Internet of Thing

Catatan Kuliah #6

Alauddin Maulana Hirzan, M. Kom
0607069401

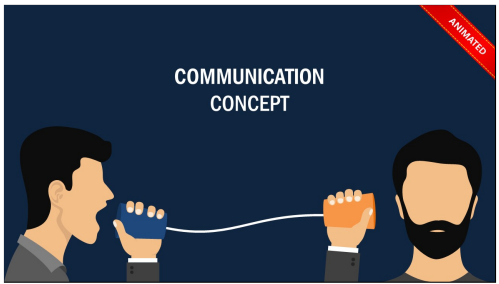
Protokol Komunikasi IoT

Protokol Komunikasi IoT

Apa itu Komunikasi?

Dilansir dari **merriam-webster.com**:

> *Sebuah proses dimana informasi dipertukarkan antara individu melalui sistem simbol, tanda, atau perilaku yang sama*



Protokol Komunikasi IoT

Komposisi Komunikasi

Agar komunikasi dapat terjadi dengan baik maka diperlukan beberapa hal seperti:

- Pengirim
- Penerima
- Media
- Bahasa/Protokol

Info

Dalam dunia komputer, komposisi komunikasi masih sama dan yang membedakan hanyalah kecepatan penyampaian yang bisa dihitung dalam satuan tertentu.

Protokol Komunikasi IoT

Komposisi Komunikasi Komputer #1

Pengirim dan **Penerima** dalam kasus ini adalah ***End Nodes*** atau perangkat yang terletak paling akhir/ujung. Sehingga tidak ada perangkat lain yang tersambung dengan perangkat tersebut. Oleh karena itulah pengguna komputer juga bisa dipanggil sebagai ***End Users***.

Berikut ini adalah perangkat yang masuk dalam kategori *End Nodes* adalah

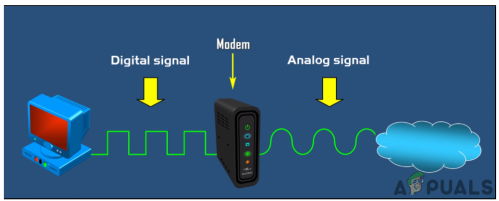
- Laptop
- Komputer
- *Thinclient*
- *Smartphone*
- *IoT Board* (tergantung situasi)

Protokol Komunikasi IoT

Komposisi Komunikasi Komputer #2

Komponen penting berikutnya adalah **media** komunikasi. Jika manusia berbicara melalui media udara yang membantu penghantaran gelombang suara. Sedangkan komputer menggunakan sinyal digital untuk dapat berkomunikasi dengan komputer lainnya.

Maka untuk dapat mengirimkan sinyal-sinyal tersebut membutuhkan proses tambahan yaitu teknik **Modulasi-Demodulasi** *-red:**modem***



Protokol Komunikasi IoT

Komposisi Komunikasi Komputer #3

Aspek terakhir yang tidak kalah penting adalah bahasa atau protokol. Protokol ini memiliki fungsi atau tujuan untuk menyamakan bentuk/pola komunikasi yang dikirimkan. Sehingga baik pengirim maupun penerima dapat sama-sama mengerti apa pesan yang dikirimkan.

Protokol utama yang digunakan oleh komputer untuk berkomunikasi adalah:

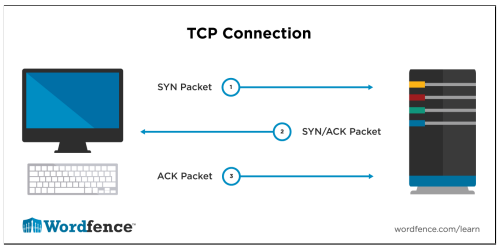
- TCP
- UDP

Protokol Komunikasi IoT

Protokol ***Transmission Control Protocol***

Protokol ini kenal sebagai *connection-oriented*. Protokol ini menggunakan mode komunikasi jaringan di mana sesi komunikasi atau koneksi semi permanen dibuat sebelum data yang berguna dapat ditransfer.

Koneksi Dulu Baru Data



Protokol Komunikasi IoT

Mekanisme TCP #1

Sebelum seseorang dapat mengirim data apa pun, klien dan server harus membuat koneksi. Server harus secara aktif mendengarkan permintaan klien setiap kali koneksi dibuat. Protokol TCP berbasis koneksi, sehingga membuat dan memelihara koneksi antara penerima dan pengirim saat data sedang dikirimkan di antara mereka. Oleh karena itu, informasi apa pun yang berjalan melalui internet dijamin akan sampai tanpa perubahan.

Sehingga dalam koneksinya terkadang memutuhkan waktu yang lebih lama dari UDP

Info
Teknik TCP meniru konsep komunikasi manusia yang diawali dengan jabat tangan atau ***Handshake***

Protokol Komunikasi IoT

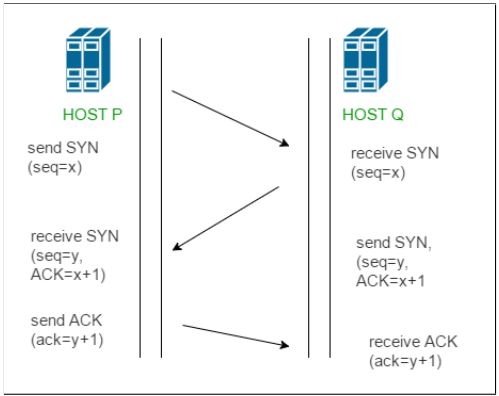
Mekanisme TCP #2 - *Three-way Handshake* 1

Koneksi dengan TCP dimulai dengan langkah-langkah berikut:

- Langkah 1 (SYN): Pada langkah pertama, klien ingin membuat koneksi dengan server, sehingga mengirimkan segmen dengan SYN (Synchronize Sequence Number)
- Langkah 2 (SYN + ACK): Server merespons permintaan klien dengan bit sinyal SYN-ACK yang ditetapkan.
- Langkah 3 (ACK): Di bagian akhir, klien mengakui respons server dan keduanya membuat koneksi yang andal yang dengannya mereka akan memulai transfer data aktual

Protokol Komunikasi IoT

Mekanisme TCP #2 - *Three-way Handshake* 2



Protokol Komunikasi IoT

Mekanisme TCP #3 - Pengiriman Data 1

1. Pengirim memulai proses dengan yang berikut:

- ***Sequence Number*** (Seq=521): Berisi nomor urut awal acak yang dihasilkan di sisi pengirim.
- ***SYN Flag*** (Syn=1): Meminta penerima untuk menyinkronkan nomor urutnya dengan nomor urut yang disediakan di atas
- ***Maximum Segment Size*** (MSS=1460 B): Pengirim memberitahukan ukuran segmen maksimumnya.
- ***Window Size*** (Window=14600 B): Pengirim menceritakan tentang kapasitas buffernya di mana ia harus menyimpan pesan dari penerima.

Protokol Komunikasi IoT

Mekanisme TCP #3 - Pengiriman Data 2

2. TCP adalah protokol full-duplex sehingga pengirim dan penerima memerlukan ***Window*** untuk menerima pesan dari satu sama lain.

- ***Sequence Number*** (Seq=2000): Berisi nomor urut awal acak yang dihasilkan di sisi penerima.
- ***SYN Flag***(Syn=1): Meminta pengirim untuk menyinkronkan nomor urutnya dengan nomor urut yang disediakan di atas
- ***Maximum Segment SIze*** (MSS=500 B): Pengirim memberitahukan ukuran segmen maksimumnya
- Karena MSSreceiver < MSSsender, kedua belah pihak menyetujui MSS minimum yaitu 500 B untuk menghindari fragmentasi paket di kedua ujungnya

Protokol Komunikasi IoT

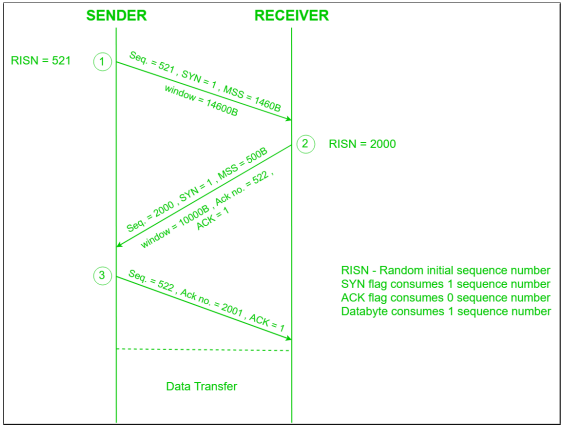
Mekanisme TCP #3 - Pengiriman Data 3

3. Pengirim membuat balasan terakhir untuk pembentukan koneksi dengan cara berikut:

- ***Sequence Number*** (Seq=522): Karena nomor urut = 521 pada langkah pertama dan flag SYN menggunakan satu nomor urut, maka nomor urut berikutnya adalah 522.
- ***SYN Number*** (No. Ack=2001): Karena pengirim menerima paket SYN=1 dari penerima dengan nomor urut 2000, maka nomor urut berikutnya yang diharapkan adalah 2001.
- ***ACK Flag*** (ACK=1): Memberitahukan bahwa nomor ACK berisi urutan berikutnya yang diharapkan oleh pengirim.

Protokol Komunikasi IoT

Mekanisme TCP #3 - Ilustrasi



Protokol Komunikasi IoT

Mekanisme TCP #4 - ***Flags***

Maka dalam protokol TCP memiliki fungsi kontrol dengan menggunakan ***Flags***:

- ***Synchronization*** (SYN) – Digunakan pada langkah pertama fase pembentukan koneksi atau proses jabat tangan 3 arah antara dua host.
- ***Acknowledgement*** (ACK) – Digunakan untuk mengakui paket yang berhasil diterima oleh host.
- ***Finish*** (FIN) – Digunakan untuk meminta pemutusan koneksi
- ***Reset*** (RST) – Digunakan untuk mengakhiri koneksi jika pengirim RST merasa ada yang salah dengan koneksi TCP atau bahwa percakapan seharusnya tidak ada.

Protokol Komunikasi IoT

Kapan Menggunakan TCP?

Berikut ini adalah kunci penting untuk menggunakan TCP:

- TCP berorientasi pada koneksi.
- Menggunakan timeout, checksum, dan acknowledgement untuk mencegah dan memperbaiki kesalahan.
- Paket data TCP memiliki nomor urut di header
- Sangat ideal untuk transmisi point-to-point saja
- Menggunakan informasi kontrol aliran untuk mengkalibrasi kecepatan transmisi data
- Mengimplementasikan algoritma penghindaran kemacetan
- Sangat andal

Protokol Komunikasi IoT

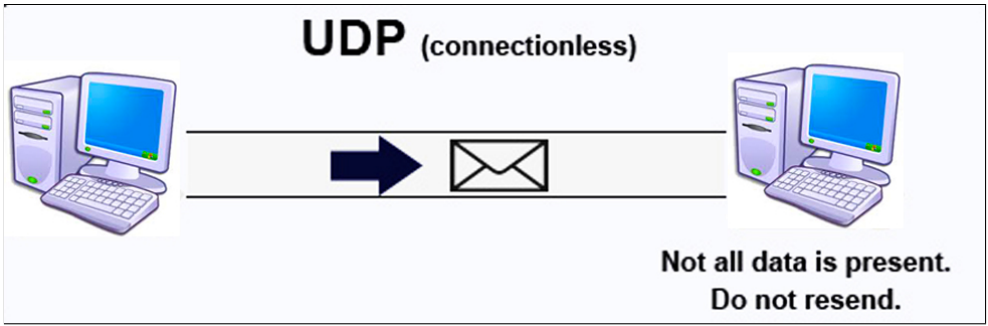
Protokol ***User Datagram Protocol***

User datagram protocol (UDP) adalah protokol komunikasi berorientasi pesan yang memungkinkan perangkat komputasi dan aplikasi untuk mengirim data melalui jaringan tanpa memverifikasi pengirimannya, yang paling cocok untuk komunikasi *realtime* dan sistem *broadcast*.

Selain itu, tidak menjamin pengiriman paket data dari server. Ini biasanya disebut sebagai protokol "tembak-dan-lupakan" karena tidak peduli apakah klien menerima data atau tidak.

Protokol Komunikasi IoT

Mekanisme UDP



Protokol Komunikasi IoT

Aplikasi UDP

UDP dapat digunakan untuk beberapa aplikasi seperti

- Digunakan untuk komunikasi permintaan-tanggapan sederhana ketika ukuran data lebih kecil dan karenanya ada kekhawatiran yang lebih kecil tentang kontrol aliran dan kesalahan.
- Untuk multicasting karena UDP mendukung packet switching.
- UDP digunakan untuk beberapa protokol pembaruan perutean seperti RIP
- Biasanya digunakan untuk aplikasi *realtime* yang tidak dapat mentolerir *delay* yang tidak merata

Protokol Komunikasi IoT

TCP atau UDP?

Kembali lagi pengguna, apakah ada data yang harus dikirim secara *realtime* atau data harus valid dari A-Z. Hal-hal ini akan menentukan jenis protokol yang digunakan.

Untuk perangkat sederhana seperti pengendali lampu, suhu, kelembaban dapat menggunakan protokol sederhana seperti UDP.

Namun untuk data monitoring yang di mana data harus valid dan tidak boleh rusak, maka TCP yang harus digunakan.

Protokol Komunikasi IoT

Layanan *Messaging* IoT

Komponen perangkat lunak yang dapat digunakan oleh perangkat IoT dapat dilihat sebagai berikut:

- Blynk (Platform)
- Cayenne (Platform)
- KaaIoT (Platform)
- Message Queueing Telemetry Transport
- Telegram Bot

Protokol Komunikasi IoT

Layanan *Messaging* IoT #1

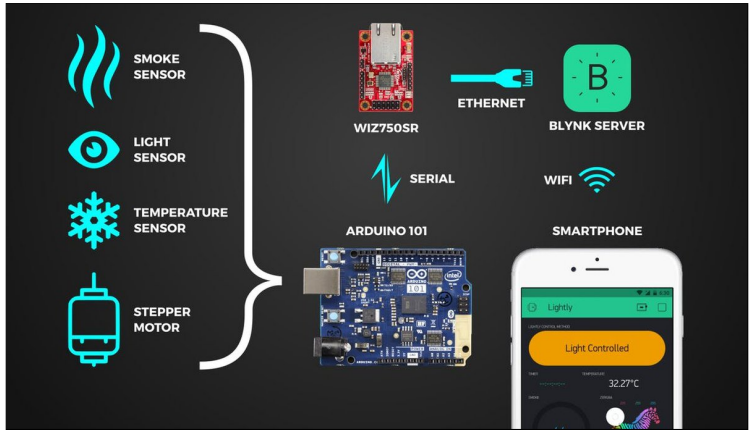
Terdapat banyak layanan perpesanan yang dihadirkan oleh berbagai macam perusahaan. Dengan adanya layanan ini para pengguna IoT tidak perlu **mengkonfigurasikan** tempat penyimpanan pesan-pesan atau data yang diambil dari perangkat.

Fitur inilah yang disebut dengan ***IoT Platform***. Platform IoT membantu memfasilitasi komunikasi, aliran data, manajemen perangkat, dan fungsionalitas aplikasi.

Meski menghadirkan versi gratis, namun perangkat yang diizinkan sangat terbatas. Dengan menggunakan IoT Platform data tersimpan secara *online* sehingga tidak perlu mengatur penyimpanan lokal

Protokol Komunikasi IoT

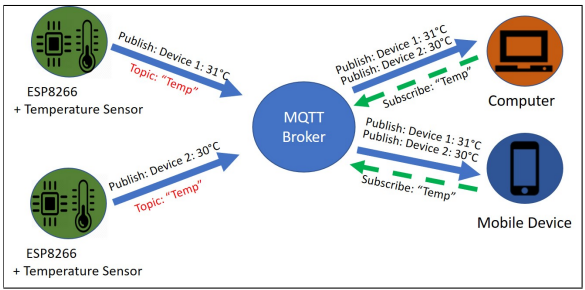
Layanan *Messaging* IoT #2



Protokol Komunikasi IoT

Layanan *Messaging* IoT #3

Message Queueing Telemetry Transport merupakan teknologi komunikasi dengan bantuan pihak ketiga sebagai penyalur informasi. Selain itu aplikasi ini sangat ringan dan cocok untuk mengirimkan data-data teks sederhana ke penerima.



Protokol Komunikasi IoT

Layanan *Messaging* IoT #4

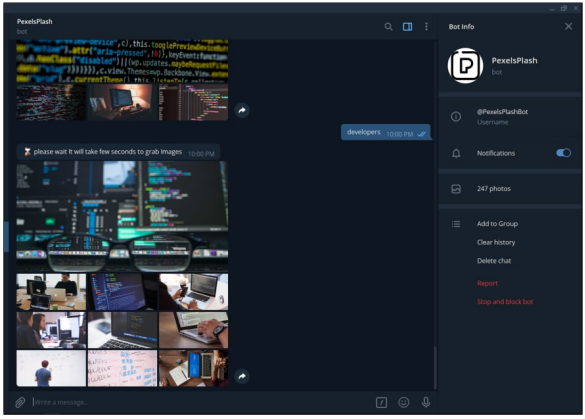
Telegram Bot merupakan alat komunikasi paling efektif dibandingkan lainnya. Selain dapat digunakan untuk komunikasi dua arah, aplikasi ini juga dapat digunakan untuk mengirimkan data multimedia seperti ***Video*** maupun ***Audio***. Sehingga cocok digunakan untuk monitoring CCTV.

Info

Sangat direkomendasikan untuk mengimplementasikan Telegram Bot di perangkat lain agar tidak mengganggu aktivitas dari perangkat sensor.

Protokol Komunikasi IoT

Ilustrasi




THANK YOU