**MAKALAH**

**MICROSERVICE**



Disusun oleh:

|  |  |
| --- | --- |
| Nama: | Syfa Aulia |
| Nim : | 2301081017 |

Dosen Pengampu

Ervan Aasri, S.Kom.,M.Kom

**PROGRAM STUDI TEKNIK KOMPUTER**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI PADANG**

**2025**

# **KATA PENGANTAR**

Puji syukur kehadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, penulis dapat menyelesaikan makalah ini dengan judul “Keamanan dalam Arsitektur Microservices”. Makalah ini disusun untuk memberikan pemahaman mendalam mengenai tantangan, strategi, dan praktik terbaik dalam mengamankan sistem berbasis microservices, yang semakin populer di era pengembangan perangkat lunak modern.

Perkembangan teknologi yang pesat, terutama dalam arsitektur terdistribusi, membawa banyak keuntungan seperti skalabilitas dan fleksibilitas. Namun, di sisi lain, pendekatan microservices juga menimbulkan kompleksitas keamanan yang harus diatasi dengan pendekatan yang tepat. Melalui makalah ini, penulis berharap pembaca dapat memahami risiko keamanan yang mungkin muncul serta solusi yang dapat diterapkan untuk memitigasinya.

Penulis menyadari bahwa penyusunan makalah ini tidak terlepas dari dukungan berbagai pihak. Ucapan terima kasih disampaikan kepada, Dosen dan rekan-rekan yang memberikan masukan berharga dalam penyempurnaan makalah ini. Berbagai literatur, penelitian, dan sumber online yang menjadi referensi dalam penulisan. Semua pihak yang telah mendukung secara langsung maupun tidak langsung.

Penulis menyadari bahwa makalah ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang. Semoga makalah ini dapat bermanfaat bagi pembaca, khususnya bagi para pengembang, arsitek sistem, dan profesional keamanan siber yang tertarik dalam mengimplementasikan microservices secara aman.

|  |
| --- |
| Padang, April 2025 |
| Syfa Aulia |

**DAFTAR ISI**

Contents

[**KATA PENGANTAR** i](#_Toc195022989)

[**BAB I** 1](#_Toc195022990)

[**PENDAHULUAN** 1](#_Toc195022991)

[**1.1** **Latar Belakang** 1](#_Toc195022992)

[**1.2** **Rumusan Masalah** 1](#_Toc195022993)

[**1.3** **Tujuan dan Manfaat** 2](#_Toc195022994)

[**BAB II** 3](#_Toc195022995)

[**PEMBAHASAN** 3](#_Toc195022996)

[**2.1** **Tentang Keamanan dalam Microservice** 3](#_Toc195022997)

[A. Peningkatan Attack Surface 3](#_Toc195022998)

[B. Manajemen Autentikasi dan Otorisasi 3](#_Toc195022999)

[C. Keamanan Komunikasi Antar-Layanan 3](#_Toc195023000)

[D. Keamanan API Gateway 3](#_Toc195023001)

[E. Manajemen Rahasia (Secrets Management) 4](#_Toc195023002)

[**2.2** **Strategi Keamanan untuk Microservices** 4](#_Toc195023003)

[A. Implementasi Zero Trust Architecture 4](#_Toc195023004)

[B. Penggunaan OAuth2 dan JWT 4](#_Toc195023005)

[C. Enkripsi Komunikasi dengan TLS/Mtls 4](#_Toc195023006)

[D. Penggunaan Service Mesh 4](#_Toc195023007)

[E. Pembatasan Akses dengan Role-Based Access Control (RBAC) 5](#_Toc195023008)

[F. Pengelolaan Rahasia yang Aman 5](#_Toc195023009)

[G. Penerapan API Security Best Practices 5](#_Toc195023010)

[**2.3** **Praktik Terbaik untuk Keamanan Microservices** 5](#_Toc195023011)

[**BAB III** 6](#_Toc195023012)

[**PENUTUP** 6](#_Toc195023013)

[**3.1** **Kesimpulan** 6](#_Toc195023014)

[**3.2** **Saran** 6](#_Toc195023015)

[**DAFTAR PUSTAKA** 7](#_Toc195023016)

# **BAB I**

# **PENDAHULUAN**

## **Latar Belakang**

Dalam beberapa tahun terakhir, arsitektur microservices telah menjadi populer karena kemampuannya untuk meningkatkan skalabilitas, kecepatan pengembangan, dan pemeliharaan sistem. Namun, pendekatan terdistribusi ini juga memperkenalkan tantangan keamanan baru, seperti manajemen identitas yang kompleks, peningkatan permukaan serangan (attack surface), dan kesulitan dalam menerapkan kebijakan keamanan yang konsisten.

Keamanan dalam microservices memerlukan pendekatan yang berbeda dibandingkan dengan arsitektur monolitik tradisional. Setiap layanan harus diamankan secara individual, sementara komunikasi antar-layanan juga harus dilindungi dari ancaman seperti Man-in-the-Middle (MITM), injeksi, dan eksploitasi API.

Oleh karena itu, penting untuk merancang strategi keamanan yang holistik dan terintegrasi, mencakup otentikasi, otorisasi, enkripsi, logging, serta pengawasan trafik layanan. Penanganan keamanan dalam microservices bukan hanya tanggung jawab tim keamanan, tetapi memerlukan kolaborasi erat antara pengembang, DevOps, dan arsitek sistem. Studi ini bertujuan untuk membahas tantangan utama dalam keamanan arsitektur microservices serta pendekatan dan solusi yang dapat diterapkan untuk memitigasi risiko-risiko tersebut secara efektif.

## **Rumusan Masalah**

Berdasarkan latar belakang tersebut, dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Apa saja tantangan utama dalam penerapan keamanan pada arsitektur microservices?
2. Bagaimana pendekatan terbaik untuk mengelola keamanan komunikasi antar-layanan dalam microservices?
3. Teknologi atau metode apa yang dapat diterapkan untuk memastikan keamanan yang konsisten di seluruh layanan?

## **Tujuan dan Manfaat**

1. Mengidentifikasi dan menjelaskan tantangan keamanan yang dihadapi dalam arsitektur microservices.
2. Menganalisis pendekatan dan teknologi yang dapat digunakan untuk mengatasi tantangan tersebut.
3. Memberikan rekomendasi implementasi keamanan yang tepat dalam sistem microservices.
4. Memberikan wawasan kepada pengembang dan arsitek sistem mengenai pentingnya penerapan keamanan yang menyeluruh dalam microservices.
5. Menjadi referensi dalam merancang arsitektur sistem yang aman dan handal.
6. Meningkatkan kesadaran akan ancaman siber yang dapat muncul akibat kurangnya pengelolaan keamanan dalam sistem terdistribusi.

# **BAB II**

# **PEMBAHASAN**

# **Tentang Keamanan dalam Microservice**

Arsitektur microservices memperkenalkan pendekatan terdistribusi yang sangat fleksibel, namun di sisi lain juga membuka berbagai tantangan dalam hal keamanan. Beberapa tantangan utama tersebut antara lain:

### Peningkatan Attack Surface

Setiap microservice yang terpapar ke jaringan akan menambah potensi titik serangan. Serangan seperti DDoS (Distributed Denial of Service), injeksi SQL, atau eksploitasi API dapat terjadi jika layanan tidak dikonfigurasi dan diamankan dengan baik. Lingkungan yang kompleks juga menyulitkan deteksi serta mitigasi serangan secara cepat dan akurat.

### Manajemen Autentikasi dan Otorisasi

Dalam sistem microservices, setiap layanan harus mampu memverifikasi identitas pengguna (autentikasi) dan menentukan hak akses (otorisasi). Hal ini menuntut konsistensi penerapan protokol keamanan seperti OAuth2, OpenID Connect (OIDC), dan JWT. Ketidakkonsistenan dapat menciptakan celah yang membahayakan keamanan sistem secara keseluruhan.

### Keamanan Komunikasi Antar-Layanan

Komunikasi antar microservice yang dilakukan melalui jaringan internal maupun eksternal berpotensi disadap. Oleh karena itu, komunikasi antar-layanan harus dienkripsi menggunakan TLS (Transport Layer Security) atau mutual TLS (mTLS) untuk menjamin kerahasiaan dan integritas data.

### Keamanan API Gateway

API Gateway menjadi pintu gerbang utama bagi semua permintaan ke sistem microservices. Jika tidak diamankan dengan baik, maka gateway bisa menjadi titik lemah yang mudah diserang, seperti melalui injeksi API, brute-force, atau eksploitasi kelemahan autentikasi.

### Manajemen Rahasia (Secrets Management)

Microservices memerlukan akses terhadap rahasia seperti kredensial database, token API, atau kunci enkripsi. Penyimpanan rahasia yang tidak aman (misalnya dalam file konfigurasi) dapat menyebabkan kebocoran data dan akses tidak sah terhadap layanan penting.

## **Strategi Keamanan untuk Microservices**

Untuk menghadapi tantangan-tantangan tersebut, dibutuhkan strategi keamanan yang menyeluruh, terstandarisasi, dan berlapis. Beberapa strategi yang umum diterapkan meliputi:

### Implementasi Zero Trust Architecture

Prinsip “never trust, always verify” diterapkan dengan memastikan bahwa setiap permintaan—baik dari dalam maupun luar sistem—harus diverifikasi secara menyeluruh. Tidak ada entitas yang dianggap tepercaya secara otomatis.

### Penggunaan OAuth2 dan JWT

* *OAuth2* digunakan sebagai protokol otorisasi yang memungkinkan kontrol akses terpusat.
* *JWT (JSON Web Token)* memungkinkan penyimpanan klaim identitas pengguna yang dapat diverifikasi oleh setiap layanan, tanpa memerlukan otentikasi ulang.

### Enkripsi Komunikasi dengan TLS/Mtls

* *LS* digunakan untuk mengamankan koneksi antara klien dan layanan.
* *mTLS* menyediakan verifikasi dua arah antara klien dan server, menambah lapisan kepercayaan dalam komunikasi antar microservices.

### Penggunaan Service Mesh

Platform seperti *Istio* atau *Linkerd* menyediakan lapisan tambahan untuk:

* Enkripsi otomatis antar layanan dengan *mTLS*.
* Pengaturan lalu lintas berbasis kebijakan.
* Monitoring, tracing, dan logging untuk analisis ancaman.

### Pembatasan Akses dengan Role-Based Access Control (RBAC)

Setiap layanan harus menerapkan pembatasan akses berdasarkan peran pengguna (role), guna membatasi siapa yang dapat melakukan tindakan tertentu terhadap suatu layanan atau sumber daya.

### Pengelolaan Rahasia yang Aman

Gunakan tools khusus seperti *HashiCorp Vault, AWS Secrets Manager*, atau *Kubernetes Secrets* untuk mengelola rahasia secara terpusat dan aman, menghindari penyimpanan rahasia secara hardcoded.

### Penerapan API Security Best Practices

* Validasi input untuk mencegah injeksi seperti SQLi atau XSS.
* Terapkan *rate limiting* untuk mencegah overload dan DDoS.
* Gunakan API Gateway yang dilengkapi dengan fitur *Web Application**Firewall (WAF)* untuk deteksi dan perlindungan terhadap serangan umum.

## **Praktik Terbaik untuk Keamanan Microservices**

Selain strategi-strategi di atas, terdapat pula sejumlah praktik terbaik yang dapat diterapkan untuk meningkatkan keamanan microservices secara menyeluruh:

* Hardening Layanan:

Nonaktifkan fitur yang tidak digunakan, gunakan konfigurasi minimalis, dan lakukan patching secara rutin.

* Monitoring dan Audit:

Gunakan sistem SIEM (Security Information and Event Management) untuk pemantauan dan deteksi dini terhadap aktivitas mencurigakan.

* Container Security:

Pastikan image container bersih dari kerentanan, gunakan registry tepercaya, dan aktifkan scanning otomatis terhadap image sebelum deployment.

* DevSecOps:

Integrasikan aspek keamanan dalam pipeline CI/CD dengan pemeriksaan otomatis melalui tools SAST (Static Application Security Testing) dan DAST (Dynamic Application Security Testing).

# **BAB III**

# **PENUTUP**

## **Kesimpulan**

Arsitektur microservices memberikan berbagai keunggulan seperti fleksibilitas, skalabilitas, dan kemudahan dalam proses pengembangan serta pemeliharaan sistem. Namun, keuntungan tersebut juga disertai dengan tantangan-tantangan keamanan yang signifikan. Permukaan serangan yang lebih luas, kompleksitas dalam manajemen identitas, serta kebutuhan akan komunikasi yang aman antar-layanan menjadi aspek penting yang harus dihadapi secara serius.

Untuk mengatasi berbagai tantangan tersebut, strategi keamanan yang komprehensif harus diterapkan, antara lain melalui pendekatan Zero Trust Architecture, penggunaan protokol OAuth2 dan JWT untuk autentikasi dan otorisasi, serta penerapan enkripsi komunikasi menggunakan TLS atau mTLS. Selain itu, tools seperti service mesh, secrets manager, dan API gateway dengan fitur keamanan yang memadai dapat memperkuat perlindungan sistem.

Keamanan dalam microservices tidak dapat dicapai hanya dengan alat atau konfigurasi teknis semata, melainkan memerlukan budaya DevSecOps yang mendorong integrasi keamanan sejak tahap pengembangan. Dengan kombinasi teknologi, strategi, dan kolaborasi tim yang tepat, sistem berbasis microservices dapat dibangun dengan tingkat keamanan yang tinggi dan tahan terhadap berbagai potensi ancaman.

## **Saran**

Dalam merancang dan mengimplementasikan arsitektur microservices, pengembang dan tim keamanan perlu berkolaborasi sejak awal proses pengembangan untuk memastikan bahwa aspek keamanan tidak diabaikan. Penggunaan best practices seperti hardening layanan, audit keamanan berkala, serta penggunaan container yang aman harus menjadi standar operasional.

Penting juga untuk melakukan pelatihan dan peningkatan kapasitas sumber daya manusia di bidang keamanan siber agar setiap anggota tim memiliki pemahaman yang baik mengenai risiko dan mitigasinya. Terakhir, disarankan untuk terus mengikuti perkembangan teknologi dan metode keamanan terbaru guna memastikan bahwa sistem microservices tetap terlindungi dari ancaman yang semakin kompleks di masa depan.

# **DAFTAR PUSTAKA**

Nasir, Arnold. "Analisa Celah Kelah Keamanan Terhadap Web Server Menggunakan Metode Attack Surface Dan Kepadatan Kerentanan." *TEMATIKA: Jurnal Penelitian Teknik Informatika dan Sistem Informasi* (2020): 67-72.

Newman, Sam. *Building microservices: designing fine-grained systems*. " O'Reilly Media, Inc.", 2021.

Karmel, Anil, Ramaswamy Chandramouli, and Michaela Iorga. "NIST Special Publication 800-180 (DRAFT)."

Asemi, Hadi. *A Study On API Security Pentesting*. MS thesis. California Polytechnic State University, 2023.

Calcote, Lee, and Zack Butcher. *Istio: Up and running: Using a service mesh to connect, secure, control, and observe*. O'Reilly Media, 2019.