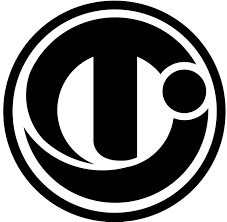
MAKALAH MICROSERVICE

“KEAMANAN DALAM MICROSERVICE”



Di susun oleh :

DEYA MIRZA (2301081004)

Dosen Pengampu :

Ervan Asri, S.kom.,M.kom

KELAS TEKNIK KOMPUTER 2 B

PROGRAM STUDI TEKNIK KOMPUTER

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI PADANG

2024

BAB 1 PENDAHULUAN

Perkembangan teknologi informasi yang pesat mendorong banyak organisasi untuk beralih dari arsitektur monolitik ke arsitektur microservice. Microservice menawarkan skalabilitas, fleksibilitas, dan kecepatan pengembangan yang lebih tinggi. Namun, seiring meningkatnya jumlah layanan kecil yang saling berkomunikasi, kompleksitas dalam menjaga keamanan juga meningkat. Oleh karena itu, pemahaman tentang keamanan dalam microservice menjadi sangat penting untuk menjaga integritas, kerahasiaan, dan ketersediaan sistem.

**1.2 Rumusan Masalah**

* Apa saja tantangan keamanan dalam arsitektur microservice?
* Strategi dan teknologi apa yang dapat digunakan untuk mengamankan microservice?

**1.3 Tujuan**

Makalah ini bertujuan untuk:

* Menjelaskan risiko dan tantangan keamanan pada microservice.
* Menyampaikan strategi terbaik dalam mengimplementasikan keamanan microservice.

**BAB II – TINJAUAN PUSTAKA**

**2.1 Arsitektur Microservice**

Microservice adalah pendekatan pengembangan perangkat lunak yang membagi aplikasi menjadi layanan-layanan kecil yang dapat berjalan secara independen dan berkomunikasi melalui API (Application Programming Interface).

**2.2 Perbedaan Microservice dan Monolitik**

Pada arsitektur monolitik, semua komponen sistem berada dalam satu unit. Sebaliknya, microservice mengisolasi setiap fitur utama menjadi layanan mandiri, sehingga memberikan keuntungan dalam hal skalabilitas dan pemeliharaan. Namun, hal ini juga membuka potensi serangan yang lebih luas karena banyaknya titik komunikasi.

**BAB III – PEMBAHASAN**

**3.1 Tantangan Keamanan Microservice**

Beberapa tantangan keamanan utama dalam microservice meliputi:

* **Permukaan Serangan yang Luas**  
  Banyaknya layanan berarti banyak endpoint yang bisa diserang.
* **Manajemen Identitas dan Akses (IAM)**  
  Autentikasi dan otorisasi harus dilakukan di setiap layanan, tidak cukup hanya di satu titik.
* **Komunikasi Antar-Layanan**  
  Microservice menggunakan komunikasi jaringan (biasanya HTTP/gRPC), yang bisa disadap jika tidak dienkripsi.
* **Isolasi Layanan**  
  Kerentanan dalam satu layanan bisa berdampak pada layanan lain jika tidak diisolasi dengan baik.

**3.2 Strategi Keamanan dalam Microservice**

**a. Autentikasi dan Otorisasi Terpusat (OAuth2, JWT)**

Gunakan protokol seperti OAuth2 untuk autentikasi, dan JSON Web Token (JWT) untuk otorisasi antar layanan.

**b. Gateway API**

Gunakan API Gateway sebagai titik masuk tunggal. Gateway dapat mengelola autentikasi, routing, rate limiting, dan logging.

**c. TLS/SSL**

Semua komunikasi antar layanan harus dienkripsi dengan TLS untuk mencegah sniffing dan MITM (Man in The Middle).

**d. Zero Trust Security Model**

Setiap layanan harus saling memverifikasi identitas, meskipun berada dalam satu jaringan.

**e. Rate Limiting dan Throttling**

Untuk mencegah serangan DoS (Denial of Service), tetapkan batasan permintaan.

**f. Monitoring dan Logging**

**BAB IV – KESIMPULAN**

Arsitektur microservice membawa banyak keuntungan dalam hal fleksibilitas dan skalabilitas, namun juga menambah kompleksitas pada aspek keamanan. Permukaan serangan yang lebih luas, komunikasi antar-layanan, dan kebutuhan akan otentikasi yang lebih granular membuat pendekatan keamanan tradisional menjadi tidak cukup.  
Diperlukan strategi komprehensif seperti enkripsi komunikasi, manajemen identitas yang kuat, pemantauan berkelanjutan, dan penerapan prinsip Zero Trust. Dengan pengelolaan yang tepat, sistem microservice dapat tetap aman dan handal dalam mendukung kebutuhan bisnis modern.

**DAFTAR PUSTAKA**

1. Newman, S. (2015). *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media.
2. NGINX. (2021). *Microservices Security*. <https://www.nginx.com>
3. OWASP Foundation. (2022). *OWASP Cheat Sheet Series - Microservices Security*
4. Kubernetes.io. (2022). *Security Best Practices for Kubernetes*