

No _____
Date _____

Nama : Krisnayanti kesia Patanda

NIM : 231401008

Kelas : Angkatan 23

tujuan dan tujuan pokok dalam mendidik dan memelihara manusia agar menjadi orang yang berguna bagi bangsa dan negara. Tujuan pendidikan diantaranya adalah untuk menciptakan manusia yang berkarakter, berilmu pengetahuan dan teknologi, berdaya saing, berwibawa, berjiwa nasionalisme, berpatriotisme, beriman dan bertakwa, berakhlak mulia, berdisiplin, berpribadi dan berwirausaha. Tujuan pokok pendidikan diantaranya adalah untuk menciptakan manusia yang berkarakter, berilmu pengetahuan dan teknologi, berdaya saing, berwibawa, berjiwa nasionalisme, berpatriotisme, beriman dan bertakwa, berakhlak mulia, berdisiplin, berpribadi dan berwirausaha.

Architects Daughter Aws IoT, Google Cloud IoT

1. Pendahuluan (latar belakang)

Perkembangan teknologi Internet of Things (IoT) telah mendorong perubahan besar dalam cara sistem jaringan dibangun dan dioperasikan. IoT tidak lagi hanya berbicara tentang aplikasi yang terlihat oleh pengguna, tetapi lebih pada bagaimana jutaan perangkat dengan sumber daya terbatas mampu berkomunikasi secara efisien, aman dan real-time melalui jaringan. Dalam konteks ini, pemrograman jaringan menjadi inti atau "nyawa" dari keseluruhan layanan IoT.

Dua platform cloud yang paling banyak digunakan dalam implementasi IoT skala besar adalah AWS IoT dan Google Cloud IoT. Keduanya menyediakan infrastruktur untuk menghubungkan perangkat, mengelola data, serta memproses informasi secara terdistribusi. Namun, kekuatan utama dari platform ini bukan hanya pada antarmuka atau fitur aplikasinya, melainkan pada desain arsitektur jaringan dan pemilihan protokol komunikasi yang tepat.

Salah satu protokol yang menjadi pondasi utama dalam AWS IoT dan Google Cloud IoT adalah Message Queuing Telemetry Transport (MQTT). MQTT dirancang sebagai protokol ringan untuk perangkat dengan keterbatasan bandwidth, memory dan daya. dengan menggunakan MQTT data dapat berpindah dari perangkat (Titik A)

- ke cloud (titik B) secara efisien melalui mekanisme Pemrograman Jaringan yang terstruktur.

2. Pembahasan Teknisi (Deep Dive network code)

a. profil produk / vendor

AWS IoT merupakan layanan Cloud dari Amazon Web Services yang memungkinkan Perangkat IoT berhubung, berkomunikasi, dan bertukar data secara aman. Google Cloud IoT menggunakan model memiliki fungsi serupa dengan fokus pada integritas data IoT ke dalam ekosistem google cloud. kedua Platform ini dirancang untuk menangani untuk menangani komunikasi Jaringan ber Skala besar dengan latensi rendah dan kredibilitas tinggi.

b. Arsitektur Jaringan.

Secara arsitektural AWS IoT dan google Cloud IoT menggunakan model Client-Server dengan Pendekatan microservices di Sisi Cloud. Perangkat IoT bertindak sebagai client, Sedangkan Cloud bertindak sebagai Server atau broker. dalam komunikasi tersebut, broker menjadi pusat distribusi Pesan, sehingga perangkat tidak berkomunikasi langsung satu sama lain.

Arsitektur ini mendukung Skalabilitas tinggi karena setiap Perangkat hanya perlu mengetahui alamat broker, bukan alamat perangkat lain. di Sisi Cloud, layanan dipecah menjadi microservices

untuk autentikasi, routing pesan, penyimpanan data dan analisis. Sehingga beban jaringan dapat dibayani secara paralel.

c. Protokol komunikasi

Protokol utama yang digunakan adalah MQTT yang berjalan di atas TCP/IP, sering kali dikombinasikan dengan TLS untuk keamanan. MQTT dipilih karena:

1. overhead paket sangat kecil
2. mendukung koneksi persisten
3. cocok untuk komunikasi real-time dan jaringan tidak stabil

di bandingkan HTTP yang bersifat request-response, MQTT menggunakan mekanisme Publish-Subscribe dalam kode jaringan, Client MQTT membuka socket TCP ke broker, lalu mempertahankan koneksi tersebut untuk pengiriman data berulang tanpa perlu handshake ulang.

d. Mekanisme Pemrograman Jaringan

Pada level pemrograman, proses perpindahan data terjadi sebagai berikut:

1. Client membuat socket TCP ke endpoint cloud.
2. Library MQTT melakukan autentikasi (sertifikat atau token)
3. data dikemas dalam payload kecil dan dikirim ke broker tertentu
4. Broker meneruskan data ke set subscriber yang relevan.

dalam kondisi high-load, broker MQTT mampu menangani ribuan koneksi simultan karena tidak perlu memproses request berat seperti HTTP. Untuk data real-time mekanisme keep-alive dan QoS memastikan pesan tetap terkirim sesuai tingkat keandalan yang diinginkan.

3. Analisis keunggulan (why is it good?)

Keunggulan utama penggunaan MQTT pada AWS IoT dan Google Cloud IoT terletak pada efisiensi jaringan dan kemudahan pengelolaan sistem. Dari sudut pandang pemrograman jaringan, MQTT menyederhanakan logika komunikasi karena client tidak perlu mengetahui detail infrastruktur cloud.

Selain itu, konsep publish-subscribe membuat sistem lebih fleksibel dan scalable. Penambahan perangkat baru tidak memerlukan perubahan besar pada kode jaringan yang sudah ada. Hal ini sangat penting dalam sistem IoT yang jumlah perangkatnya terus bertambah.

Dari sisi performa MQTT mampu bekerja dengan baik pada kondisi jaringan tidak stabil, seperti pada sensor jarak jauh. Ini membuktikan bahwa desain protokol yang tepat sangat menentukan keberhasilan layanan IoT secara keseluruhan.

4. kesimpulan (insight (why is it good))

AWS IoT dan google Cloud IoT bukan sekedar layanan Cloud biasa, melainkan sistem Jaringan kompleks yang dirancang dengan Prinsip efisiensi dan skalabilitas tinggi. Pemrograman Jaringan menjadi fondasi utama yang memungkinkan layanan ini berjalan secara real-time dan andal. dengan demikian, MQTT layak dianggap sebagai protokol inti dalam ekosistem IoT modern, dan pemahaman mendalam tentang Pemrograman Jaringan menjadi keterampilan penting bagi pengembangan teknologi masa depan.

⇒ kesimpulan (insight pribadi)

kekuatan MQTT bukan hanya pada ukurannya yang ringan, tetapi pada desain arsitekturnya yang sederhana namun sangat efektif. Sebagai seorang engineer, memahami bagaimana socket protokol dan broker bekerja memberikan sudut pandang baru bahwa aplikasi yang terlihat sederhana sebenarnya dibangun oleh mekanisme jaringan yang kompleks dan terstruktur.