

Perancangan Aplikasi *E-ticketing* Dengan Model Arsitektur *Microservice* Menggunakan Kafka

Awal Pradinata^{a,1,*}, Poetri Lestari Lokapitasari B^{a,2} dan Huzain Azis^{a,3}

^a Universitas Muslim Indonesia, Jl. Urip Sumoharjo Km. 5, Kota Makassar, 90231, Indonesia
¹ 13020180034@umi.ac.id; ² poetrilestari@umi.ac.id; ³ huzain.azis@umi.ac.id

INFORMASI ARTIKEL	ABSTRAK
Diterima : 17 – 07 – 2023 Direvisi : 26 – 08 – 2023 Diterbitkan : 31 – 08 – 2023	Arsitektur <i>microservice</i> memecah sistem yang kompleks dan besar menjadi serangkaian layanan kecil dan mandiri. Salah satu pola arsitektur yang umum digunakan adalah pendekatan <i>event driven</i> , yang memungkinkan komunikasi berbasis <i>event</i> antar layanan. Namun, pendekatan ini juga membawa risiko kehilangan data, yang dapat diatasi dengan pola orkestrasi menggunakan Apache Kafka sebagai <i>message broker</i> . Kafka menyediakan platform yang cocok untuk komunikasi <i>event driven</i> dengan kemampuannya dalam menyimpan, menerima, dan mengirim pesan secara asinkron. Penelitian ini bertujuan membangun aplikasi <i>e-ticketing</i> berbasis web dengan menggunakan arsitektur <i>microservice</i> dan Kafka. Hasilnya adalah sebuah aplikasi <i>e-ticketing</i> yang menggunakan Kafka untuk komunikasi antar layanan, dengan implementasi lima topik untuk proses transaksi antar <i>gate-ticketing-service</i> dan <i>gate-acl-service</i> secara asinkron menggunakan Kafka sebagai media pengiriman <i>event</i> .
Kata Kunci: Aplikasi <i>E-ticketing</i> Microservice Kafka	

This is an open access article under the [CC-BY-SA](#) license



I. 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. Saat ini salah satu yang paling berpengaruh dalam perkembangan dunia teknologi adalah perangkat lunak komputer yang dikemas dalam bentuk layanan jasa sehingga mempermudah manusia. Layanan jasa dalam bentuk perangkat lunak saat ini sangat masif perkembangannya di Indonesia sehingga membutuhkan persiapan yang baik terhadap teknologi yang digunakan. Seiring dengan perkembangan teknologi yang sangat pesat, arsitektur *microservice* muncul menjadi sebuah tren yang digunakan oleh pengembang aplikasi dalam beberapa tahun terakhir, hal ini terjadi karena dengan menggunakan arsitektur *microservice* akan memudahkan sistem untuk di-maintain dan di-test, loosely coupled, di-deploy independen, terorganisir berdasarkan kapabilitas bisnis, dan dapat dimiliki oleh tim kecil karena *Microservice* architecture membagi sistem kompleks dan besar menjadi *service – service* kecil dan sederhana yang dibangun secara terpisah[1].

Untuk membangun arsitektur *microservice* terdapat beberapa pola arsitektur yang digunakan salah satunya adalah *event driven*. *Event driven* merupakan sebuah konsep dalam pengembangan *microservice* yang bersifat komunikasi berdasarkan *event* sehingga terjadinya proses kehilangan data lebih besar [2], untuk itu digunakanlah salah satu pattern untuk menangani kesalahan tersebut yaitu pattern orchestration dengan Apache Kafka sebagai message broker atau media komunikasi antar *service* yang sangat sesuai dengan konsep *event driven* karena apache kafka dapat menyimpan, menerima dan mengirim data atau pesan, dengan cara kerja apache kafka yaitu 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 [3].

Penelitian [2] Implementasi teknologi *event-driven microservice* dengan choreography pattern telah berhasil diterapkan pada aplikasi penerimaan peserta didik baru dengan *service* yang telah dibuat yaitu antara lain API Gateway, Auth Service, Registration Service, Registrant Service, Announcement Service, Institution Service, Payment Service, Mail Service, dan Redis instance. Selanjutnya pengujian menggunakan Software Architecture Analysis Method (SAAM) antara lain menentukan skenario, menentukan kandidat arsitektur, mengklasifikasi skenario, melakukan evaluasi skenario, mengamati interaksi skenario, hingga memiliki hasil

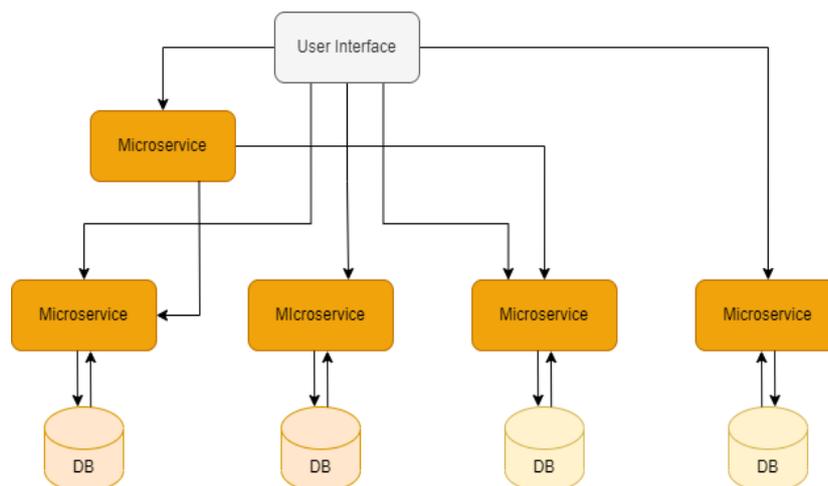
evaluasi dengan nilai *modification cost* pada arsitektur *microservice* sebesar 5,94 sedangkan arsitektur monolith sebesar 78,335. Setelah membandingkan hasil evaluasi berdasarkan sifat *maintainability*, *reusability*, *scalability*, dan *availability* diantara arsitektur monolith dengan *microservices*, maka dapat disimpulkan *microservices* lebih unggul dalam penerapannya dibandingkan dengan monolith.

Penelitian [4] mengimplementasikan arsitektur *microservice* dengan pola CQRS dan *event sourcing* pada OpenAPI, integrasi antar *service* berbasis API-driven dan *event-driven*. Penerapan arsitektur yang tepat dapat membuat performansi dari aplikasi menjadi lebih cepat. Terdapat dua aktivitas pengujian yang berguna untuk mengetahui perbedaan frekuensi request pada API-driven dan *event driven* yang dapat mempengaruhi *response time*, *error rate* dan juga *throughput*, serta menganalisis arsitektur mana yang memiliki performa yang lebih baik. Pengujian dilakukan dengan teknik *load test* yang menggunakan tool JMeter. Penelitian ini membuktikan bahwa *microservice* dengan pola CQRS dan *Event Sourcing* lebih cepat 3,7% dibandingkan dengan API-driven, serta komunikasi antar *service* tidak berpengaruh pada *error rate* dan *throughput*.

Tujuan penelitian ini adalah merancang aplikasi *e-ticketing* dengan menerapkan pendekatan arsitektur *microservice*, sebuah konsep yang mengurai sistem menjadi entitas layanan kecil dan otonom. Tidak hanya itu, penelitian ini juga memanfaatkan Apache Kafka sebagai *message broker* untuk menciptakan lingkungan komunikasi antar layanan yang optimal dan dapat diandalkan. Lebih lanjut, penelitian ini mengintegrasikan Kafka sebagai unsur kunci dalam mendukung pola arsitektur *event-driven*, di mana setiap layanan berinteraksi berdasarkan kejadian atau *event*, menjadikan sistem responsif terhadap perubahan dengan kecepatan dan efisiensi yang tinggi. Dengan demikian, penelitian ini menghadirkan suatu pendekatan yang holistik dalam merancang aplikasi *e-ticketing* yang tidak hanya berfokus pada skalabilitas dan kemandirian layanan, tetapi juga pada kemampuan sistem untuk menanggapi dinamika peristiwa secara adaptif dan efektif.

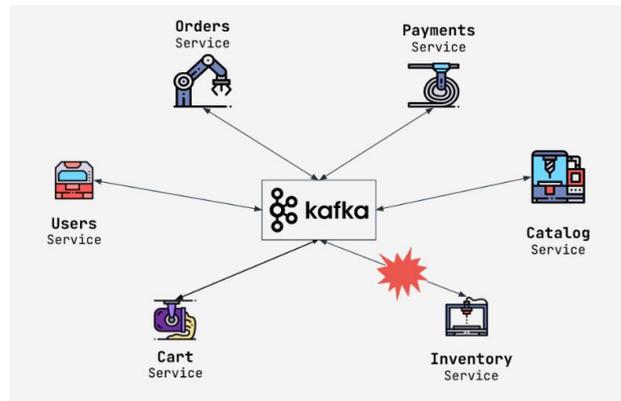
II. Metode

Metode pengembangan aplikasi yang digunakan pada penelitian ini adalah *microservice*. *Microservice* merupakan sebuah arsitektur yang memisahkan seluruh *service* menjadi sebuah program yang bersifat *independent* sehingga komunikasi dilakukan menggunakan teknik tertentu salah satunya adalah koneksi berbasis jaringan (*http protocol*) [5]. *Microservice architecture* merupakan arsitektur yang membagi suatu sistem kompleks dan besar menjadi *service – service* kecil dan sederhana yang dibangun secara *independent* [6]. Sistem yang kecil dan terbagi membuat sebuah *microservice* dapat bertahan dengan banyaknya sebuah *request* atau permintaan karena titik beban yang dibagi-bagi. Namun dalam perkembangannya teknik pembagian beban ini sudah tidak relevan karena adanya sebuah *delay* pada setiap *service* sehingga terdapat *broker* untuk menangani masalah tersebut [7].



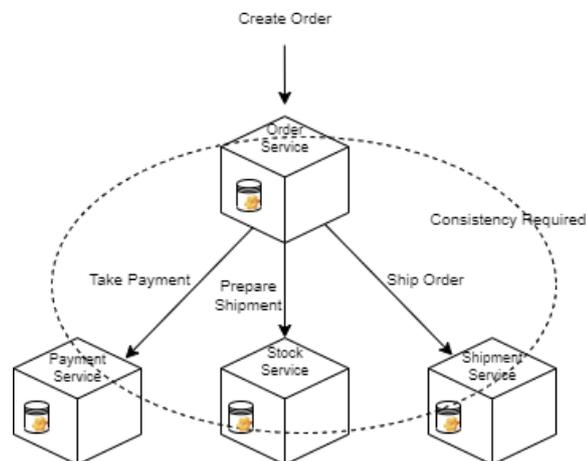
Gambar 1. Arsitektur *Microservice* [6]

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*. Adapun cara kerja apache kafka yaitu *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 [3].



Gambar 2. Apache Kafka

Event Driven Architecture menjadi bagian penting untuk membangun sistem *microservice*, dalam hal ini komunikasi *event driven* akan berjalan menggunakan *message broker*. *Event Driven Architecture* merupakan sebuah arsitektur terdistribusi asinkronus yang digunakan pada aplikasi yang membutuhkan skalabilitas dan performa yang tinggi. Dimana pada arsitektur ini sebuah sistem akan melakukan eksekusi setelah menerima satu atau beberapa *event* dari sistem lain. Pada *event driven architecture*, terdapat 2 jenis topologi utama, yakni topologi mediator dan topologi broker [8].

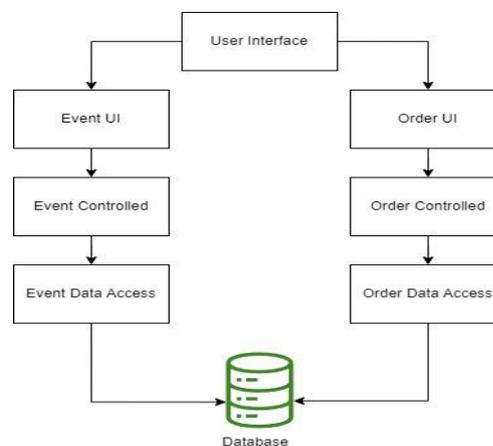


Gambar 3. Event Driven

III. Hasil dan Pembahasan

1. Arsitektur Sistem Monolitik *E-ticketing*

Adapun arsitektur sistem monolitik pada aplikasi *e-ticketing*, sebagai berikut:



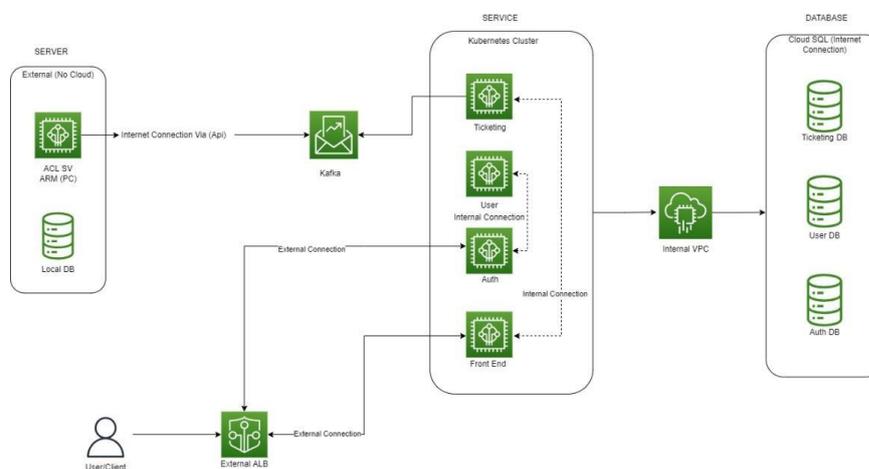
Gambar 4. Arsitektur sistem *E-ticketing*

Gambar 4 diatas merupakan arsitektur sistem *e-ticketing* dengan penjelasan sebagai berikut:

- User Interface mencakup *Event UI* dan *Order UI* pada sistem *e-ticketing*
- Event Controlled*, didalamnya terdapat *event* yang berjalan ataupun yang baru akan di buat,staff *event*.
- Order Controlled*, didalamnya terdapat transaksi order tiket, pembayaran serta refund.
- Event Data Access*, didalamnya keseluruhan transaksi *event* akan diurutkan dan di teruskan ke database
- Order Data Access*, didalamnya keseluruhan transaksi yang ada di order akan diurutkan dan di teruskan ke database

2. Arsitektur Sistem *E-ticketing* Berbasis *Microservice*

Hal tersebut akan didukung dengan pengembangan sistem informasi menggunakan web *service* dengan konsep infrastruktur *microservice* yakni sistem informasi dengan model arsitektur yang menyatukan aplikasi sebagai kumpulan layanan yang digabungkan secara fleksibel sesuai penerapan business rule. Arsitektur *microservice* memungkinkan pengiriman / penyebaran aplikasi besar dan kompleks secara terusmenerus sehingga memungkinkan suatu organisasi untuk mengembangkan sistem dengan berbagai teknologi yang berbeda-beda [9]. Adapun arsitektur sistem *microservice* pada aplikasi *e-ticketing*, sebagai berikut:



Gambar 5. Arsitektur Sistem *E-ticketing*

Adapun penjelasan terkait arsitektur sistem *e-ticketing* berbasis *microservice*, sebagai berikut:

- Di blok server terdapat ACL SV berfungsi untuk mengelola dan melayani permintaan klien atau pengguna dan database-nya yang berfungsi untuk menampung datanya, dimana kafka bertindak sebagai perantara ACL Server jika terdapat interaksi pada sistem.
- Service* dalam sistem ini, yaitu *Access Control List Service*, dimana Kafka bertindak sebagai perantara ketika terdapat interaksi atau bila terjadi *event* pada *Ticketing Service*, *Authentication Service* bertindak sebagai pemberi authorisasi pada semua *service* dengan memberikan token dan sistem unik (API Key) pada semua *service*, Front End atau tampilan yang berinteraksi langsung dengan pengguna, di dalamnya terdapat *Ticketing Content*, *Management System* dan *Access Control Lis Interface*. *External Application Load Balancer (ALB)* bertindak sebagai penghubung untuk koneksi eksternal (public) agar seluruh sistem aman dari tindak kejahatan.
- Ada 3 database, yaitu *Ticketing DB*, *User DB*, dan *Auth DB*. Untuk melindungi sistem dari serangan luar (public) terdapat *Internal VPC* koneksi jaringan yang bersifat internal.

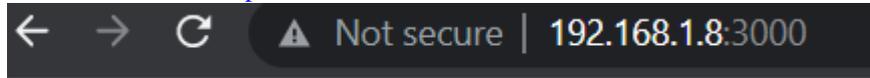
CMS (Content Management System) merupakan fasilitas yang disediakan untuk mempermudah pengelola data/konten website. setelah melakukan login melalui form yang tampil pada halaman website dengan alamat url <http://192.168.1.8:3000/>, anda akan diarahkan secara otomatis ke halaman Dashboard inilah pengelolaan data dilakukan.

cara serta langkah-langkah dipaparkan melalui bahasan berikut ini:

1. Login

Sebelum masuk ke halaman Dashboard, pengguna diharuskan melakukan login terlebih dahulu, untuk melakukan login ikuti langkah berikut :

- a. Buka aplikasi melalui browser
- b. Ketikkan alamat url : <http://192.168.1.8:3000/>



Gambar 6. Alamat Url

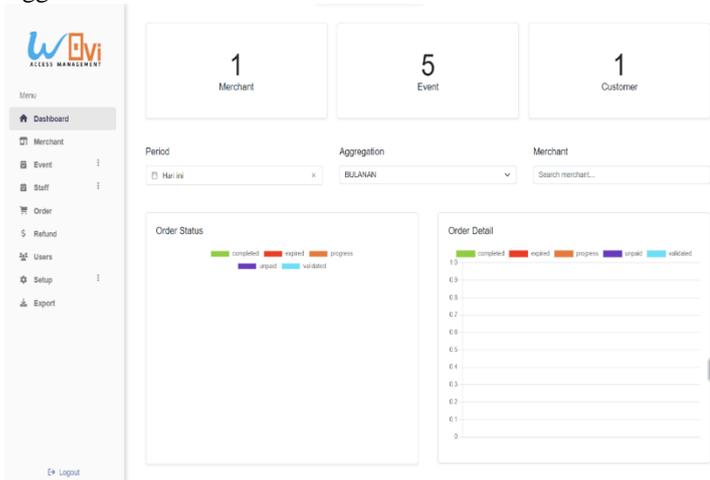
- c. Tekan tombol login, hingga tampil form login untuk mengakses halaman Dashboard

Gambar 7. Tampilan Login

- d. Jika tidak ada kesalahan pada username/password yang anda isikan, maka secara otomatis akan diarahkan ke halaman dashboard.

2. Menu Dashboard

Selanjutnya pengguna akan masuk ke halaman CMS Dashboard Wovi



Gambar 8. Menu Dashboard

Dashboard merupakan halaman untuk mengelolah situs anda. beberapa penjelasan yang ada di halaman Dashboard adalah sebagai berikut :

- a. Merchant, Adalah kotak yang menampilkan informasi keseluruhan jumlah dari Merchant.
- b. Event, Adalah kotak yang menampilkan keseluruhan jumlah dari Event.
- c. Customer, Adalah kotak yang menampilkan keseluruhan jumlah Customer.
- d. Periode, Adalah kotak yang di gunakan oleh pengguna untuk mengambil data grafik berdasarkan tanggal.
- e. Aggregation, Adalah kotak yang digunakan oleh pengguna untuk mengambil data grafik

berdasarkan bulan.

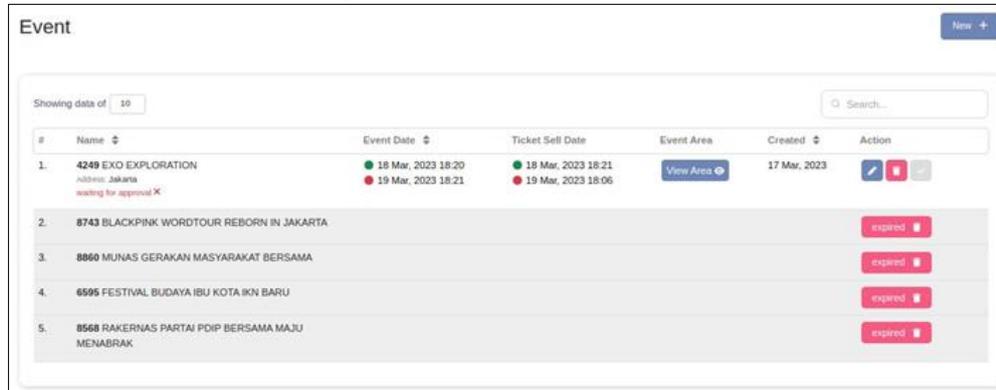
3. Menu Merchant

Pada menu ini pengguna di berikan informasi berupa jenis merchant. Pada menu ini juga pengguna dapat membuat data baru atau mengubah data merchant.

4. Menu *Event*

Pada menu ini terdapat dua submenu yaitu list *event* dan staff *event*, dapat di lihat pada gambar di bawah ini.

- a. *Event List*, submenu ini menampilkan daftar *event* yang telah di buat.

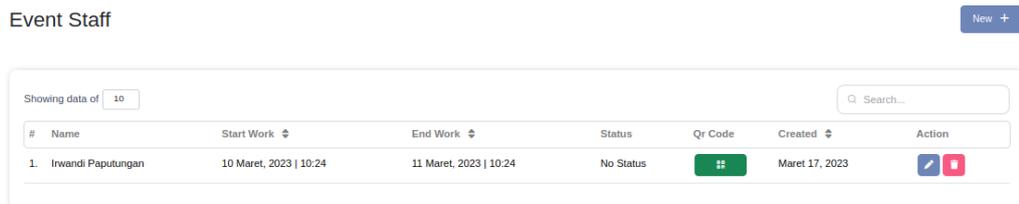


#	Name	Event Date	Ticket Sell Date	Event Area	Created	Action
1.	4249 EXO EXPLORATION Address: Jakarta waiting for approval X	18 Mar, 2023 18:20 19 Mar, 2023 18:21	18 Mar, 2023 18:21 19 Mar, 2023 18:06	View Area	17 Mar, 2023	[Edit] [Delete]
2.	8743 BLACKPINK WORDTOUR REBORN IN JAKARTA					expired [Delete]
3.	8060 MUNAS GERAKAN MASYARAKAT BERSAMA					expired [Delete]
4.	6595 FESTIVAL BUDAYA IBU KOTA IKN BARU					expired [Delete]
5.	8568 RAKERNAS PARTAI PDIP BERSAMA MAJU MENABRAK					expired [Delete]

Gambar 9. *Event List*

b. Staff *Event*

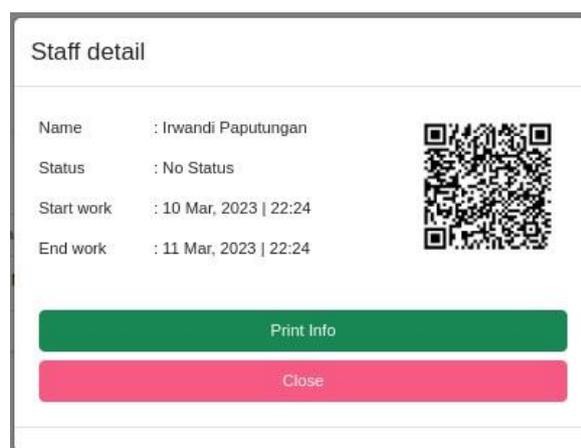
Pada menu ini pengguna dapat memperoleh informasi mengenai staff.



#	Name	Start Work	End Work	Status	Qr Code	Created	Action
1.	Irwandi Papatungan	10 Maret, 2023 10:24	11 Maret, 2023 10:24	No Status	[QR Code]	Maret 17, 2023	[Edit] [Delete]

Gambar 10. Data Staff

Untuk menampilkan info tentang staff, pengguna bisa mengklik pada menu Qr Code. QR Code mempermudah pengecekan data [10]. Setelah mengklik menu tersebut maka akan muncul tampilan seperti berikut.



Staff detail

Name : Irwandi Papatungan

Status : No Status

Start work : 10 Mar, 2023 | 22:24

End work : 11 Mar, 2023 | 22:24

[QR Code]

Print Info

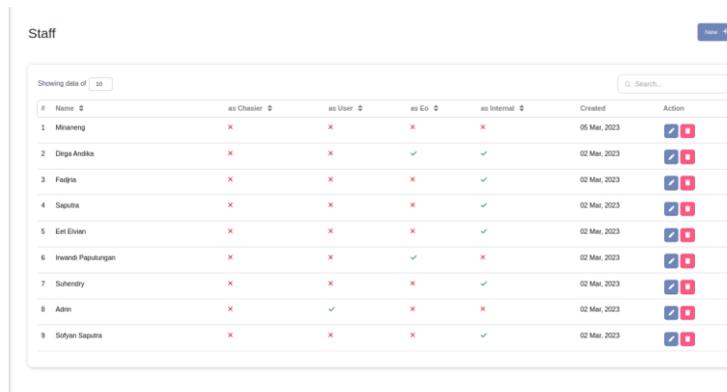
Close

Gambar 11. *QR Code* detail Staff

Pada gambar di atas terdapat dua button yaitu print info dan close. Button print info digunakan untuk menampilkan print preview sedangkan button close untuk keluar dari menu *QR Code*.

5. Menu Staff

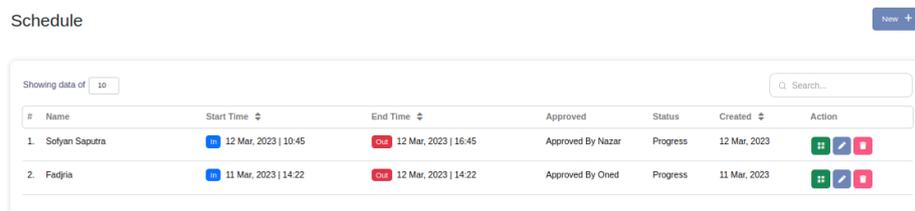
Submenu ini menampilkan daftar staff, dapat dilihat pada gambar di bawah.



#	Name	as Chasier	as User	as Co	as Internal	Created	Action
1	Miraning	x	x	x	x	05 Mar, 2023	[edit] [delete]
2	Diga Andika	x	x	✓	✓	02 Mar, 2023	[edit] [delete]
3	Fadria	x	x	x	✓	02 Mar, 2023	[edit] [delete]
4	Saputra	x	x	x	✓	02 Mar, 2023	[edit] [delete]
5	Ere Elvan	x	x	x	✓	02 Mar, 2023	[edit] [delete]
6	Inward Papanungan	x	x	✓	x	02 Mar, 2023	[edit] [delete]
7	Subenby	x	x	x	✓	02 Mar, 2023	[edit] [delete]
8	Adin	x	✓	x	x	02 Mar, 2023	[edit] [delete]
9	Sofyan Saputra	x	x	x	✓	02 Mar, 2023	[edit] [delete]

Gambar 12. Daftar Staff

Submenu ini memberikan informasi mengenai schedule dari setiap staff, dapat di lihat pada gambar di bawah ini.

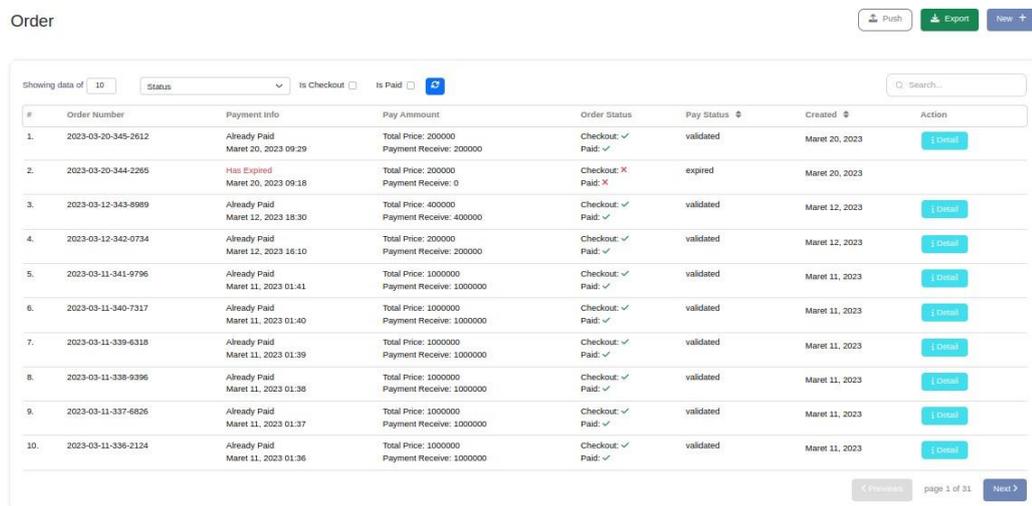


#	Name	Start Time	End Time	Approved	Status	Created	Action
1	Sofyan Saputra	12 Mar, 2023 10:45	12 Mar, 2023 16:45	Approved By Nazar	Progress	12 Mar, 2023	[edit] [delete]
2	Fadria	11 Mar, 2023 14:22	12 Mar, 2023 14:22	Approved By Oned	Progress	11 Mar, 2023	[edit] [delete]

Gambar 13. Data Schedule

6. Menu Order

Pada gambar di bawah pengguna diberikan informasi mengenai orderan dari costumer.



#	Order Number	Payment Info	Pay Ammount	Order Status	Pay Status	Created	Action
1.	2023-03-20-345-2612	Already Paid Maret 20, 2023 09:29	Total Price: 200000 Payment Receive: 200000	Checkout: ✓ Paid: ✓	validated	Maret 20, 2023	[Detail]
2.	2023-03-20-344-2265	Has Expired Maret 20, 2023 09:18	Total Price: 200000 Payment Receive: 0	Checkout: X Paid: X	expired	Maret 20, 2023	[Detail]
3.	2023-03-12-343-8989	Already Paid Maret 12, 2023 18:30	Total Price: 400000 Payment Receive: 400000	Checkout: ✓ Paid: ✓	validated	Maret 12, 2023	[Detail]
4.	2023-03-12-342-0734	Already Paid Maret 12, 2023 16:10	Total Price: 200000 Payment Receive: 200000	Checkout: ✓ Paid: ✓	validated	Maret 12, 2023	[Detail]
5.	2023-03-11-341-9796	Already Paid Maret 11, 2023 01:41	Total Price: 1000000 Payment Receive: 1000000	Checkout: ✓ Paid: ✓	validated	Maret 11, 2023	[Detail]
6.	2023-03-11-340-7317	Already Paid Maret 11, 2023 01:40	Total Price: 1000000 Payment Receive: 1000000	Checkout: ✓ Paid: ✓	validated	Maret 11, 2023	[Detail]
7.	2023-03-11-339-6318	Already Paid Maret 11, 2023 01:39	Total Price: 1000000 Payment Receive: 1000000	Checkout: ✓ Paid: ✓	validated	Maret 11, 2023	[Detail]
8.	2023-03-11-338-9396	Already Paid Maret 11, 2023 01:38	Total Price: 1000000 Payment Receive: 1000000	Checkout: ✓ Paid: ✓	validated	Maret 11, 2023	[Detail]
9.	2023-03-11-337-6826	Already Paid Maret 11, 2023 01:37	Total Price: 1000000 Payment Receive: 1000000	Checkout: ✓ Paid: ✓	validated	Maret 11, 2023	[Detail]
10.	2023-03-11-336-2124	Already Paid Maret 11, 2023 01:36	Total Price: 1000000 Payment Receive: 1000000	Checkout: ✓ Paid: ✓	validated	Maret 11, 2023	[Detail]

Gambar 14. Informasi Orderan

7. Menu Refund

Pada menu ini pengguna diberikan informasi tentang pengembalian dana (refund), dapat dilihat pada gambar berikut.

The screenshot shows a 'Refund' management interface. At the top, there is a search bar and a 'Showing data of 10' dropdown. Below is a table with the following columns: #, Order Id, Total Refund, Status, Created, and Action. The table contains 10 rows of data, all with a status of 'refunded ✓' and a creation date of 'Maret 10, 2023'. Each row has a red 'Action' button with a minus sign icon. At the bottom right, there are 'Previous' and 'Next' navigation buttons.

#	Order Id	Total Refund	Status	Created	Action
1.	285	1000000	refunded ✓	Maret 10, 2023	[Action]
2.	283	3700000	refunded ✓	Maret 10, 2023	[Action]
3.	289	3000000	refunded ✓	Maret 10, 2023	[Action]
4.	288	1800000	refunded ✓	Maret 10, 2023	[Action]
5.	280	4000000	refunded ✓	Maret 10, 2023	[Action]
6.	285	5000000	refunded ✓	Maret 10, 2023	[Action]
7.	283	2000000	refunded ✓	Maret 10, 2023	[Action]
8.	249	1000000	refunded ✓	Maret 10, 2023	[Action]
9.	249	8500000	refunded ✓	Maret 10, 2023	[Action]
10.	245	8000000	refunded ✓	Maret 10, 2023	[Action]

Gambar 15. Daftar Refund

Dapat kita lihat pada tampilan tersebut terdapat dua button yaitu button previews dan next. button previews digunakan untuk kembali ke slide sebelumnya sedangkan button next digunakan untuk lanjut ke slide berikutnya.

8. Menu Users

Pada menu ini kita diberikan informasi tentang pengguna (user). bisa dilihat pada gambar berikut.

The screenshot shows a 'Users' management interface. At the top right, there is a 'New +' button. Below is a table with the following columns: #, Name, Confirmed, Setup, Account, Contact, and Action. The table contains 5 rows of user data. Each row has a 'Setup' button (green with a plus icon) and a red 'Action' button with a minus sign icon. The 'Confirmed' column shows either a checkmark or 'Incumbent'.

#	Name	Confirmed	Setup	Account	Contact	Action
1.	Fadli Renaldi	Incumbent	[Setup]	Username: Fadli@Mail.Com Email: Fadli@Mail.Com	Phone: 628554546758 Telp: 0229090887	[Action]
2.	Fajriyan Crack	✓	[Setup]	Username: Fajrian Email: Fajri@Mail.Com	Phone: 626123343456 Telp: 028776787	[Action]
3.	Putri Kapasitor	✓	[Setup]	Username: Putri Email: Putri@Mail.Com	Phone: 6261275564756 Telp: 0229685767	[Action]
4.	Nazar Puki	✓	[Setup]	Username: Nazar Email: Nazar@Mail.Com	Phone: 626123343456 Telp: 0217767888	[Action]
5.	Invandi Papatungan	✓	Current Login	Username: Oned Email: Wandi@Mail.Com	Phone: 6262189037993 Telp: 0219803374	[Action]

Gambar 16. Data User

Pada tampilan di atas terdapat beberapa kolom yang harus di isi oleh pengguna sesuai dengan ketentuan. pada form tersebut juga terdapat dua button yaitu button cancel dan process. Button cancel di gunakan untuk membatalkan perubahan data pada user sedangkan button process digunakan untuk memproses data baru atau perubahan pada user. Pada setup terdapat dua button yaitu button save dan back. button save digunakan untuk menyimpan data yang telah di edit sedangkan button back digunakan untuk kembali ke tampilan sebelumnya.

9. Menu Setup

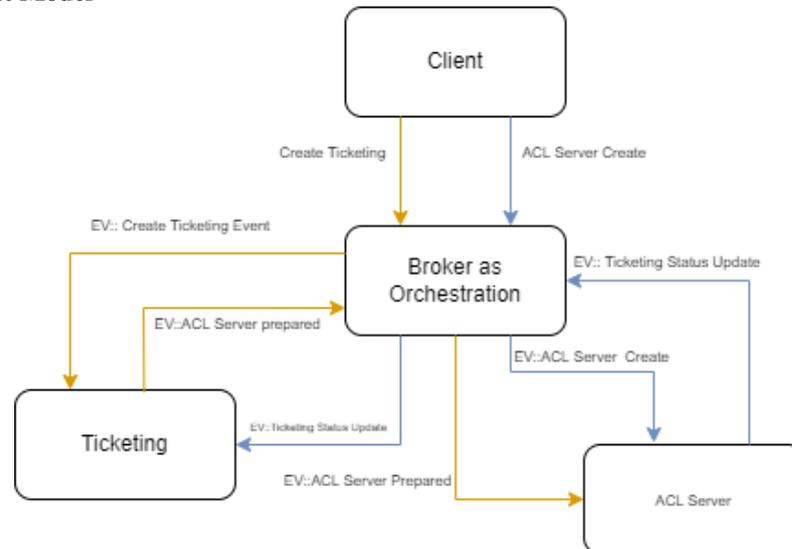
Pada menu ini terdapat payment method yaitu menu yang mengatur metode pembayaran

The screenshot shows a 'Payment Method' management interface. At the top right, there is a 'New +' button. Below is a table with the following columns: #, Name, Created, and Action. The table contains 1 row of data for 'CASH' with a creation date of '24 Februari, 2023'. Each row has a blue 'Action' button with a plus icon and a red 'Action' button with a minus sign icon.

#	Name	Created	Action
1.	CASH	24 Februari, 2023	[Action] [Action]

Gambar 17. Tampilan Setup

a. Event Request Model



Gambar 18. Alur Event Request Model

Gambar 18 merupakan *event request model* dimana *Client Service* melakukan aksi *Create Ticketing* dan *ACL Server Create*, kemudian meneruskannya ke *Broker* yang bertindak sebagai Orkestrasi. Kemudian, *Broker* melakukan aksi *Create Ticketing Event* ke *Ticketing Service* dan *Ticketing Service* mengembalikan *event* ke *broker* berupa *ACL server prepared*, kemudian *broker* meneruskan *event* ke *ACL Server* dan *ACL Server* mengembalikan *Event* ke *Broker* berupa *Ticketing Status Update*.

IV. Kesimpulan dan saran

Berdasarkan analisis data yang dikumpulkan dari kasus yang diangkat, dapat disimpulkan bahwa perancangan aplikasi *e-ticketing* dengan model arsitektur *microservice* menggunakan Kafka memberikan solusi yang efektif terhadap permasalahan yang muncul. Pendekatan ini memungkinkan pengembangan sistem yang lebih optimal, menghindarkan penumpukan permintaan pada satu layanan seperti yang sering terjadi dalam praktik perusahaan saat ini.

Daftar Pustaka

- [1] R. Lisgiani and S. Nurmajid, "Implementasi Autentikasi Dari Sisi Backend Pada Arsitektur Microservices Menggunakan Express JS," *Infotronik J. Teknol. Inf. dan Elektron.*, vol. 7, no. 1, p. 27, Jun. 2022, doi: 10.32897/infotronik.2022.7.1.1330.
- [2] U. Syarif and P. Pizaini, "Penerapan Event-Driven Microservices Pada Aplikasi Layanan Penerimaan Peserta Didik Baru," *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 7, no. 3, pp. 745–756, Aug. 2022, doi: 10.29100/jipi.v7i3.3067.
- [3] F. Nabawi, "Jurnal Implementasi Sistem Distribusi Pesan dan Proses Data Secara Real Time dengan Apache Kafka," *J. Teknol. Inform. dan Komput.*, vol. 8, no. 1, pp. 173–189, Mar. 2022, doi: 10.37012/jtik.v8i1.836.
- [4] D. Al Fansha, M. Y. H. Setyawan, and M. N. Fauzan, "Load Test pada Microservice yang menerapkan CQRS dan Event Sourcing," *J. Buana Inform.*, vol. 12, no. 2, pp. 126–134, Nov. 2021, doi: 10.24002/jbi.v12i2.4749.
- [5] Y. C. T. Putra, T. A. P. Sidi, and J. E. Samodra, "Implementasi Arsitektur Microservice pada Aplikasi Web Pengajaran Agama Islam Home Pesantren," *J. Inform. Atma Jogja*, vol. 1, no. 1, pp. 88–97, 2020.
- [6] K. Sayyidi, R. Surya, A. Bhawiyuga, and A. Basuki, "Implementasi Monitoring Platform untuk Microservice Architecture menggunakan Elastic Stack (Studi Kasus : Alterra Indonesia)," vol. 6, no. 4, pp. 1668–1676, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [7] D. R. L. Hayun and W. Wibisono, "Optimasi Pemilihan Child Broker(S) Pada Model Komunikasi Publish/Subscribe Pada Protokol Data Distribution Service Di Area Multi-Zone," *JUTI J. Ilm. Teknol. Inf.*, vol. 15, no. 1, p. 11, Jan. 2017, doi: 10.12962/j24068535.v15i1.a631.
- [8] R. Adila, R. J. Akbar, and H. Studiawan, "Rancang Bangun Modul Portofolio Pegawai pada Aplikasi MyITS Human Capital Management dengan Arsitektur Event Driven," *J. Tek. ITS*, vol. 11, no. 1, Apr.

-
- 2022, doi: 10.12962/j23373539.v11i1.82475.
- [9] P. L. Lokapitasari Belluano, P. Purnawansyah, B. L. E. Panggabean, and H. Herman, "Sistem Informasi Program Kreativitas Mahasiswa berbasis Web Service dan Microservice," *Ilk. J. Ilm.*, vol. 12, no. 1, pp. 8–16, Apr. 2020, doi: 10.33096/ilkom.v12i1.492.8-16.
- [10] M. A. Asis and L. B. Ilmawan, "Implementasi Quick Response (QR) Code pada Aplikasi Pengajuan Surat Keterangan Pendamping Ijazah (App-SKPI)," *Bul. Sist. Inf. dan Teknol. Islam*, vol. 4, no. 1, pp. 9–14, 2023, doi: 10.33096/busiti.v4i1.1611.