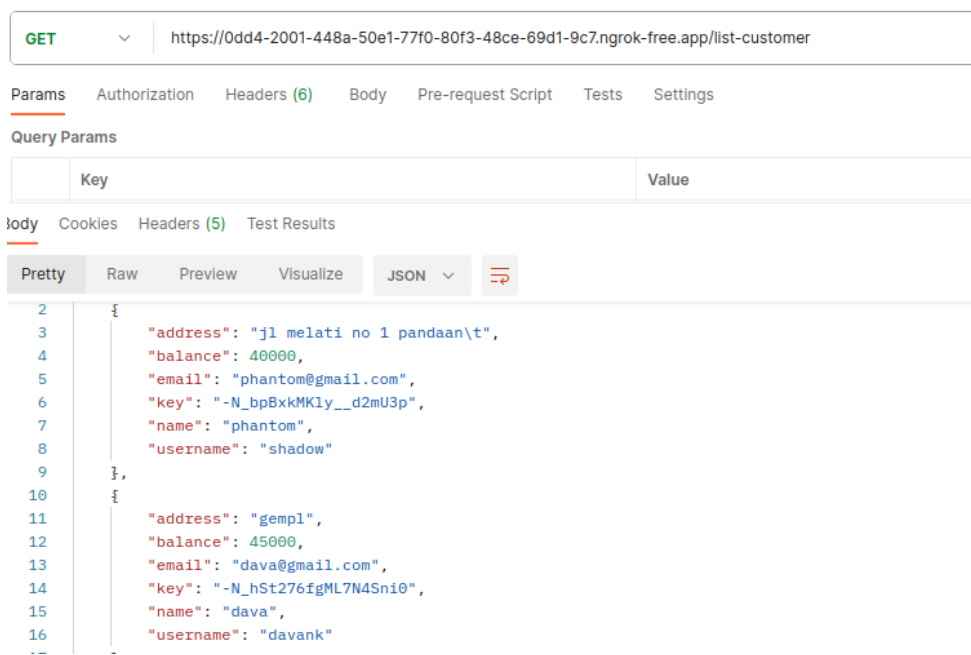
3.1.2 API Customer

Pada gambar 8 dibawah ini, menjelaskan bagaimana cara untuk menambahkan customer. Pada tulisan post itu adalah method yang akan direquest, disamping nya itu adalah url yang harus diisi untuk menuju ke service tambah customer. Untuk body dan form data ialah parameter yang harus dibawa untuk menambahkan customer, setelah itu akan mengembalikan sebuah json data yang berisi “berhasil menambahkan customer”.



Gambar 8. API request untuk menambahkan data customer

Pada gambar 9 dibawah ini, menjelaskan bagaimana cara untuk menampilkan data semua customer. Pada tulisan get itu adalah method yang akan direquest, disamping nya itu adalah url yang harus diisi untuk menuju ke service list customer. Setelah itu akan mengembalikan sebuah json data yang berisi semua data customer.



GET https://0dd4-2001-448a-50e1-77f0-80f3-48ce-69d1-9c7.ngrok-free.app/list-customer

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
-----	-------

body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

```

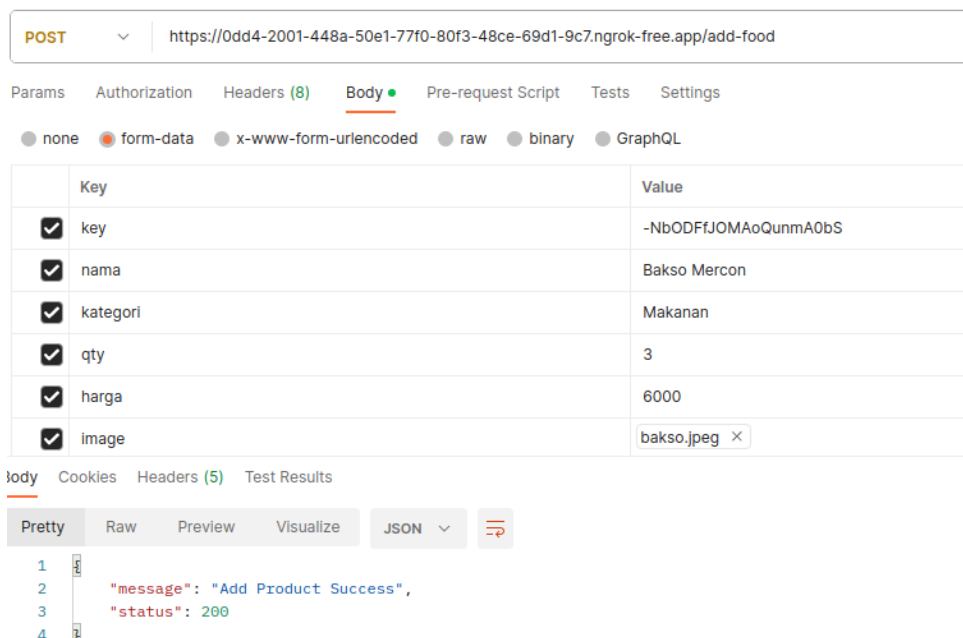
2  {
3    "address": "jl melati no 1 pandaan\t",
4    "balance": 40000,
5    "email": "phantom@gmail.com",
6    "key": "-N_bpBxkMKly__d2mU3p",
7    "name": "phantom",
8    "username": "shadow"
9  },
10 {
11  "address": "gempl",
12  "balance": 45000,
13  "email": "dava@gmail.com",
14  "key": "-N_hSt276fgML7N4Sni0",
15  "name": "dava",
16  "username": "davank"
--

```

Gambar 9. API untuk list semua data customer

3.1.3 API Produk

Pada gambar 10 dibawah ini, menjelaskan bagaimana cara untuk menambahkan produk. Pada tulisan post itu adalah method yang akan direquest, disamping nya itu adalah url yang harus diisi untuk menuju ke service tambah produk. Untuk body dan form data ialah parameter yang harus dibawa untuk menambahkan produk, setelah itu akan mengembalikan sebuah json data yang berisi “berhasil menambahkan produk”.



POST https://0dd4-2001-448a-50e1-77f0-80f3-48ce-69d1-9c7.ngrok-free.app/add-food

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value
<input checked="" type="checkbox"/> key	-NbODFfJOMaOqunMA0bS
<input checked="" type="checkbox"/> nama	Bakso Mercon
<input checked="" type="checkbox"/> kategori	Makanan
<input checked="" type="checkbox"/> qty	3
<input checked="" type="checkbox"/> harga	6000
<input checked="" type="checkbox"/> image	bakso.jpeg ×

body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize JSON

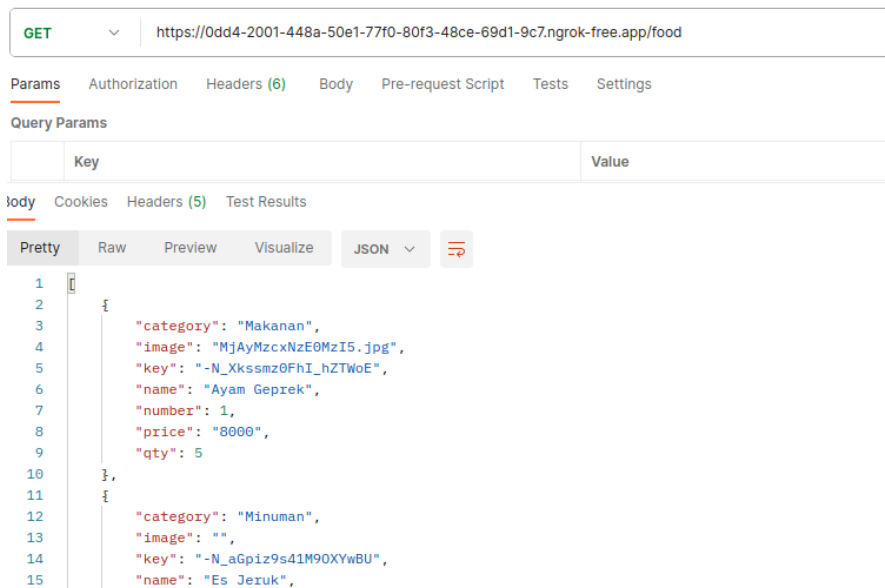
```

1  {
2    "message": "Add Product Success",
3    "status": 200
4  }

```

Gambar 10. API request untuk menambah data produk

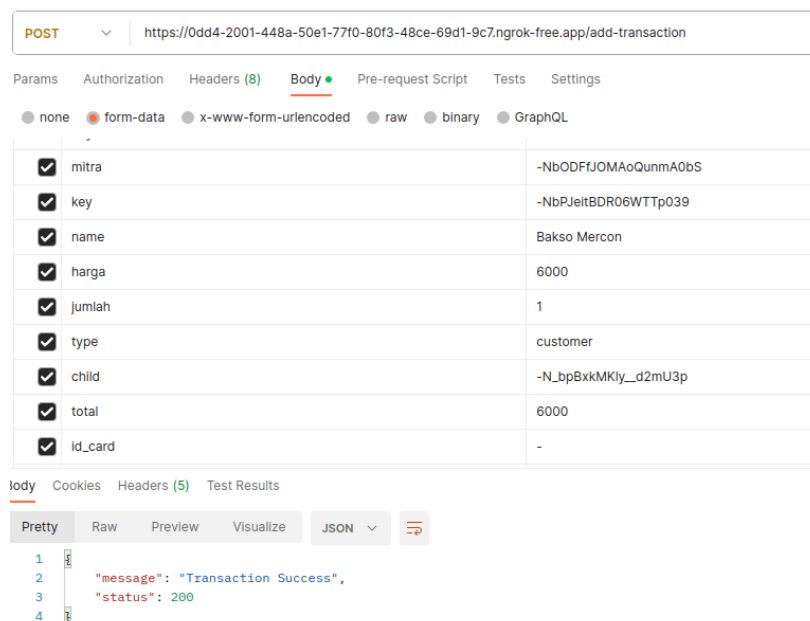
Pada gambar 11 dibawah ini, menjelaskan bagaimana cara untuk menampilkan data semua produk. Pada tulisan get itu adalah method yang akan direquest, disamping nya itu adalah url yang harus diisi untuk menuju ke service list produk. Setelah itu akan mengembalikan sebuah json data yang berisi semua data produk.



Gambar 11. API list semua data produk

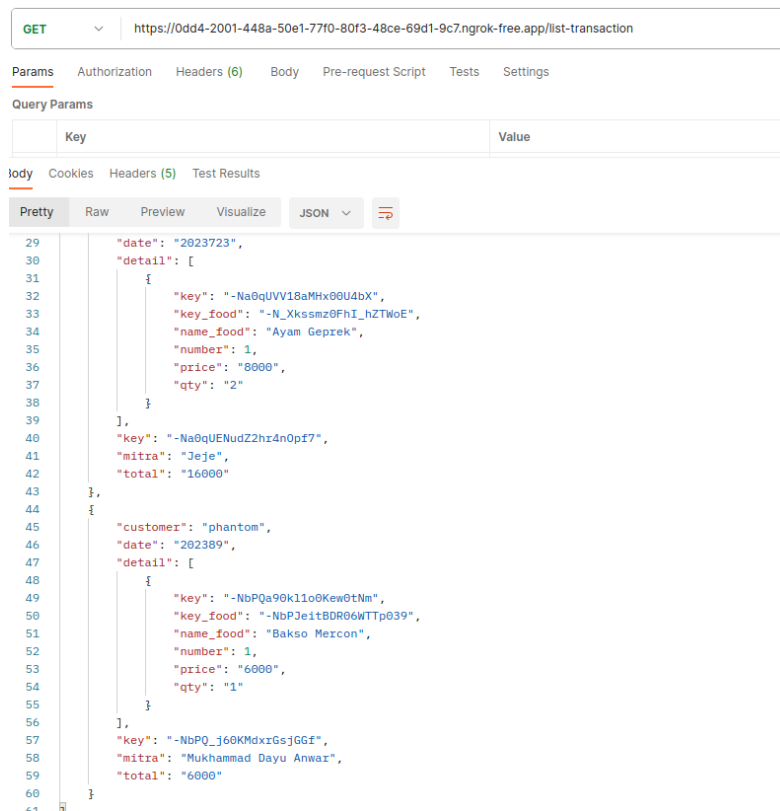
3.1.4 API Transaksi

Pada gambar 12 dibawah ini, menjelaskan bagaimana cara untuk melakukan transaksi pembelian produk. Pada tulisan post itu adalah method yang akan direquest, disamping nya itu adalah url yang harus diisi untuk menuju ke service transaksi pembelian produk. Untuk body dan form data ialah parameter yang harus dibawa untuk bisa melakukan transaksi pembelian produk, setelah itu akan mengembalikan sebuah json data yang berisi “transaksi berhasil”.



Gambar 12. API request transaksi pembelian atas customer

Pada gambar 13 dibawah ini, menjelaskan bagaimana cara untuk menampilkan data semua transaksi yang berhasil masuk. Pada tulisan get itu adalah method yang akan direquest, disamping nya itu adalah url yang harus diisi untuk menuju ke service transaksi yang berhasil masuk. Setelah itu akan mengembalikan sebuah json data yang berisi semua data transaksi yang berhasil masuk.



```

GET https://Odd4-2001-448a-50e1-77f0-80f3-48ce-69d1-9c7.ngrok-free.app/list-transaction

Params Authorization Headers (6) Body Pre-request Script Tests Settings
Query Params
Key Value
body Cookies Headers (5) Test Results
Pretty Raw Preview Visualize JSON
29 "date": "2023723",
30 "detail": [
31   {
32     "key": "-Na0quVV18aMhx004bX",
33     "key_food": "-N_Xkssmz0FhI_hZTW0E",
34     "name_food": "Ayam Geprek",
35     "number": 1,
36     "price": "8000",
37     "qty": "2"
38   }
39 ],
40 "key": "-Na0quENudZ2hr4n0pf7",
41 "mitra": "Jeje",
42 "total": "16000"
43 },
44 {
45   "customer": "phantom",
46   "date": "202309",
47   "detail": [
48     {
49       "key": "-NbPQa90k11o0Kew0tNm",
50       "key_food": "-NbPJeitBDR06WTTp039",
51       "name_food": "Bakso Mercon",
52       "number": 1,
53       "price": "6000",
54       "qty": "1"
55     }
56   ],
57   "key": "-NbPQ_j60KMDxrGsJGGf",
58   "mitra": "Mukhammad Dayu Anwar",
59   "total": "6000"
60 }
61 ]

```

Gambar 13. API request list semua data transaksi yang berhasil masuk

3.2 Pengujian Services

Dari pengujian API menggunakan postman diatas, diketahui pada tabel 1 dibawah ini menunjukkan kalau setiap services nya berjalan dengan lancar dan sesuai dengan kebutuhan yang sudah direncanakan. Dan juga pada tabel 1 dibawah ini akan menjelaskan bahwa pengujian terhadap endpoint yang ada pada setiap services nya telah berhasil dibuat dan dijalankan tanpa ada masalah.

Tabel 1. Pengujian endpoint / services API menggunakan Postman

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
1.	API Tambah Mitra	Menampilkan data json yang berhasil	[✓] Berhasil [] Tidak Berhasil
2.	API List Mitra	Menampilkan list mitra yang sudah dimasukkan	[✓] Berhasil [] Tidak Berhasil
3.	API Tambah Customer	Menampilkan data json yang berhasil	[✓] Berhasil [] Tidak Berhasil
5.	API List Customer	Menampilkan list customer yang sudah dimasukkan	[✓] Berhasil [] Tidak Berhasil

6.	API Tambah Produk	Menampilkan data json yang berhasil	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
7.	API List Produk	Menampilkan list produk yang sudah dimasukkan	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
8.	API Transaksi	Menampilkan data json yang berhasil	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
9.	API List Transaksi Berhasil	Menampilkan list transaksi yang berhasil	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil

IV. SIMPULAN

Berdasarkan penelitian diatas disimpulkan bahwa microservices yang telah dibuat dapat membantu memaksimalkan performa data yang akan dikonsumsi oleh user pengguna. Selain itu untuk pengembangan dengan mengikuti bahasa pemrograman / framework yang terbaru dan terpopuler akan sangat mudah, karena struktur dan setiap servicesnya mudah untuk diubah dan ditambah. Dari sisi Frontend / Tampilan pun mudah dikonsumsi dan meskipun services nya berubah tidak harus mengubah semua tampilan yang ada, hanya menyesuaikan services yang diubah saja.

REFERENSI

- [1] Irwan Alnarus Kautsar, M. Ruslianor Maika, Agoes Nur Budiman, Arik Bagus Setyawan, Jagad Yudha Awali, "Microservice Based Architecture: The Development of Rapid Prototyping Supportive Tools for Project Based Learning," 22963954, April 2023, 10.1109/EDUNINE57531.2023.10102884.
- [2] Hatma Suryotrisongko, "Arsitektur Microservice untuk Resiliensi Sistem Informasi," Jurnal Sisfo., vol. 06, no. 02, pp. 235–250, Jan. 2017.
- [3] Alessandro Sinambela, Ernawati, Funny Farady Coastera, "Implementasi Arsitektur Microservices pada Rancang Bangun Aplikasi Marketplace Berbasis Web," Jurnal Rekursif., vol. 9, no. 1, Mar. 2021.
- [4] Schuerer, Katja dan Corinne Maufrais (2010). "Introduction to Programming using Python," Boston: Pearson, hal. 1–242. ISBN: 0132747189.
- [5] D. Ghimire, "Comparative study on Python web frameworks: Flask and Django," no. May, pp. 1–40, 2020, [Online]. Available: <http://www.theseus.fi/handle/10024/339796>.
- [6] Xu, Hao, Hong Wang, and Suo Zhang. "Application programming interface (api) service apparatus and application programming interface (api) service system." U.S. Patent Application No. 16/181,927.
- [7] Rulloh, Amin, Dewi Erla Mahmudah, and Herman Kabetta. "Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android." Teknikom: Teknologi Informasi, Ilmu Komputer dan Manajemen 1.2 (2017): 8589.
- [8] Luffi Aditya Sandy, Rizky Januar, dan Ridho Rahman Hariadi, "Rancang Bangun Aplikasi Chat pada Platform Android dengan Media Input berupa Canvas dan Shareable Canvas untuk Bekerja Dalam Satu Canvas secara Online," JURNAL TEKNIK ITS., vol. 6, no. 2, 2337-3520, 2017.

Conflict of Interest Statement:

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.