

Perancangan Pembuatan Program Investasi Saham dan Reksa Dana

Suma Renata Wijaya

Jurusan Informatika, UIN Sunan Gunung Djati Bandung, Indonesia

Info Artikel

Sejarah artikel:

Diterima pada 23 Desember
2023

Kata kunci:

Investasi Saham
Reksa Dana
Arsitektur Microservices
User Service
Trading Service

ABSTRAK

Artikel ini membahas perancangan program investasi saham dan reksa dana dengan menggunakan Arsitektur Microservices. Dalam rancangan ini, terdapat beberapa layanan penting seperti User Service, Trading Service, Portfolio Service, Pricing Service, dan Audit Service. Tujuan utama dari artikel ini adalah untuk mensimulasikan pekerjaan seorang arsitek perangkat lunak dalam merancang program. Solusi yang disarankan dalam artikel ini adalah penggunaan Arsitektur Microservices. Arsitektur ini dikenal dapat meningkatkan fleksibilitas dan skalabilitas sistem, sehingga sangat cocok untuk digunakan dalam perancangan program investasi. Temuan penting dalam artikel ini mencakup efisiensi dan efektivitas Arsitektur Microservices dalam mengelola berbagai layanan dalam program investasi. Kesimpulan utama yang dapat diambil dari artikel ini adalah bahwa Arsitektur Microservices dapat menjadi solusi yang kuat dan efektif untuk perancangan program investasi. Dengan demikian, artikel ini memberikan wawasan baru dan penting bagi para arsitek perangkat lunak dan pengembang yang tertarik dalam merancang program investasi dengan menggunakan Arsitektur Microservices.

Penulis yang sesuai:

Suma Renata Wijaya,
Jurusan Informatika Fakultas Sains & Teknologi UIN Sunan Gunung Djati Bandung
Jl. AH Nasution No.105, Cibiru, Bandung, Indonesia. 40614
Email: sumastudy101@gmail.com

1. PENDAHULUAN (10 PT)

Pada era digital saat ini, investasi saham dan reksa dana telah menjadi bagian integral dari ekonomi global. Dengan kemajuan teknologi, transaksi finansial semakin bergantung pada sistem digital. Namun, perancangan program untuk transaksi jenis ini sering kali menimbulkan tantangan tersendiri. Salah satu tantangan utama adalah menciptakan sistem yang efisien dan efektif. Dalam konteks ini, peran seorang arsitek perangkat lunak menjadi sangat penting[1][2].

Dalam artikel ini, kita membahas permasalahan utama dalam mencari jenis arsitektur yang paling efisien dan efektif untuk program investasi saham dan reksa dana. Ini adalah masalah yang kompleks dan memerlukan pemahaman mendalam tentang berbagai aspek teknis dan bisnis[1][2]. Berbagai literatur terkait, baik dari buku maupun jurnal kredibel, telah menunjukkan bahwa Arsitektur Microservices dapat menjadi solusi yang tepat untuk permasalahan ini[1][2][3][4].

Arsitektur Microservices dipilih karena beberapa alasan. Pertama, arsitektur ini memungkinkan setiap layanan untuk dirancang, dikembangkan, dan dikelola secara terpisah. Ini berarti bahwa setiap layanan dapat diubah, ditingkatkan, atau diganti tanpa mengganggu layanan lain dalam sistem. Ini

memberikan fleksibilitas yang besar dan memungkinkan sistem untuk berkembang seiring dengan perubahan kebutuhan bisnis[1][2][3][4].

Kedua, arsitektur ini juga mendukung struktur yang terorganisir dengan baik, di mana setiap layanan dapat terhubung melalui satu manajemen. Ini memudahkan koordinasi antar layanan dan memastikan bahwa semua layanan dapat bekerja sama dengan efektif untuk mencapai tujuan bisnis[1][2][3][4].

Ketiga, arsitektur ini memungkinkan peningkatan fleksibilitas dan skalabilitas sistem. Dengan kemampuan untuk menambah atau mengurangi layanan sesuai kebutuhan, sistem dapat dengan mudah disesuaikan untuk menangani beban kerja yang meningkat atau berkurang[1][2][3][4].

Nilai baru dari penelitian ini adalah menunjukkan bahwa program transaksi yang terdiri dari berbagai layanan, seperti transaksi investasi dalam konteks ini, sangat cocok untuk menggunakan Arsitektur Microservices. Dengan demikian, penelitian ini memberikan kontribusi penting bagi praktisi dan peneliti di bidang arsitektur perangkat lunak dan investasi[1][2][3][4].

2. METODE

Desain Penelitian Penelitian ini dirancang sebagai studi eksploratif untuk memahami bagaimana Arsitektur Microservices dapat digunakan dalam perancangan program investasi saham dan reksa dana[1][2]. Dalam rancangan ini, terdapat beberapa layanan penting seperti User Service, Trading Service, Portfolio Service, Pricing Service, dan Audit Service[1][2].

Prosedur Penelitian Penelitian ini dimulai dengan studi literatur yang mendalam tentang Arsitektur Microservices dan aplikasinya dalam perancangan program investasi[1][2][3][4]. Selanjutnya, kami merancang dan mengembangkan program investasi menggunakan Arsitektur Microservices. Setiap layanan dirancang, dikembangkan, dan dikelola secara terpisah[1][2][3][4].

Pengujian Setelah program selesai dikembangkan, kami melakukan serangkaian pengujian untuk memastikan bahwa setiap layanan berfungsi dengan baik dan sistem secara keseluruhan beroperasi dengan efisiensi[1][2][3][4]. Pengujian ini melibatkan berbagai skenario transaksi investasi untuk memastikan bahwa sistem dapat menangani berbagai jenis transaksi dengan baik[1][2][3][4].

Perolehan Data Data diperoleh selama proses pengujian[1][2][3][4]. Kami mencatat kinerja setiap layanan dan sistem secara keseluruhan dalam berbagai skenario transaksi. Data ini kemudian dianalisis untuk menentukan efisiensi dan efektivitas Arsitektur Microservices dalam konteks program investasi[1][2][3][4].

3. HASIL DAN PEMBAHASAN

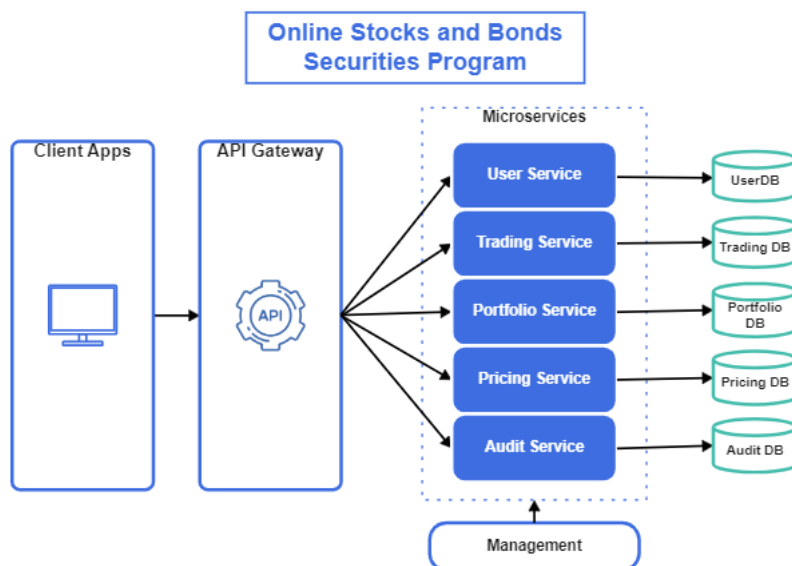
Microservices adalah desain arsitektur untuk membuat sebuah aplikasi yang terdiri dari berbagai unit layanan tersendiri tapi tetap saling terhubung. Setiap unit layanan dalam aplikasi tersebut menjalankan fungsi berbeda, tapi tetap mendukung satu sama lain. Dengan kata lain, microservices sama dengan membangun aplikasi dalam aplikasi.

Karakter utama dari Arsitektur Microservices:

- Multiple Component Services:** Microservices terdiri dari komponen layanan individu, longgar, dan dapat dikembangkan, dikerahkan, dioperasikan, diubah, dan dikerahkan kembali tanpa mengganggu fungsi layanan lain atau integritas aplikasi.
- Highly Maintainable and Testable:** Microservices memungkinkan tim untuk bereksperimen dengan fitur baru dan menggulirkan kembali jika mereka tidak berhasil. Ini memudahkan pembaruan kode dan mempercepat waktu ke pasar untuk fitur baru.
- Owned by Small Teams:** Tim kecil dan independen biasanya membangun layanan dalam microservices, yang mendorong adopsi praktik agile dan DevOps.
- Organized Around Business Capabilities:** Pendekatan microservices mengatur layanan berdasarkan kemampuan bisnis. Tim adalah lintas fungsional dan memiliki berbagai keterampilan yang diperlukan untuk pengembangan.
- Automated Infrastructure:** Tim yang membangun dan memelihara microservices biasanya menggunakan praktik otomatisasi infrastruktur seperti integrasi terus menerus (CI), pengiriman (CD), dan penyebaran (juga CD).

Microservices adalah teknologi yang mulai banyak perusahaan adopsi, terutama yang bergerak secara digital. Bukan tanpa sebab, arsitektur microservices memang memungkinkan startup dan perusahaan melayani konsumen dengan lebih baik, meski secara virtual. Mulai dari menawarkan berbagai layanan, sampai memproses transaksi yang lebih cepat.

3.1. Perancangan



Dalam arsitektur Microservices, aplikasi dibagi menjadi serangkaian layanan yang lebih kecil dan terisolasi yang dapat dikembangkan, diuji, dan dikerahkan secara independen.

- a. Client Apps: Ini adalah aplikasi yang digunakan oleh pengguna untuk berinteraksi dengan sistem. Aplikasi ini dapat berupa aplikasi web, mobile, atau desktop.
- b. API Gateway: Ini bertindak sebagai titik masuk tunggal ke sistem untuk aplikasi klien. API Gateway bertanggung jawab untuk mengarahkan permintaan dari aplikasi klien ke layanan yang tepat.
- c. Microservices: Ini adalah layanan individu yang membentuk aplikasi. Dalam kasus ini, layanan tersebut adalah User Service, Trading Service, Portfolio Service, Pricing Service, dan Audit Service. Setiap layanan ini memiliki tanggung jawab yang jelas dan terisolasi dari layanan lain.
 1. User Service: Layanan ini bertanggung jawab untuk mengelola data pengguna. Ini mencakup autentikasi pengguna, manajemen profil, dan penanganan data pribadi pengguna. Layanan ini memastikan bahwa setiap pengguna memiliki akses yang tepat dan aman ke aplikasi.
 2. Trading Service: Layanan ini bertanggung jawab untuk menangani semua aspek perdagangan saham dan reksa dana. Ini mencakup pembelian dan penjualan saham atau reksa dana, pelacakan pesanan, dan manajemen portofolio. Layanan ini memastikan bahwa semua transaksi dilakukan dengan lancar dan efisien.
 3. Portfolio Service: Layanan ini bertanggung jawab untuk mengelola portofolio investasi pengguna. Ini mencakup pelacakan kinerja investasi, analisis risiko, dan rekomendasi investasi. Layanan ini membantu pengguna membuat keputusan investasi yang lebih baik berdasarkan data dan analisis terkini.
 4. Pricing Service: Layanan ini bertanggung jawab untuk menyediakan data harga real-time untuk saham dan reksa dana. Ini memastikan bahwa pengguna selalu memiliki informasi harga terkini saat membuat keputusan perdagangan.
 5. Audit Service: Layanan ini bertanggung jawab untuk melacak dan mencatat semua aktivitas dalam sistem. Ini mencakup pelacakan transaksi, perubahan data pengguna, dan aktivitas sistem lainnya. Layanan ini memastikan bahwa ada catatan lengkap dari semua aktivitas yang dapat digunakan untuk tujuan audit dan kepatuhan.
- d. Management: Ini adalah komponen yang bertanggung jawab untuk mengelola dan mengkoordinasikan layanan. Ini dapat mencakup tugas seperti penjadwalan, pemantauan, dan penyebaran layanan.
- e. Database: Setiap layanan memiliki database sendiri yang digunakan untuk menyimpan data yang relevan dengan layanan tersebut. Ini memungkinkan setiap layanan untuk mengelola datanya sendiri dan memastikan bahwa tidak ada layanan lain yang dapat mengakses data tersebut secara langsung.

3.2. Operasi Program

a. Add user: Perintah ini digunakan untuk menambahkan pengguna baru ke sistem. Ini melibatkan pembuatan profil pengguna dengan detail seperti nama, alamat email, dan informasi kontak lainnya. Selain itu, pengaturan detail akun seperti nama pengguna dan kata sandi juga dilakukan pada tahap ini.

```
==STOCKS AND BONDS TRANSACTION PROGRAM==
Command List:
1. Add user
2. Remove user
3. Get user
4. Buy stock
5. Sell stock
6. Buy bond
7. Sell bond
8. Get total stock value
9. Get total bond value
10. Get transactions
11. Exit
Enter command number:
1

== ADD USER ==
Enter user ID: Suma
Enter user name: suma
= User Added =
```

b. Remove user: Perintah ini digunakan untuk menghapus pengguna dari sistem. Ini melibatkan penghapusan profil pengguna dan semua data terkait seperti detail akun, riwayat transaksi, dan portofolio investasi. Perintah ini biasanya digunakan saat pengguna memilih untuk menghapus akun mereka atau jika akun tersebut dinonaktifkan oleh administrator.

```
== REMOVE USER ==
Enter user ID: Suma
= User removed =
```

c. Buy stock: Perintah ini digunakan untuk membeli saham. Ini melibatkan penentuan jumlah saham untuk dibeli dan eksekusi transaksi pembelian. Transaksi ini dicatat dalam sistem dan portofolio pengguna diperbarui sesuai.

```
== BUY STOCKS ==
Stocks List:
0 : PT Bank Central Asia Tbk (BBCA) : Rp9250/share
1 : PT Bank Rakyat Indonesia (Persero) Tbk (BBRI) : Rp5550/share
2 : PT Bank Mandiri (Persero) Tbk (BMRI) : Rp5975/share
3 : PT Bank Negara Indonesia (Persero) Tbk (BBNI) : Rp5200/share
4 : PT Astra International Tbk (ASII) : Rp5650/share
Enter user ID:
Suma
Enter stock index:
1
Enter lot quantity:
20
Suma bought stocks with the amount 20lot of BBRI
== Transaction Success ==
```

c. Sell stock: Perintah ini digunakan untuk menjual saham. Ini melibatkan penentuan jumlah saham untuk dijual dan eksekusi transaksi penjualan. Seperti pembelian saham, transaksi ini juga dicatat dan portofolio pengguna diperbarui.

```

== SELL STOCKS ==

Stocks:
0 : PT Bank Central Asia Tbk (BBCA) : Rp9250/share
1 : PT Bank Rakyat Indonesia (Persero) Tbk (BBRI) : Rp5550/share
2 : PT Bank Mandiri (Persero) Tbk (BMRI) : Rp5975/share
3 : PT Bank Negara Indonesia (Persero) Tbk (BBNI) : Rp5200/share
4 : PT Astra International Tbk (ASII) : Rp5650/share
Enter user ID:
Suma
== Sell Input ==
Enter stock index:
1
Enter lot quantity:
5

== Stock Sold ==
User ID: Suma
Quantity: 5 lot
Stock Name: BBRI
== Transaction Success ==

```

d. Buy bond: Perintah ini digunakan untuk membeli unit reksa dana. Ini melibatkan penentuan jumlah unit untuk dibeli dan eksekusi transaksi pembelian. Transaksi ini dicatat dalam sistem dan portofolio pengguna diperbarui sesuai

```

== BUY BONDS ==

Bonds:
0 : Sucorinvest Equity Fund
1 : Sucorinvest Maxi Fund
2 : Sucorinvest Sharia Equity Fund
3 : Trim Syariah Saham
4 : Manulife Saham Andalan
Enter user ID:
Suma
Enter bond index:
1
Enter quantity:
5
Suma BOUGHT bonds with the amount 5 of Sucorinvest Maxi Fund
== Transaction Success ==

```

e. Sell bond: Perintah ini digunakan untuk menjual unit reksa dana. Ini melibatkan penentuan jumlah unit untuk dijual dan eksekusi transaksi penjualan. Seperti pembelian reksa dana, transaksi ini juga dicatat dan portofolio pengguna diperbarui.

```
== SELL BONDS ==  
Bonds:  
0 : Sucorinvest Equity Fund  
1 : Sucorinvest Maxi Fund  
2 : Sucorinvest Sharia Equity Fund  
3 : Trim Syariah Saham  
4 : Manulife Saham Andalan  
Enter user ID:  
Suma  
Enter bond index:  
1  
Enter quantity:  
2  
Suma SOLD bonds with the amount 2 of Sucorinvest Maxi Fund  
== Transaction Success ==
```

f. Get total stock value: Perintah ini digunakan untuk menghitung nilai total saham yang dimiliki oleh pengguna. Ini melibatkan penjumlahan nilai semua saham yang dimiliki oleh pengguna berdasarkan harga pasar saat ini.

```
Command List:  
1. Add user  
2. Remove user  
3. Get user  
4. Buy stock  
5. Sell stock  
6. Buy bond  
7. Sell bond  
8. Get total stock value  
9. Get total bond value  
10. Get transactions  
11. Exit  
Enter command number:  
8  
  
Enter user ID:  
Suma  
Total stock value: 8325000
```

g. Get total bond value: Perintah ini digunakan untuk menghitung nilai total unit reksa dana yang dimiliki oleh pengguna. Ini melibatkan penjumlahan nilai semua unit reksa dana yang dimiliki oleh pengguna berdasarkan nilai aset bersih per unit (NAVPU) saat ini.

```
Command List:
1. Add user
2. Remove user
3. Get user
4. Buy stock
5. Sell stock
6. Buy bond
7. Sell bond
8. Get total stock value
9. Get total bond value
10. Get transactions
11. Exit
Enter command number:
9

Enter user ID:
Suma
Total bond value: 3000000
```

h. Get transactions: Perintah ini digunakan untuk mengambil daftar transaksi yang telah dilakukan oleh pengguna. Ini bisa mencakup transaksi pembelian dan penjualan saham dan reksa dana, serta transaksi lainnya yang terkait dengan akun pengguna.

```
Command List:
1. Add user
2. Remove user
3. Get user
4. Buy stock
5. Sell stock
6. Buy bond
7. Sell bond
8. Get total stock value
9. Get total bond value
10. Get transactions
11. Exit
Enter command number:
10

Enter user ID:
Suma
Transaction{userId='Suma', type='buyStock', assetName='BBRI', quantity=15}
Transaction{userId='Suma', type='buyBond', assetName='Sucorinvest Maxi Fund', quantity=5}
Transaction{userId='Suma', type='sellBond', assetName='Sucorinvest Maxi Fund', quantity=2}
```


i. Exit: Perintah ini digunakan untuk keluar dari sistem atau mengakhiri sesi pengguna saat ini. Ini biasanya melibatkan penutupan koneksi ke sistem dan pembersihan sumber daya yang digunakan selama sesi.

```

Command List:
1. Add user
2. Remove user
3. Get user
4. Buy stock
5. Sell stock
6. Buy bond
7. Sell bond
8. Get total stock value
9. Get total bond value
10. Get transactions
11. Exit
Enter command number:
11
Goodbye, Have a great day!
PS C:\Users\ASUS\Documents\Investing Program>

```

6. KESIMPULAN

Hasil penelitian ini menunjukkan bahwa Arsitektur Microservices dapat menjadi solusi yang efektif untuk perancangan program investasi. Setiap layanan dalam arsitektur ini dapat dirancang, dikembangkan, dan dikelola secara independen, memungkinkan fleksibilitas dan skalabilitas yang lebih besar. Selain itu, penggunaan API Gateway memastikan bahwa semua layanan dapat berinteraksi dengan lancar dan efisien.

Mengenai prospek pengembangan hasil penelitian, ada potensi untuk mengeksplorasi lebih lanjut bagaimana Arsitektur Microservices dapat diaplikasikan dalam konteks lain di luar investasi saham dan reksa dana. Selain itu, penelitian lanjutan dapat fokus pada optimasi performa dan keamanan dalam Arsitektur Microservices. Secara keseluruhan, penelitian ini membuka jalan untuk berbagai kemungkinan penelitian dan aplikasi lanjutan dalam bidang arsitektur perangkat lunak dan investasi.

REFERENSI (10 PT)

- [1] Valdivia, J.A., Lora-González, A., Limón, X. et al. Patterns Related to Microservice Architecture: a Multivocal Literature Review. *Program Comput Soft* 46, 594–608 (2020). <https://doi.org/10.1134/S0361768820080253>
- [2] Aljawawdeh, H., Sabri, M., Maghrabi, L. (2023). Toward Serverless and Microservices Architecture: Literature, Methods, and Best Practices. In: Hannon, A., Mahmood, A. (eds) *Artificial Intelligence, Internet of Things, and Society 5.0. Studies in Computational Intelligence*, vol 1113. Springer, Cham. https://doi.org/10.1007/978-3-031-43300-9_47 [3]C. Shorten dan TM Khoshgoftaar, "Survei tentang Augmentasi Data Gambar untuk Pembelajaran Mendalam," *Jurnal Big Data*, vol. 6, tidak. 1, 2019, doi: 10.1186/s40537-019-0197-0.
- [4] Mehmet Soylemez, Bedir Tekinerdogan, Ayca Kolukisa Tarhan. Challenges and Solution Directions of Microservice Architectures: A Systematic Literature Review. Department of Computer Engineering, Hacettepe University, 06800 Ankara, Turkey. *Information Technology Group, Wageningen University & Research*, 6706 PB Wageningen, The Netherlands. *Appl. Sci.* 2022, 12(11), 5507; <https://doi.org/10.3390/app12115507>