Pertemuan 5

Internet of Things

Internet of Things

- Databases
- Security
- IoT Simulation

How to select the right IoT database architecture ?

- Untuk memilih arsitektur database terbaik untuk inisiatif IoT organisasi, teknolog IoT harus memahami dasar-dasar semua opsi database.

- Basis data IoT yang tepat bergantung pada persyaratan setiap proyek IoT.

- Teknologi IoT harus menentukan jenis data yang akan disimpan dan dikelola; aliran data; persyaratan fungsional untuk analitik, manajemen, dan keamanan; dan kinerja dan persyaratan bisnis.

- Setelah mengidentifikasi persyaratan organisasi untuk database, admin TI harus menilai arsitektur database IoT dan bagaimana mereka akan mempromosikan atau menghambat kebutuhan data IoT.

IoT Database Requirement

1 Scalability

Basis data untuk aplikasi IoT harus dapat diskalakan. Idealnya, database IoT dapat diskalakan secara linier sehingga menambahkan satu server lagi ke kluster 10-node meningkatkan throughput sebesar 10%. Basis data IoT biasanya akan didistribusikan kecuali aplikasi hanya mengumpulkan sejumlah kecil data yang tidak akan tumbuh secara substansial.

2 Fault Tolerance

Basis data IoT juga harus toleran terhadap kesalahan dan sangat tersedia. Jika sebuah node di cluster database sedang down, node tersebut seharusnya masih dapat menerima permintaan baca dan tulis

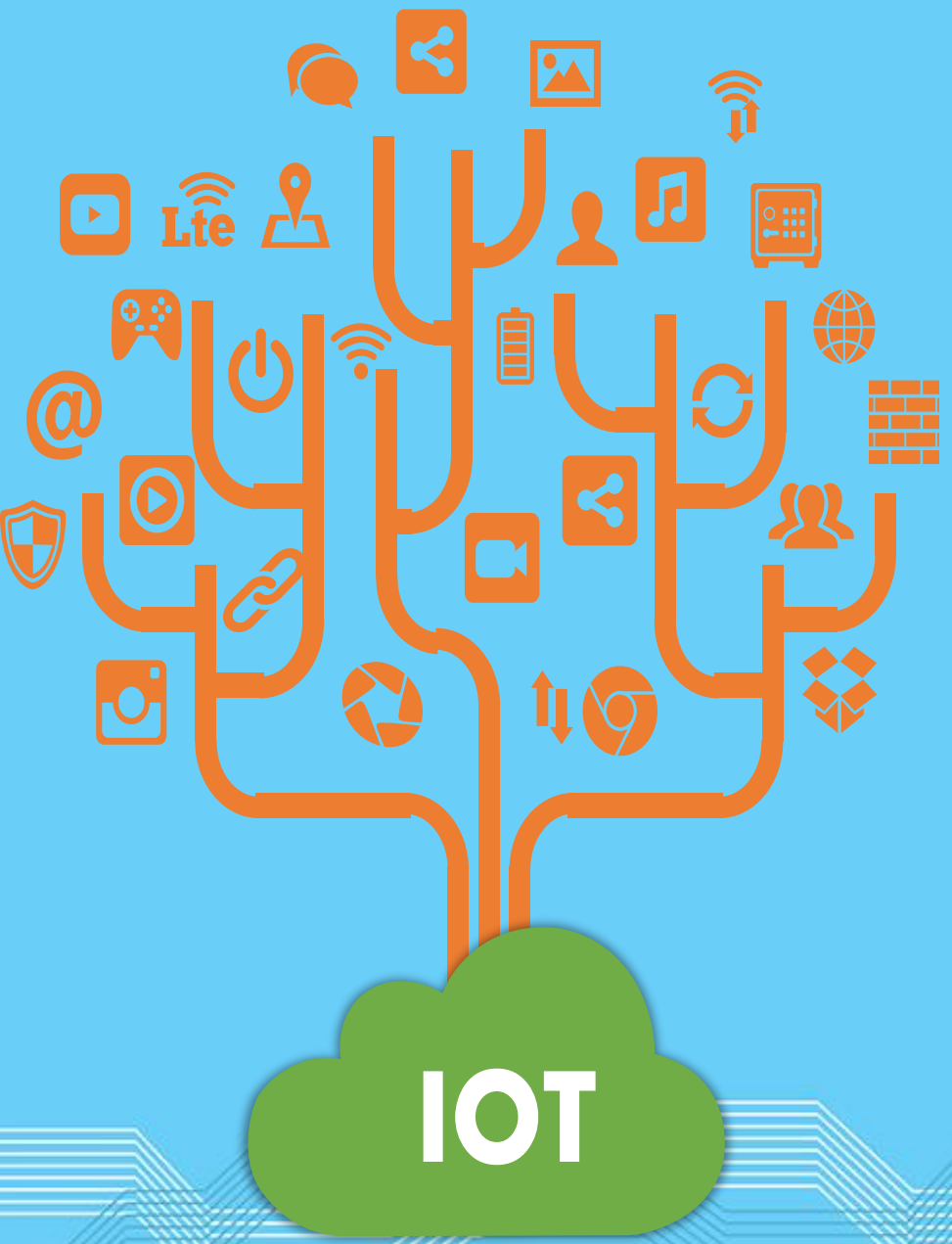
IoT Database Requirement

3 High Availability

Pastikan ketersediaan tinggi terkait penulisan dengan menggunakan sistem pesan terdistribusi seperti Apache Kafka atau Amazon Kinesis. Jika server sedang down atau volume penulisan terlalu tinggi untuk diserap database terdistribusi secara real time, data dapat disimpan dalam sistem pesan sampai database memproses backlog data atau node tambahan ditambahkan ke cluster database.

4 Flexibility

Basis data IoT harus sefleksibel yang dibutuhkan oleh aplikasi. Database NoSQL adalah pilihan yang baik ketika sebuah organisasi memiliki beberapa tipe data dan tipe data tersebut kemungkinan akan berubah seiring waktu.

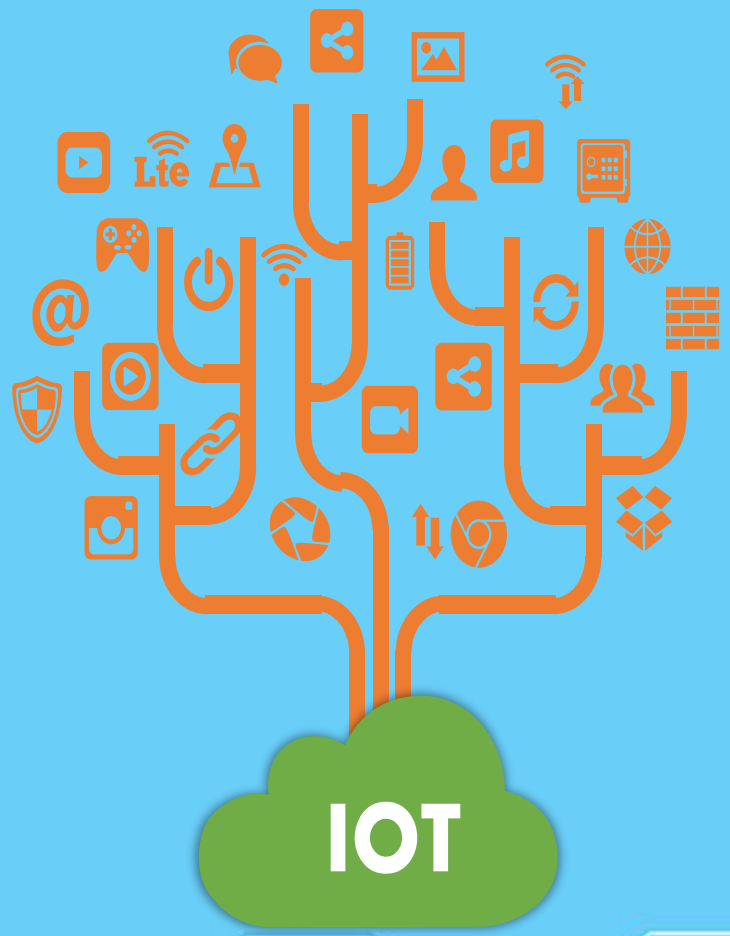


Static and Streaming IoT Database

Basis Data Statis

Basis data statis, juga dikenal sebagai basis data batch, mengelola data saat istirahat. Data yang perlu diakses pengguna berada sebagai data tersimpan yang dikelola oleh sistem manajemen basis data (DBMS). Pengguna membuat kueri dan menerima tanggapan dari DBMS, yang biasanya, tetapi tidak selalu, menggunakan SQL.

Static and Streaming IoT Database

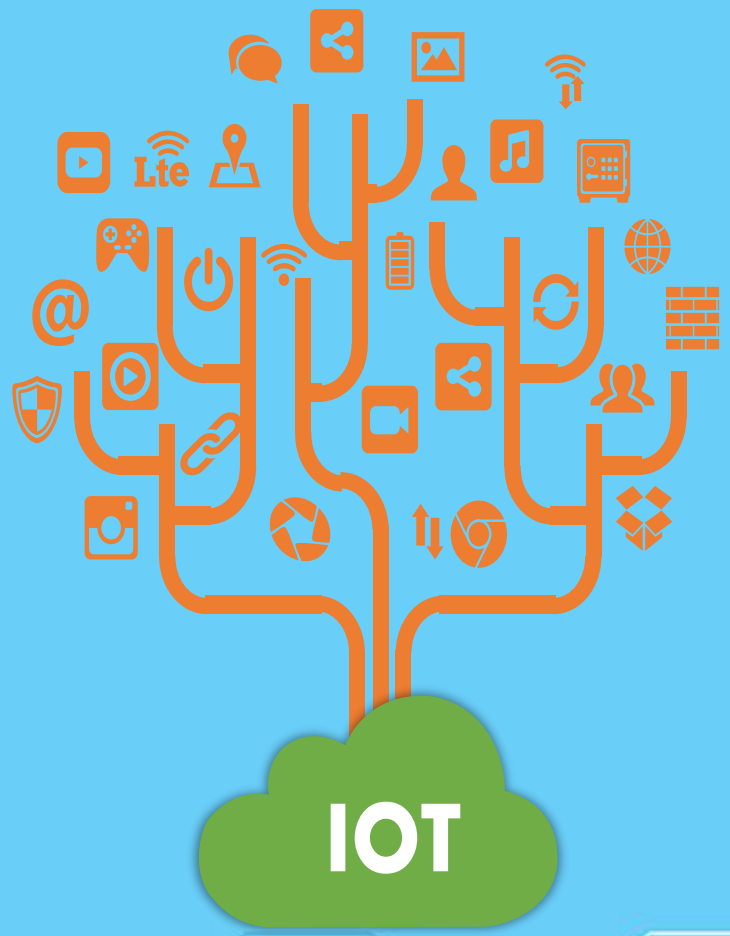


Basis data streaming menangani data yang bergerak. Keluaran database streaming pada akhirnya dapat disimpan di tempat lain, seperti di cloud, dan diakses melalui mekanisme kueri standar.

Basis data streaming biasanya didistribusikan untuk menangani skala dan persyaratan beban volume data yang sangat besar. Saat ini, ada berbagai database streaming komersial, berpemilik, dan open source, termasuk Google Cloud Dataflow, Microsoft StreamInsight, Azure Stream Analytics, IBM InfoSphere Streams, dan Amazon Kinesis.

Sistem open source sebagian besar berbasis di sekitar Apache dan termasuk Apache Spark Streaming yang disediakan oleh Databricks, Apache Flink yang disediakan oleh Data Artisans, Apache Kafka yang disediakan oleh Confluent dan Apache Storm, yang dimiliki oleh Twitter.

Which Database is the Best for IoT ?

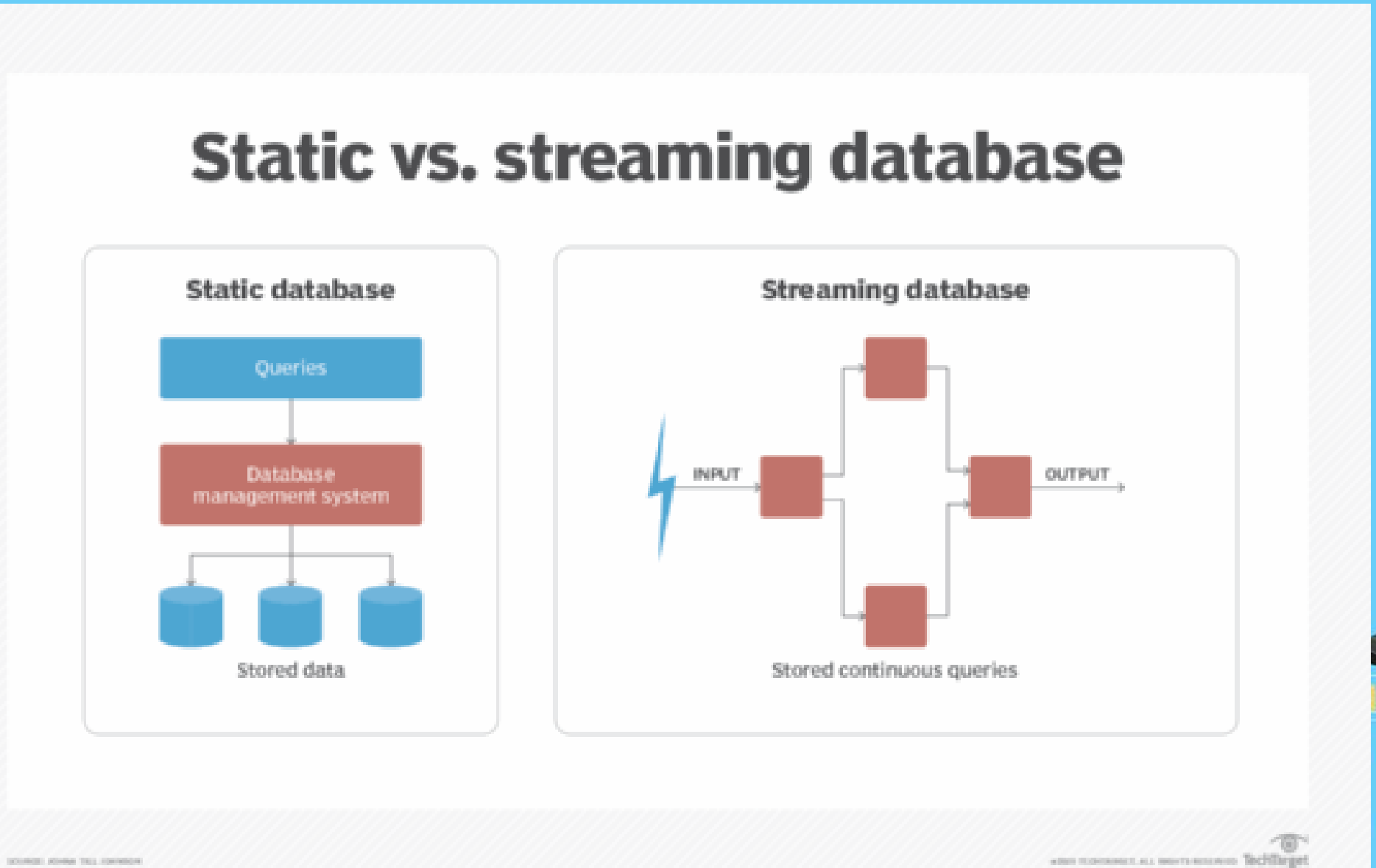


Pengembang IoT dapat mengambil manfaat dari teknik dan skema kueri standar, itulah sebabnya banyak database streaming juga menyertakan komponen database statis.

Basis data terpadu ini menggabungkan yang terbaik dari dunia streaming dan basis data statis, karena keduanya mendukung kemampuan real-time dari basis data streaming dan fleksibilitas proses dan skema kueri basis data statis.

Basis data terbaik untuk sebagian besar aplikasi IoT adalah basis data terpadu yang menggabungkan kemampuan streaming dan statis. Basis data vendor yang paling populer mencakup kedua jenis basis data karena alasan ini.

Static vs Streaming Database



SQL vs NoSQL ?

- Database SQL adalah skema statis relasional dan fitur yang menggambarkan bagaimana informasi diatur. Hal ini membuat mereka sangat mudah dikelola. Namun, mereka mengalami masalah penskalaan secara efektif.

- Basis data NoSQL bersifat nonrelasional, tidak memiliki skema, dan umumnya dipromosikan sebagai basis data yang sangat skalabel dan berkinerja lebih baik daripada basis data SQL.

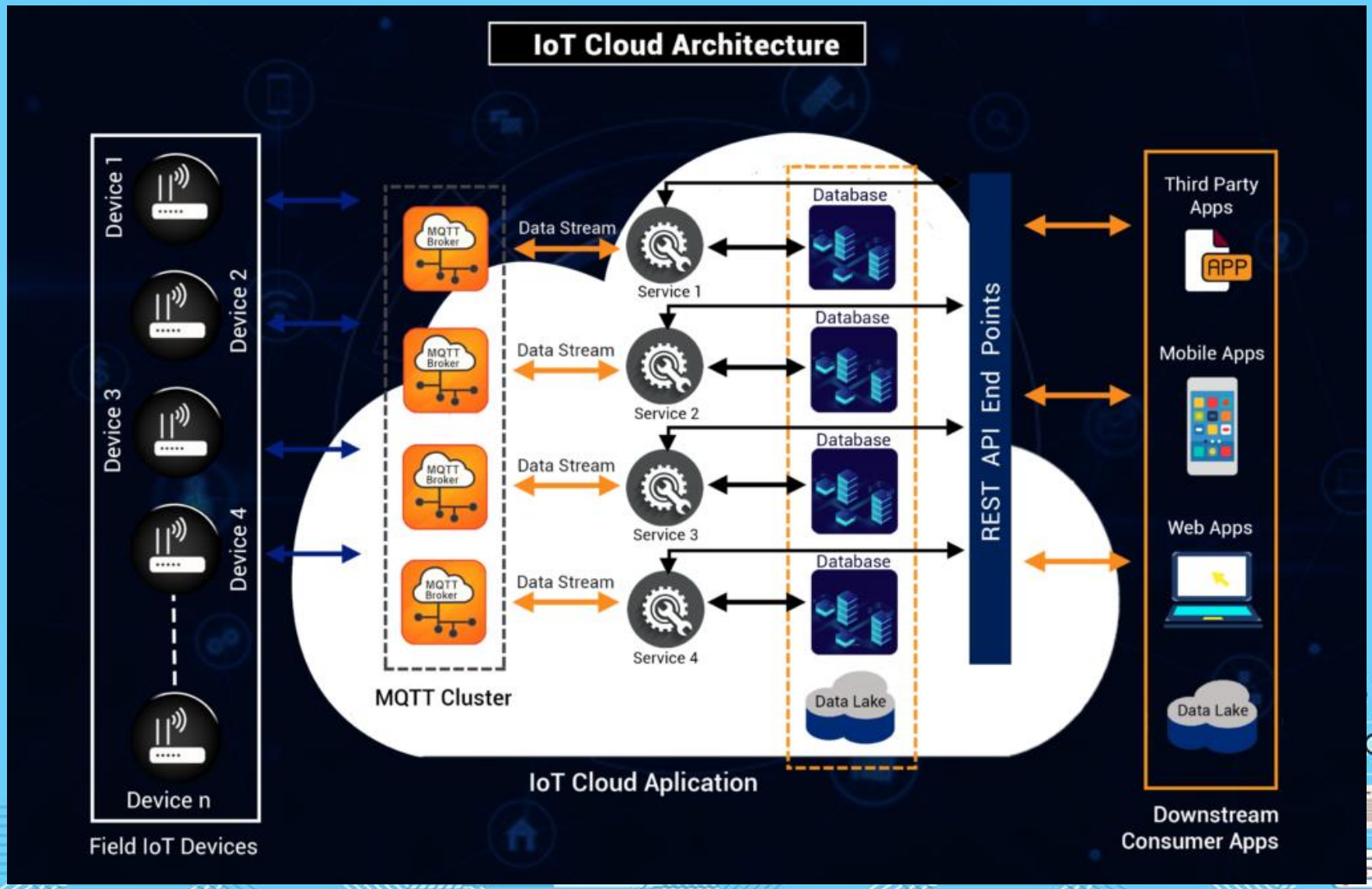
SQL vs NoSQL ?

- Organisasi mungkin juga merasa kesulitan menggabungkan database statis dan streaming saat menyertakan pilihan antara SQL dan NoSQL. Secara teori, database statis atau streaming bisa berupa SQL atau NoSQL. Dalam praktiknya, database secara khusus diatur ke satu atau yang lain. Para teknolog IoT yang tertarik pada database terpadu tertentu dapat menemukan keputusan SQL vs. NoSQL mereka didorong oleh desain database.

- Apakah suatu organisasi harus memilih database SQL atau NoSQL tergantung pada rangkaian persyaratan fungsional dan teknis yang lebih luas, terutama skalabilitas, kinerja, dan kebutuhan untuk berintegrasi ke dalam sistem lama.

IOT CLOUD ARCHITECTURE

IoT Cloud Architecture



IoT Cloud Architecture

- Perangkat IoT

Perangkat terintegrasi dengan sensor atau aktuator dan membuat koneksi dengan IoT Integration Middleware. Dalam beberapa kasus penggunaan, beberapa perangkat dapat dikelompokkan bersama dan dihubungkan ke infrastruktur gateway IoT yang mengirimkan data perangkat ke IoT Integration Middleware. Perangkat memiliki driver, yaitu perangkat lunak yang memungkinkan akses ke sensor/aktuator.

- Middleware Integrasi IoT

Middleware Integrasi IoT adalah lapisan integrasi untuk berbagai perangkat saat terhubung dengan cloud. Ini menerima data dari perangkat yang terhubung, memprosesnya dan mengirimkan informasi ini ke aplikasi hilir. Pemrosesan data dapat mencakup evaluasi aturan kondisi-tindakan, dan penyebaran perintah ke sensor perangkat berdasarkan evaluasi.

IoT Cloud Architecture

- Cloud Servers

Server adalah bagian terpenting dari cloud IoT, karena ini diperlukan untuk menyediakan layanan bisnis kepada pelanggan. Ini adalah mesin virtual yang terhubung ke database individu.

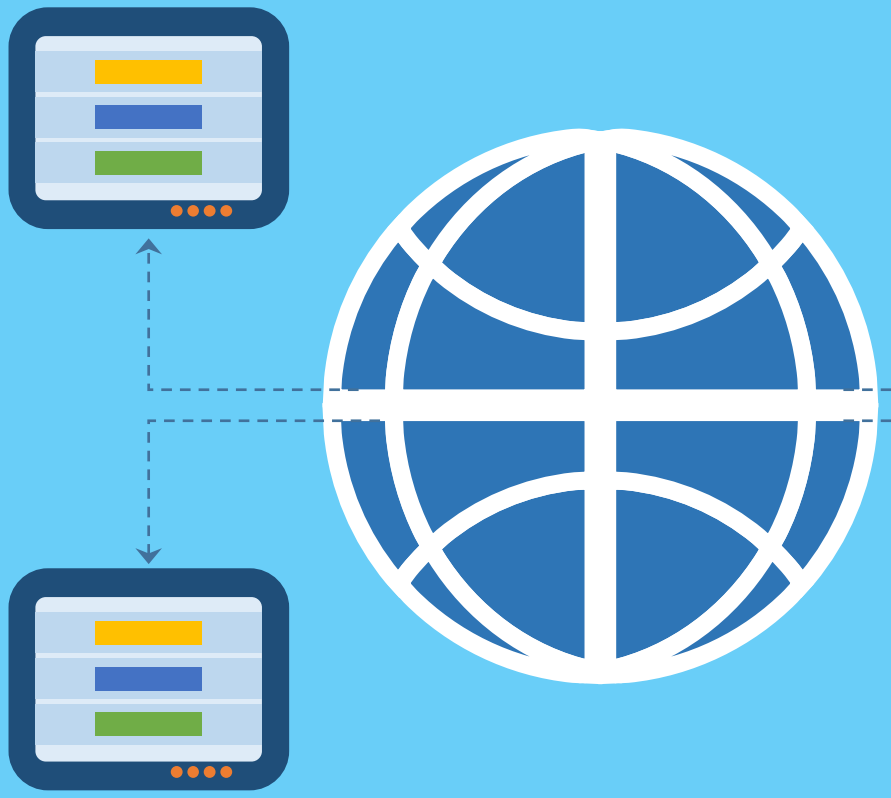
- Database

Berdasarkan persyaratan bisnis untuk penyimpanan dan pemrosesan data, database SQL dan No SQL dapat dikonfigurasi di cloud IoT. Database SQL menyimpan data dalam bentuk tabel dua dimensi. Kerugian utama dari jenis database ini adalah kinerjanya. Tidak ada database SQL yang jauh lebih efisien dan real-time.

- Downstream Applications/BI Tools

Server cloud terhubung ke aplikasi pihak ketiga, aplikasi seluler/web, atau alat intelijen bisnis melalui titik akhir REST API.

IoT Database Challenges



A

Data Volume and Variety

Organisasi sering mengumpulkan berbagai jenis data IoT yang harus dikelola bersama.

B

Scalability

Mungkin ada ribuan perangkat di lingkungan tertentu, masing-masing menjalankan instance database.

C

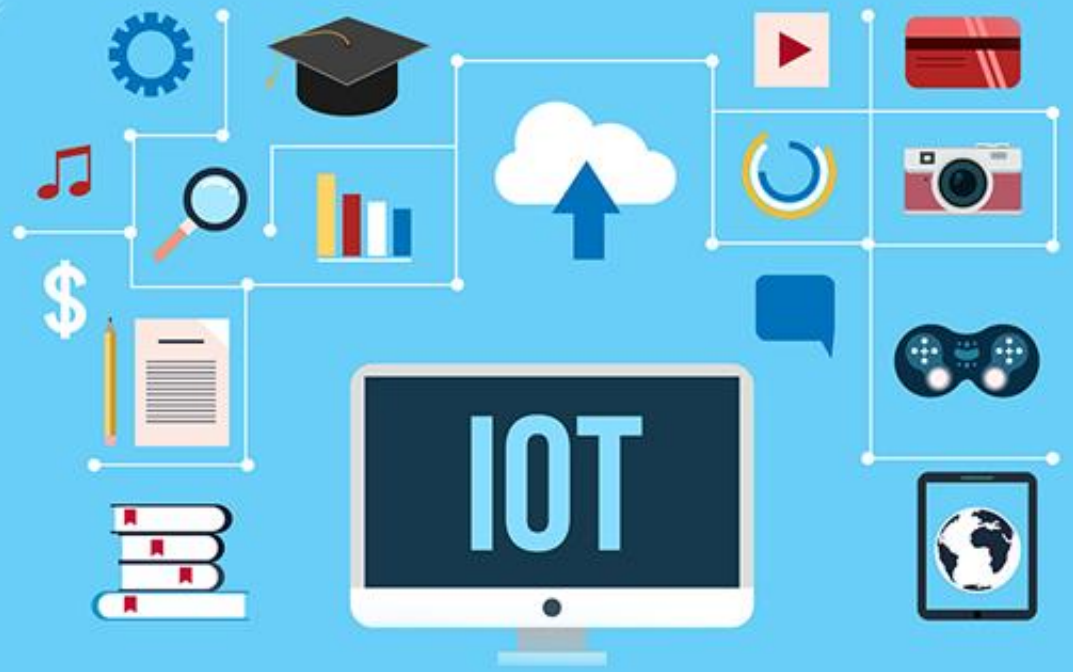
Time Series Data

Data sensor ditangkap di lokasi tertentu pada interval waktu tertentu.

D

High Performance

Aplikasi IoT sering kali menyertakan elemen waktu nyata atau hampir nyata yang memerlukan perangkat lunak basis data berkinerja tinggi.



SECURITY

The Most Important Security Problem in IoT

1

Incorrect Access Control

Layanan yang ditawarkan oleh perangkat IoT hanya boleh diakses oleh pemilik dan orang-orang di lingkungan terdekat yang mereka percayai. Perangkat IoT dapat memercayai jaringan lokal ke tingkat sedemikian rupa sehingga tidak diperlukan autentikasi atau otorisasi lebih lanjut. Masalah umum adalah bahwa semua perangkat dengan model yang sama dikirimkan dengan kata sandi default yang sama (mis. "admin" atau "password123").
Perangkat IoT sering kali memiliki satu akun atau tingkat hak istimewa, baik yang diekspos ke pengguna maupun secara internal. Artinya, ketika hak istimewa ini diperoleh, tidak ada kontrol akses lebih lanjut. Tingkat perlindungan tunggal ini gagal melindungi dari beberapa kerentanan.

The Most Important Security Problem in IoT

2 Overly Large Attack Surface

Setiap koneksi yang dapat dibuat ke sistem memberikan serangkaian peluang baru bagi penyerang untuk menemukan dan mengeksploitasi kerentanan. Semakin banyak layanan yang ditawarkan perangkat melalui Internet, semakin banyak layanan yang dapat diserang. Ini dikenal sebagai Surface Attack

3 Outdated Software

Software yang out of date biasanya rentan terkena bug. Ini berarti bahwa perangkat IoT harus dikirimkan dengan perangkat lunak terbaru tanpa kerentanan yang diketahui, dan perangkat tersebut harus memiliki fungsionalitas pembaruan untuk memperbaiki setiap kerentanan yang diketahui.

The Most Important Security Problem in IoT

4

Lack of Encryption

Masalah umum dalam kategori ini adalah menggunakan versi teks biasa dari protokol (misalnya HTTP) di mana versi terenkripsi tersedia (HTTPS). Serangan Man-in-the-Middle di mana penyerang secara diam-diam mengakses, dan kemudian menyampaikan komunikasi, mungkin mengubah komunikasi ini, tanpa disadari oleh salah satu pihak.

Bahkan ketika data dienkripsi, kelemahan mungkin ada jika enkripsi tidak lengkap atau tidak dikonfigurasi dengan benar. Misalnya, perangkat mungkin gagal memverifikasi keaslian pihak lain. Meskipun koneksi dienkripsi, itu dapat dicegat oleh penyerang Man-in-the-Middle.

Data sensitif yang disimpan di perangkat (tidak aktif) juga harus dilindungi dengan enkripsi. Masalah lainnya adalah penggunaan algoritma kriptografi yang lemah atau penggunaan algoritma kriptografi dengan cara yang tidak diinginkan.

The Most Important Security Problem in IoT

5 Application Vulnerabilities

Bug perangkat lunak memungkinkan untuk memicu fungsionalitas di perangkat yang tidak dimaksudkan oleh pengembang. Dalam beberapa kasus, ini dapat mengakibatkan penyerang menjalankan kode mereka sendiri di perangkat, sehingga memungkinkan untuk mengekstrak informasi sensitif atau menyerang pihak lain.

6 Lack of Trusted Execution Environment

Sebagian besar perangkat IoT secara efektif merupakan komputer serba guna yang dapat menjalankan perangkat lunak tertentu. Ini memungkinkan penyerang untuk menginstal perangkat lunak mereka sendiri yang memiliki fungsi yang bukan merupakan bagian dari fungsi normal perangkat. Dengan membatasi fungsionalitas perangkat dan perangkat lunak yang dapat dijalankannya, kemungkinan penyalahgunaan perangkat menjadi terbatas

The Most Important Security Problem in IoT

7 Vendor Security Posture

Bug perangkat lunak memungkinkan untuk memicu fungsionalitas di perangkat yang tidak dimaksudkan oleh pengembang. Dalam beberapa kasus, ini dapat mengakibatkan penyerang menjalankan kode mereka sendiri di perangkat, sehingga memungkinkan untuk mengekstrak informasi sensitif atau menyerang pihak lain.

8 Insufficient Privacy Protection

Perangkat konsumen biasanya menyimpan informasi sensitif. Perangkat IoT dan layanan terkait harus menangani informasi sensitif dengan benar, aman, dan hanya setelah persetujuan pengguna akhir perangkat. Ini berlaku untuk penyimpanan dan distribusi informasi sensitif.

The Most Important Security Problem in IoT

9 Intrusion ignore

Ketika perangkat disusupi, sering kali tetap berfungsi normal dari sudut pandang pengguna. Sebagian besar perangkat tidak memiliki fungsi pencatatan atau peringatan untuk memberi tahu pengguna tentang masalah keamanan apa pun. Jika ada, ini dapat ditimpa atau dinonaktifkan saat perangkat diretas. Akibatnya, pengguna jarang mengetahui bahwa perangkat mereka sedang diserang atau telah disusupi, sehingga mencegah mereka mengambil tindakan mitigasi.

10 Insufficient Physical Security

Jika penyerang memiliki akses fisik ke perangkat, mereka dapat membuka perangkat dan menyerang perangkat keras. Misalnya, dengan membaca konten komponen memori secara langsung, perangkat lunak pelindung apa pun dapat dilewati. Selain itu, perangkat mungkin memiliki kontak debug, yang dapat diakses setelah membuka perangkat, yang memberikan kemungkinan tambahan bagi penyerang.

The Most Important Security Problem in IoT

11

User Interaction

Interaksi pengguna adalah kategori penting untuk memastikan langkah-langkah keamanan yang diterapkan diaktifkan dan digunakan dengan benar. Jika mungkin untuk mengubah kata sandi default, tetapi pengguna tidak tahu atau tidak dapat menemukan fungsinya, itu tidak berguna.

Sumber

- https://www.embitel.com/iot-insights/iot-cloud-architecture-insights-why-database-design-matters
- https://internetofthingsagenda.techtarget.com/tip/How-to-select-the-right-IoT-database-architecture
- https://internetofthingsagenda.techtarget.com/feature/Find-the-IoT-database-that-best-fits-your-enterprises-needs
- https://www.mssqltips.com/sqlservertip/5915/sql-server-iot-database-design-example/
- https://www.eurofins-cybersecurity.com/news/security-problems-iot-devices/

Modul 5

A. Tujuan

Mahasiswa mampu memahami arsitektur sistem IoT :

- Protocols Concepts
- IoT-oriented Protocols
- Databases
- Security

B. Dasar teori

1. Database

Database digunakan untuk penyimpanan data jangka pendek maupun jangka panjang. Selain untuk keperluan analisis, aplikasi IoT mungkin bergantung pada database untuk mengambil data dalam jangka waktu tertentu. Terdapat 3 database yang sering digunakan dalam pembuatan aplikasi IoT :

- NoSQL. Merupakan basis data tidak terstruktur dan digunakan di banyak sistem IoT. Database noSQL cukup sederhana karena tidak memiliki skema. Basis data noSQL menyimpan data dengan merepresentasikan data sebagai pasangan nilai kunci. Keuntungan menggunakan basis data noSQL adalah penyebaran data yang cepat walaupun dapat menyebabkan beberapa masalah ketika pemeliharaan.
- Amazon Simple Storage Service (Amazon S3) (https://aws.amazon.com/s3/) merupakan penyimpanan objek dengan antarmuka layanan Web. Data dapat dimasukkan ke layanan penyimpanan lain yang lebih murah jika data digunakan untuk jangka panjang ataupun penggunaan data yang jarang. Notifikasi dapat dikeluarkan ketika objek dioperasikan.
- Google Cloud Storage (https://cloud.google.com/storage) adalah penyimpanan objek untuk data tidak terstruktur. Cloud Storage menyediakan tiga model layanan yang berbeda pada titik latensi/latensi/harga yang berbeda. Cloud SQL dapat digunakan untuk melakukan operasi database. Transfer streaming didukung menggunakan pengkodean HTTP.

2. Security

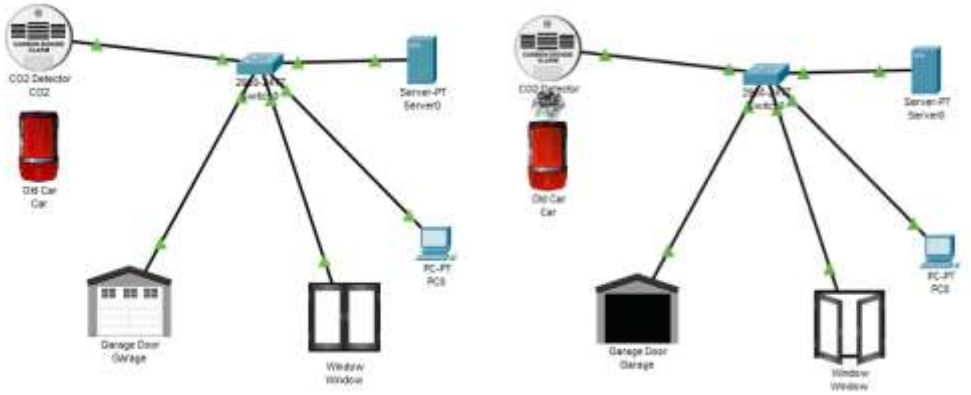
Keamanan adalah properti sistem; sistem hanya bisa seaman komponen terlemahnya. Fitur keamanan disediakan oleh komponen di beberapa lapisan dalam tumpukan IoT: perangkat, jaringan fisik, dan middleware.

Berikut adalah beberapa masalah keamanan yang ada pada sistem IoT:

- Incorrect Access Control
- Overly Large Attack Surface
- Outdated Software
- Lack of Encryption
- Application Vulnerabilities
- Lack of Trusted Execution Environment
- Vendor Security Posture
- Insufficient Privacy Protection
- Intrusion Ignore
- Insufficient Physical Security
- User Interaction

C. Contoh Program

1. Registration Server



Registration Server Login

Username:
Password:

Sign In

Don't have an IoE account?Sign up now

Sitem bekerja dengan mendeteksi kadar gas CO2. Ketika CO2 yang dideteksi melebihi ambang batas tertentu, maka pintu garasi dan jendela akan terbuka. Akses sistem dilakukan

menggunakan web browser dengan user dan password yang sudah ditentukan. Contoh ada di video.

D. TUGAS INDIVIDU

1. Membuat simulasi arsitektur IoT menggunakan Registration Server. User dan Password bebas, boleh di edit sendiri. Sistem yang dibuat dengan tema Smart Home (tidak boleh sama dengan contoh). Gunakan minimal 2 device, sensor, end device, 1 server.

2. Pengumpulan Tugas Praktikum.

- Kode program, penjelasan program, dan screenshot arsitektur IoT diupload ke akun github masing-masing. Kemudian link github dikumpulkan di SPADA.
- **Untuk kelas TI D pengumpulan paling lambat tanggal 26 September 2021 jam 23.59**
- **Untuk kelas TI E pengumpulan paling lambat tanggal 28 Semptember 2021 jam 23.59**
- Format penamaan file SKD_namakelas_nim_nama
- Contoh : https://github.com/fadilrahman46/IoT