

# **Internet of Things (IoT): A vision, architectural elements, and future directions**

## **Sub-Bab:**

### ***7.9 Cloud Computing***

Nama: Ahmad Fitri Rashad

Kelas: SK7A

NIM: 09121001023

Pada sub-bab 7.9 dengan judul *cloud computing*, sub-bab tersebut menjelaskan bagaimana jika IoT (Internet of Things) di integrasi dengan *cloud computing* untuk memungkinkan terciptanya sebuah Kota Pintar (*Smart Cities*). Namun, jika melakukan integrasi, hal yang patut diperhatikan adalah:

- Mengkombinasikan layanan yang ditawarkan dari berbagai orang yang berwenang memegang kekuasaan / saham, dan;
- Mendukung untuk dengan angka skala yang besar pada pengguna agar dapat bisa diandalkan dan perlakuan secara desentralisasi.

Selain itu, dua hal tersebut harus bisa dioperasikan dengan perantara kabel maupun dengan nirkabel (wireless), serta bisa menghadapi kendala seperti mengakses sebuah alat atau sumber data dengan tenaga yang terbatas dan koneksi yang tidak bisa diandalkan. Aplikasi platform *cloud* tersebut harus ditingkatkan untuk mendukung:

- Penciptaan aplikasi secara cepat dari sebuah domain yang menyediakan alat spesifik untuk membuat sebuah program dan lingkungan, serta;
- Mengeksekusi sebuah aplikasi secara mulus (seamless) dengan memanfaatkan kemampuan dari sumber yang banyak secara dinamis dan heterogen untuk mendapatkan persyaratan layanan yang berkualitas dari beberapa pengguna.

Sumber manajemen *cloud* dan sistem penjadwalan harus bisa secara dinamis memprioritas dan ketentuan sumber seperti permintaan yang kritis yang harus dilaksanakan secara waktu yang nyata. Untuk mendapatkan hasil yang dapat diandalkan, penjadwalan tersebut harus

ditambah dengan menduplikasi tugas algoritma untuk kegagalan manajemen. Khususnya, penjadwalan algoritma pada aplikasi *cloud* harus mencakupi kemampuan seperti berikut:

- Optimisasi secara multi-objektif: algoritma penjadwalan harus bisa menghadapi dengan parameter QoS (Quality of Service) seperti waktu respon, biaya dari layanan yang digunakan, angka harga per unit maksimum pada sumber daya yang tersedia, dan hukuman untuk layanan degradasi.
- Menduplikasi tugas berdasarkan kesalahan yang ditoleransi: tugas yang kritikal dari sebuah aplikasi akan di replikasi secara transparan dan dieksekusi berdasarkan sumber daya yang berbeda, jadi jika salah satu sumber daya gagal untuk menyelesaikan tugas tersebut, versi replikasi ini akan bisa digunakan. Logika ini sangat penting untuk tugas yang dilakukan secara waktu nyata yang harus diproses untuk memberikan layanan secara tepat waktu.