

SIMULASI HINGGA IMPLEMENTASI IoT dan WSN

**Ahmad Wilda Yulianto, ST., MT.
M. Nanak Zakaria, S.T., M.T.
Dr. Mochammad Junus, ST., MT.
Rieke Adriati Wijayanti, ST., MT.**

SIMULASI HINGGA IMPLEMENTASI IoT dan WSN

Copyrights © 2022. All Rights Reserved
Hak cipta dilindungi undang-undang

Penulis:

Ahmad Wilda Yulianto, ST., MT.
M. Nanak Zakaria, S.T., M.T.
Dr. Mochammad Junus, ST., MT.
Rieke Adriati Wijayanti, ST., MT.

Penyunting:

Dhega Febiharsa

Desain & Tata Letak:

Tim Penerbit Cerdas Ulet Kreatif

ISBN : **978-623-8025-21-3**

Cetakan Pertama : **2022**

Penerbit :

Cerdas Ulet Kreatif (Anggota IKAPI No. 156 / JTI / 2014)

Jl. Manggis 72 RT 03 RW 04 Jember Lor - Patrang

Jember - Jawa Timur 68118

Telp. 0331-4431347, 412387 Faks. 4431347

e-mail : info@cerdas.co.id

Distributor Tunggal:

Cerdas Ulet Kreatif

Jl. Manggis 72 RT 03 RW 04 Jember Lor - Patrang

Jember - Jawa Timur 68118

Telp. 0331-4431347, 412387 Faks. 4431347

e-mail : info@cerdas.co.id

Undang-Undang RI Nomor 19 Tahun 2002 Tentang Hak Cipta

Ketentuan Pidana

Pasal 72 (ayat 2)

Barang Siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran Hak Cipta atau hak terkait sebagaimana dimaksud pada ayat (1), dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah).

KATA PENGANTAR

Alhamdulillah, segala puji kami haturkan kepada Allah yang selalu memberikan kemudahan dan kefasihan dalam melengkapi buku yang berjudul **Simulasi hingga Implementasi IoT dan WSN**.

Buku ini merupakan buku yang digunakan dalam keilmuan bidang *Internet of Things (IoT)* dan *Wireless Sensor Network (WSN)*. Buku ini berisi dasar-dasar sistem operasi linux, simulator ns-3, raspberry, cloud dan aplikasi mobile.

Rasa terima kasih kami yang terdalam diberikan kepada semua kolega yang telah membantu kami dan dengan tulus memberikan dukungan moral dalam setiap tahap proses pembuatan buku ini.

Akhirnya, kami mengucapkan selamat membaca dan tentunya kami tidak lupa untuk mengharapkan kritik dan saran agar kami selalu rajin berbenah.

Malang, Juli 2022

Penulis

DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI	iv
BAB 1 PLATFORM IOT BERBASIS CLOUD	1
A. Pendahuluan	1
B. Alat dan Bahan	2
C. Simulasi dan Implementasi	2
D. Ringkasan	11
E. Latihan:	11
F. Tugas:	11
BAB 2 WSL dan Simulator NS-3.....	13
A. Pendahuluan	13
B. Alat dan Bahan	13
C. Simulasi dan Implementasi	13
D. Ringkasan	23
E. Latihan:	23
F. Tugas:	24
BAB 3 SIMULASI WSN.....	25
A. Pendahuluan	25
B. Alat dan Bahan	27
C. Simulasi dan Implementasi	27
D. Ringkasan	30
E. Latihan:	30
F. Tugas:	30

BAB 4 TOPOLOGI JARINGAN DAN ROUTING WSN.....	31
A. Pendahuluan.....	31
B. Alat dan Bahan.....	32
C. Simulasi dan Implementasi.....	32
D. Ringkasan	33
E. Latihan:.....	33
F. Tugas:.....	34
BAB 5 AMBIL DATA SENSOR DENGAN RASPBERRY PI.....	35
A. Pendahuluan	35
B. Alat dan Bahan.....	37
C. Simulasi dan Implementasi.....	37
D. Ringkasan	42
E. Latihan:.....	42
F. Tugas:.....	43
BAB 6 CHATBOT UNTUK IoT	45
A. Pendahuluan	45
B. Alat dan Bahan.....	46
C. Simulasi dan Implementasi.....	46
D. Ringkasan	49
E. Latihan:.....	49
F. Tugas:.....	49
BAB 7 GATEWAY IoT DAN WSN.....	51
A. Pendahuluan.....	51
B. Alat dan Bahan.....	52
C. Simulasi dan Implementasi.....	52
D. Ringkasan	56
E. Latihan:.....	56

F. Tugas:	56
BAB 8 APLIKASI IoT M5STACK KIT	57
A. Pendahuluan	57
B. Alat dan Bahan	58
C. Simulasi dan Implementasi	58
D. Ringkasan	61
E. Latihan:	61
F. Tugas:	61
BAB 9 PLATFORM NODE-RED dan MQTT pada Raspberry Pi	63
A. Pendahuluan	63
B. Alat dan Bahan	64
C. Simulasi dan Implementasi	64
D. Ringkasan	68
E. Latihan:	68
F. Tugas:	68
BAB 10 LoRA	69
A. Pendahuluan	69
B. Alat dan Bahan	71
C. Simulasi dan Implementasi	71
D. Ringkasan	78
E. Latihan:	78
F. Tugas:	79
DAFTAR PUSTAKA	80
GLOSARIUM	82
INDEKS	83

BAB 1

PLATFORM IOT BERBASIS CLOUD

Setelah menyelesaikan bab ini, mahasiswa diharapkan:

1. Memahami konsep platform IoT berbasis cloud
2. Memahami format pertukaran data
3. Menjelaskan cara kerja sensor

A. Pendahuluan

Sensor merupakan komponen penting yang dibutuhkan untuk pengumpulan data-data dalam proyek *Internet of Things (IoT)* dan *Wireless Sensor Network (WSN)*. Namun, membeli setiap sensor per item dan menyediakan tempat untuk penyimpanannya merupakan suatu masalah tersendiri, terutama saat kita mulai membuat lebih banyak proyek. Untungnya, setidaknya ada sepuluh sensor berteknologi tinggi dalam perangkat yang kita gunakan hampir setiap hari di ponsel pintar kita. Sensor mulai dari jenis akselerometer, giroskop, sensor tekanan dan cahaya, hingga GPS, dimana perangkat ini dapat digunakan untuk mengumpulkan data yang diperlukan untuk memulai berbagai proyek IoT.

Data dari sensor-sensor dikumpulkan secara terpusat pada server untuk selanjutnya akan disimpan guna keperluan analisa. Selain itu pada proses pengumpulan data, proses masuknya data juga ditampilkan dalam bentuk visualisasi pada suatu system berbasis dashboard. Sistem dashboard bermanfaat dalam mengamati dan memastikan bahwa node sumber informasi bekerja sesuai yang direncanakan. Jika ada permasalahan data tidak terkirim ke server, maka bisa dengan segera diketahui.

Salah satu cara terbaik dan termudah untuk mengumpulkan, memvisualisasikan, dan menganalisis data ponsel pintar itu adalah dengan memanfaatkan uBeac. uBeac merupakan platform IoT serbaguna untuk transformasi digital terpusat, integrasi, dan visualisasi data. Hub IoT uBeac memungkinkan untuk menghubungkan, memproses, dan memvisualisasikan

data waktu nyata dengan cara yang aman. Sehubungan dengan aplikasi Data Collector, kita akan mengerjakan proyek IoT dalam waktu singkat.

B. Alat dan Bahan

Alat dan bahan yang dibutuhkan pada Praktikum ini adalah:

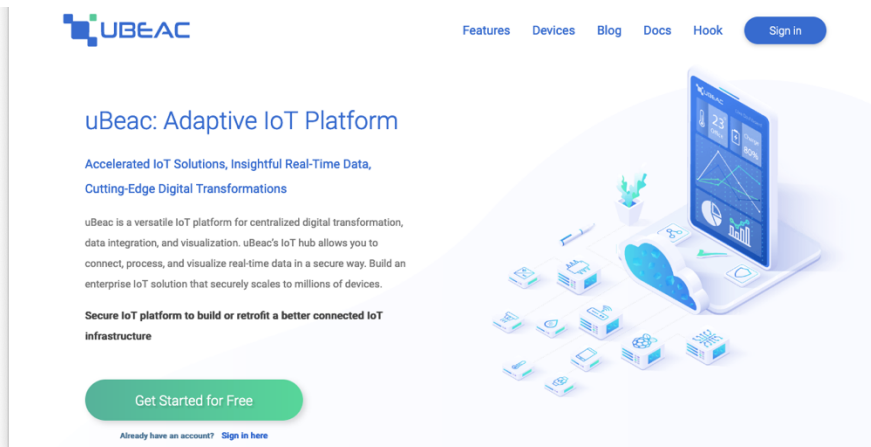
1. Perangkat Ponsel Android
2. Platform IoT uBeac
3. Aplikasi Data Collector: <https://play.google.com/store/apps/details?id=tech.unismart.dc&hl=en>

C. Simulasi dan Implementasi

Adapun cara menggunakan aplikasi Data Collector dengan platform IoT uBeac adalah sebagai berikut.

(1) Mendaftar pada website uBeac

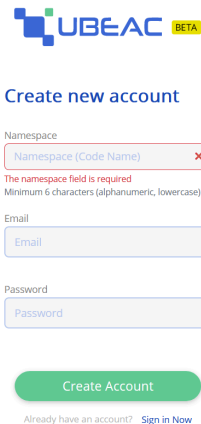
Buka laman <https://www.ubeac.io> kemudian klik tombol *Get Started for Free* seperti pada Gambar 1.1.



Gambar 1.1 Laman awal uBeac

Mendaftar dengan uBeac hanya perlu menambahkan email dan membuat kata sandi. Kemudian dilanjutkan dengan membuat Tim dengan cara memasukkan nama tim, kode, dan alamat. Setelah menyelesaikan

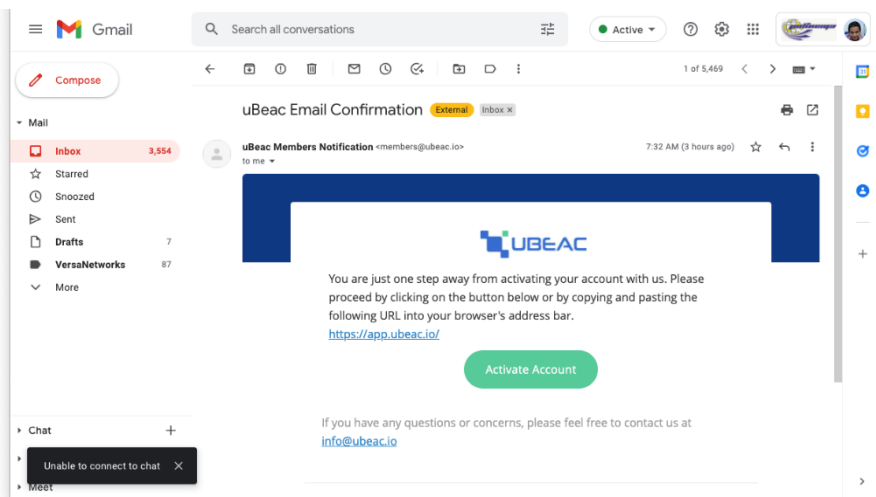
penambahan tim, kemudian dialihkan pada beranda kosong untuk tim kita.



The image shows the registration page for uBeac. At the top is the uBEAC logo with a 'BETA' badge. Below it is the heading 'Create new account'. There are three input fields: 'Namespace (Code Name)' with a red border and an error message 'The namespace field is required. Minimum 6 characters (alphanumeric, lowercase)', 'Email', and 'Password'. A green 'Create Account' button is at the bottom, with a link 'Sign in Now' for existing users.

Gambar 1.2 Form Pendaftaran uBeac

Setelah menyelesaikan langkah ini, akan berada di beranda akun. Konfirmasi pada email diperlukan untuk memvalidasi akun. Buka akun email yang didaftarkan pada uBeac untuk mendapatkan email konfirmasi seperti pada gambar



Gambar 1.3 Konfirmasi uBeac pada email

Create Team

Namespace
newOffice ✓

Minimum 6 characters (alphanumeric)

Name
my office

Address 1
180 West Beaver Creek Road

Address 2 (optional)
Address 2 (optional)

City
Richmond Hill

Province
Ontario

Country
Canada

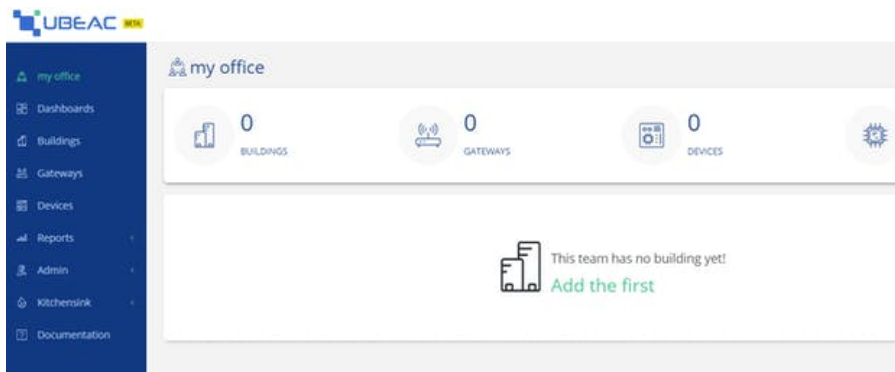
Postal Code
L4B 1L1

Description (optional)
Brief description about your team or project

Submit

Gambar 1.4 Membuat tim di uBeac

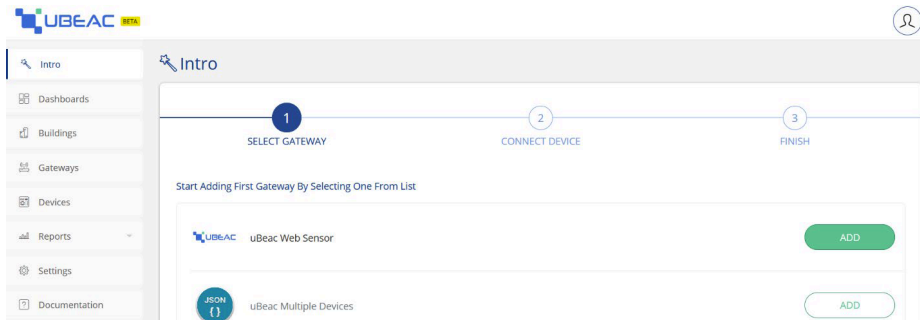
Pembuatan tim seperti pada gambar 1.4 digunakan untuk mengidentifikasi pengguna uBeac secara berkelompok.



Gambar 1.5 Tampilan halaman tim di uBeac

Tampilan halaman Tim menunjukkan informasi mengenai jumlah parameter-parameter yang ditentukan meliputi jumlah gedung, gateway, perangkat yang

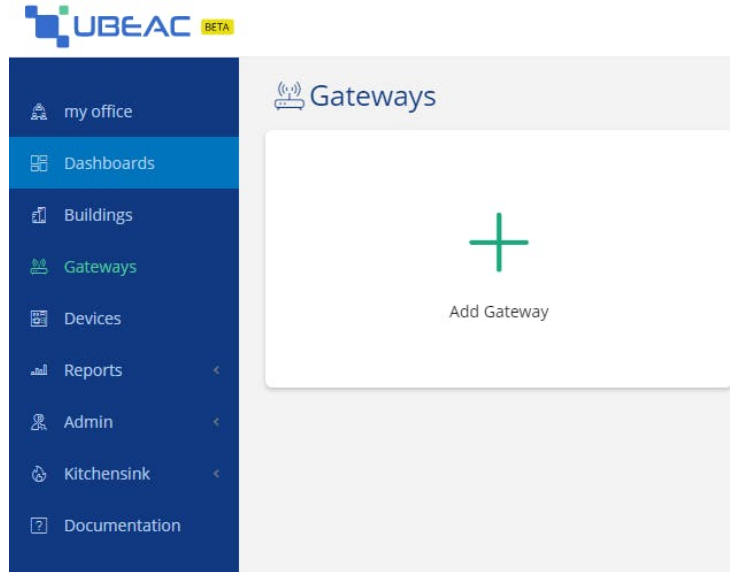
terhubung dalam sistem, dan parameter lain yang bisa kita atur.



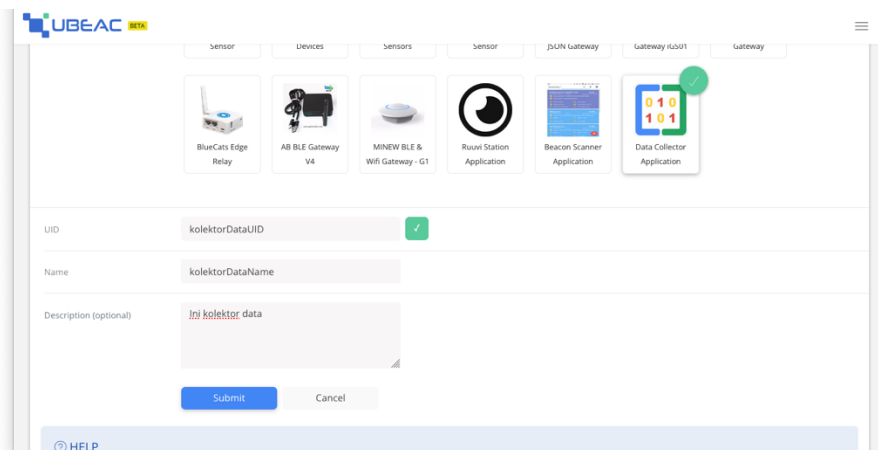
Gambar 1.6 Tampilan laman beranda uBeac.

(2)Menyiapkan uBeac

Sekarang setelah menyiapkan tim, selanjutnya perlu membuat gateway untuk menghubungkan semua perangkat. Dari beranda uBeac , klik modul Gateways dan tambahkan gateway baru. Di bawah tab General , buat UID dan nama untuk gateway. Untuk tujuan dalam Job Sheet ini, pilih Aplikasi Pengumpul Data sebagai gateway. Selanjutnya, di bawah tab HTTP , akan ada dua URL gateway: satu HTTP dan satu HTTPS. Url ini akan digunakan untuk menghubungkan ke ponsel. Klik kirim untuk menambahkan gateway seperti tampak pada Gambar 1.7



Gambar 1.7 Membuat Gateway baru



Gambar 1.8 Menambahkan info ke gateway

(3)Memulai Data Collector

Data Collector adalah aplikasi berbasis Android yang mengumpulkan data dari banyak sensor dan mengirimkannya dalam format JSON ke server dalam

jumlah tak terbatas pada waktu tertentu menggunakan HTTP, HTTPS, Web Socket , dan/atau Web Socket Secure .

Kolektor Data dilengkapi dengan empat modul: Sensor, Server, Tasks, dan Settings.

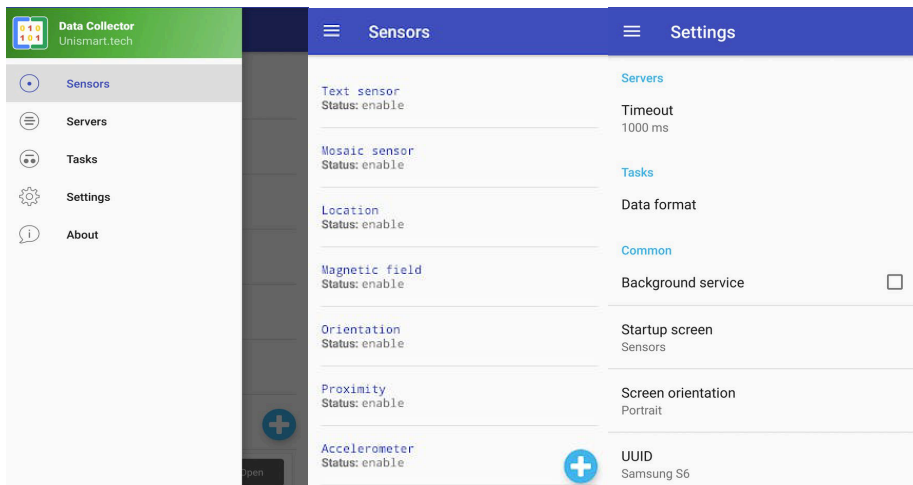
a) Modul Sensors menampilkan daftar semua sensor yang terhubung ke perangkat Android. Ketika melakukan klik pada setiap sensor maka akan menampilkan data saat ini yang sedang dibaca dari sensor tertentu.

b) Modul Server memungkinkan pengguna untuk menambahkan jumlah server yang tidak terbatas untuk membaca data sensor.

c) Modul Tasks memungkinkan pengguna menentukan apa, di mana, kapan, dan bagaimana data akan dikirim dari perangkat Android ke server.

d) Modul Settings menunjukkan parameter umum seperti waktu tunggu server dan format data yang dikirim.

Untuk informasi lebih lanjut tentang Kolektor Data, kunjungi https://unismart.tech/projects/data_collector/ . Install aplikasi melalui google playstore “Data Collector App” di ponsel.



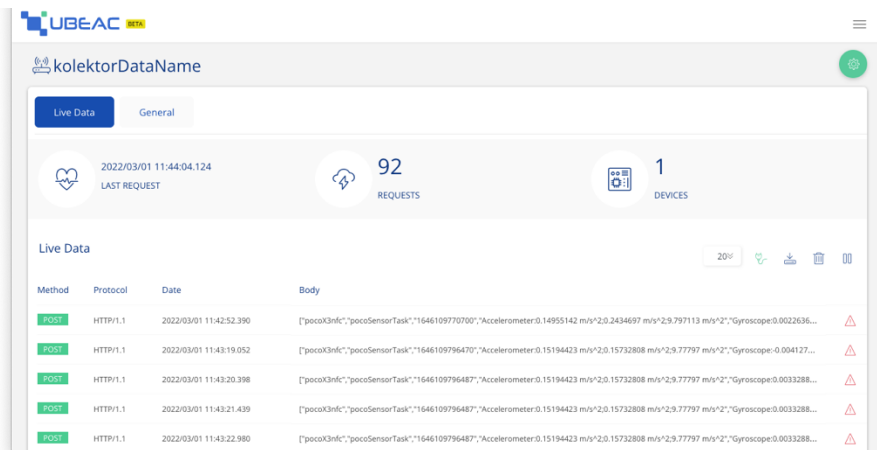
Gambar 1.9 Konfigurasi Data Collector App

(4) Menghubungkan Kolektor Data ke uBeac

Pada aplikasi Kolektor Data, pilih modul Server dan tambahkan server baru. Berikan nama untuk server ini dan tambahkan salah satu URL Gateway dari uBeac. Klik simpan untuk menambahkan server.

Selanjutnya, buat tugas untuk dikirim ke server. Pilih modul Tasks dan tambahkan tugas baru. Beri nama tugas ini, lalu pilih sensor yang akan digunakan dari ponsel. Pilih server yang baru saja dibuat, dan tambahkan waktu pengambilan sampel, seberapa sering ingin Data Collector mengirimkan data ke uBeac. Setelah ini selesai, klik simpan untuk menambahkan tugas.

Terakhir, di Settings, pilih Data Format dan pilih semua opsi kecuali Strict JSON. Setelah ini selesai, saatnya untuk bekerja dengan uBeac.



Gambar 1.10 Tampilan data sensor yang masuk ke server uBeac.

(5) Debug

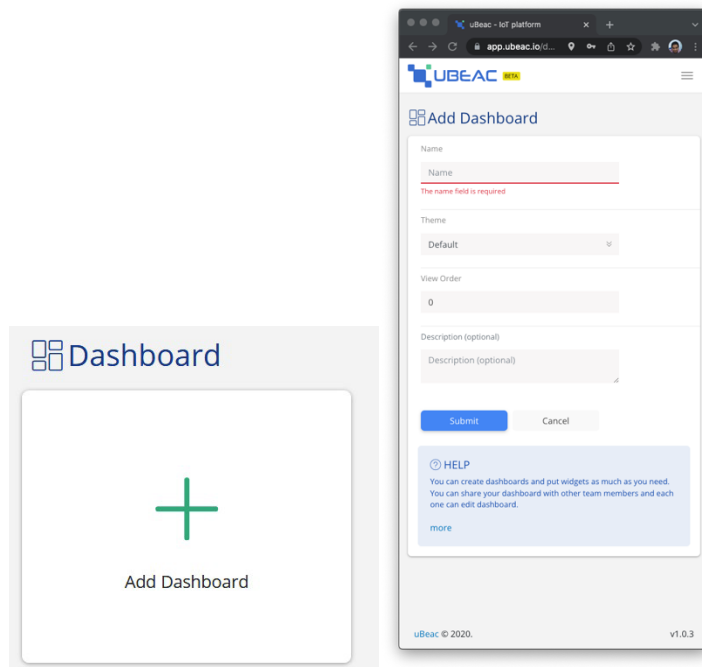
Kembali ke uBeac dan pilih modul Gateways untuk melihat bahwa perangkat telah ditambahkan ke gateway. Jika di klik pada gateway, kita dapat melihat semua

permintaan HTTP POST yang dikirimkan Android ke uBeac . Jika dipilih modul Device dan klik perangkat yang baru ditambahkan, yaitu perangkat Android, kita mendapati semua data yang dikirim setiap sensor ke uBeac .

Sekarang setelah memiliki semua data mentah ini, maka dapat menggunakan modul Dasbor untuk memvisualisasikannya.

(6)Membuat Dasbor uBeac

Memiliki dasbor untuk memvisualisasikan data yang masuk sangat berguna, terutama jika ingin menganalisis dan memanfaatkan data setelahnya. Pertama, harus mengatur dasbor dengan cara membuka modul Dasbor dan tambahkan dasbor baru. Pilih nama untuk dasbor lalu klik kirim. Dasbor kosong akan muncul, yang dapat disesuaikan. Di sudut kanan atas halaman dasbor, klik ikon papan klip untuk mulai mengedit dasbor.



Gambar 1.11 tampilan Dashbord

Widget seperti indikator, bagan, dan pelacak perangkat bisa dipilih bebas untuk memvisualisasikan data. Misalnya, jika membuat indikator untuk sensor cahaya sekitar ponsel, tarik dan lepas widget indikator ke dasbor. Selanjutnya klik tombol "connect to data" untuk mengedit pengaturan widget. Hal ini termasuk mengubah ikon tampilan, memilih perangkat untuk mengumpulkan data, dan fitur lain yang unik untuk setiap widget. Simpan tahapan untuk penambahan widget tersebut, selanjutnya bisa menambah widget lainnya.

Di bawah ini adalah beberapa contoh dasbor yang bisa dibuat hanya dengan sensor di ponsel.

(7)Meninjau Kembali Rekaman

Tampilan dasbor menunjukkan aktivitas sensor, itu tidak menampilkan data Anda sebelumnya. Itu disimpan dalam modul Laporan . Di sana, Anda dapat menemukan semua catatan historis data sensor Anda, sejak saat Anda memasang sensor. Anda juga bisa mendapatkan laporan dari seluruh gateway Anda. Data ini dapat disaring berdasarkan tanggal, rentang waktu, perangkat, dan sensor. Terakhir, jika Anda perlu menggunakan data ini untuk proyek lain, Anda dapat mengekspornya dalam JSON atau CSV.

Method	Name	Date	Body
Success	MyAppCollector	2022/03/02 01:30:11.491	{\"gateway\": \"gateway1\", \"accelerometer\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"magnetism\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"gyroscope\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}}
Success	MyAppCollector	2022/03/02 01:30:07.665	{\"gateway\": \"gateway1\", \"accelerometer\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"magnetism\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"gyroscope\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}}
Success	MyAppCollector	2022/03/02 01:30:06.710	{\"gateway\": \"gateway1\", \"accelerometer\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"magnetism\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"gyroscope\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}}
Success	MyAppCollector	2022/03/02 01:30:05.532	{\"gateway\": \"gateway1\", \"accelerometer\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"magnetism\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"gyroscope\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}}
Success	MyAppCollector	2022/03/02 01:30:01.716	{\"gateway\": \"gateway1\", \"accelerometer\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"magnetism\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"gyroscope\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}}
Success	MyAppCollector	2022/03/02 01:29:57.818	{\"gateway\": \"gateway1\", \"accelerometer\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"magnetism\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"gyroscope\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}}
Success	MyAppCollector	2022/03/02 01:29:53.738	{\"gateway\": \"gateway1\", \"accelerometer\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"magnetism\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"gyroscope\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}}
Success	MyAppCollector	2022/03/02 01:29:49.695	{\"gateway\": \"gateway1\", \"accelerometer\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"magnetism\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"gyroscope\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}}
Success	MyAppCollector	2022/03/02 01:29:45.743	{\"gateway\": \"gateway1\", \"accelerometer\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"magnetism\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"gyroscope\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}}
Success	MyAppCollector	2022/03/02 01:29:41.670	{\"gateway\": \"gateway1\", \"accelerometer\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"magnetism\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}, \"gyroscope\": {\"x\": 0.0009845, \"y\": 0.0009845, \"z\": 0.0009845}}

Gambar 1.12 Data rekam Gateway sehari sebelumnya

D. Ringkasan

Jobsheet ini membahas cara menggunakan platform uBeac untuk membuat dasbor dari sensor yang terdapat pada ponsel. Sekarang kita dapat menggunakannya untuk berbagai proyek, seperti melacak ke mana dan bagaimana bepergian ke dan dari tempat kerja, dan mengoptimalkan kegunaan sensor yang ada dalam ponsel.

E. Latihan:

Jawablah pertanyaan berikut ini.

1. Apakah perbedaan antara format JSON dengan CSV?
2. Sebutkan platform IoT selain uBeac!
3. Sebutkan disertai penjelasan masing-masing menu pada platform uBeac!
4. Sebutkan dan jelaskan masing-masing sensor yang terdapat pada ponsel Anda!
5. Jelaskan kelebihan dan kekurangan dari uBeac!

F. Tugas:

1. Buatlah sistem dashboard yang menampilkan semua sensor yang memungkinkan dalam ponsel Anda!
2. Buatlah laporan dari Modul 1 yang menjelaskan tahapan pembuatan Sistem Dashboard untuk

menampilkan dan menyimpan data dari sensor-sensor di ponsel Anda menggunakan Platform uBeac! (Menyertakan konfigurasi Gateway dan Layout Dashboard, Tugas 1, dan jawaban dari pertanyaan

BAB 2

WSL dan Simulator NS-3

Setelah mengambil bab ini, siswa hendaknya:

1. Memahami konsep Virtualisasi di windows
2. Memahami cara kerja simulator
3. Memebuat simulasi sederhana menggunakan NS-3

A. Pendahuluan

Windows Subsystem for Linux (WSL) merupakan fitur yang memungkinkan file binari Linux berjalan di Windows tanpa modifikasi. WSL memungkinkan pengguna untuk memanggil proses di Linux dari Windows dan sebaliknya, mengakses file di kedua sistem operasi, berbagi variabel lingkungan, dan menggabungkan perintah yang berbeda bersama-sama

B. Alat dan Bahan

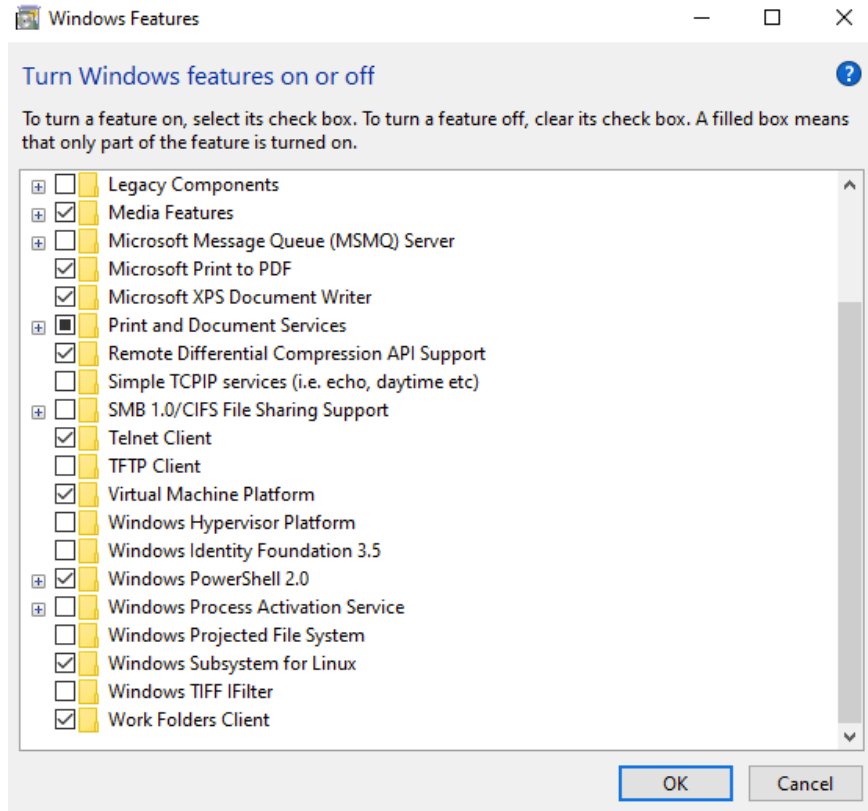
Alat dan bahan yang dibutuhkan pada Praktikum ini adalah:

1. Perangkat Laptop/PC
2. WSL
3. Distro Linux Ubuntu
4. Program ns-3

C. Simulasi dan Implementasi

Digital image terdiri atas:

- (1) Langkah pertama yang dilakukan adalah mengaktifkan fitur windows yaitu Virtual Machine Platform dan juga Windows Subsystem for linux.



- (2) Cara ke 2 melalui windows power sheel, dengan membuka windows power sheel sebagai *run as administrator*
Selanjutnya ketik di program run:

```
dism.exe /online /enable-feature  
/featurename:VirtualMachinePlatform /all /norestart
```

- (3) Pastikan untuk virtual machine platform dan Windows Subsystem for linux telah aktif
(4) kemudian download wsl nya Link tertera di modul (Link) dan download linux juga di microsoft store (Link), kemudian setelah download buat user dan pass baru di ubuntu.
(5) Setelah itu set wsl ke versi 2 dengan cara membuka windows power sheel dengan run administrator power sheel nya agar bisa mendapatkan hak akses tertinggi.

Kemudian ketik : `wsl --set-default-version 2` untuk mengganti wsl ke versi 2

- (6) Cek apakah wsl sudah terinstall linux dan cek wsl nya apakah sudah berubah ke versi 2, Dengan cara mengetik

```
wsl -l -v
```

- (7) Buka wsl untuk menjalankan ubuntu
(8) Tampilan CLI (comand line interface)
(9) Berikut cara menyiapkan program simulator ns-3
(10) Prasyarat dan Langkah Instalasi ke NS-3 di Ubuntu 20.04

- a. Perbarui Repositori dari Ubuntu

```
$ sudo sed -i  
's#archive.ubuntu.com#mirrors.tuna.tsinghua.edu.cn#g'  
/etc/apt/sources.list
```

- b. Update status repo dan Upgrade sistemnya

```
$ sudo apt update && sudo apt -y upgrade
```

- c. Pasangkan lingkungan untuk Desktop

```
$ sudo apt install xfce4 xfce4-goodies
```

- d. Pasangkan kebutuhan Pustaka inti

```
$ sudo apt install build-essential libsqlite3-dev  
libboost-all-dev libssl-dev git python3-setuptools  
castxml
```

- e. Pasangkan kebutuhan untuk bynding python ns-3

```
$ sudo apt install gir1.2-gocanvas-2.0 gir1.2-gtk-  
3.0 libgirepository1.0-dev python3-dev python3-gi  
python3-gi-cairo python3-pip python3-pygraphviz  
python3-pygccxml  
$ pip3 install kiwi
```

- f. Pasangkan kebutuhan Pustaka khusus

```
sudo apt install g++ pkg-config sqlite3 qt5-default
mercurial ipython3 openmpi-bin openmpi-common
openmpi-doc libopenmpi-dev autoconf cvs bzip2 unrar gdb
valgrind uncrustify doxygen graphviz imagemagick
python3-sphinx dia tcpdump libxml2 libxml2-dev cmake
libc6-dev libc6-dev-i386 libclang-6.0-dev llvm-6.0-
dev automake
```

g. Unduh dan ekstraksi paket ns-3

```
$ cd
$ wget -c https://www.nsnam.org/releases/ns-allinone-
3.35.tar.bz2
$ tar -xvzf ns-allinone-3.35.tar.bz2
```

(11) Perintah waf pada ns-3

```
$ cd ns-allinone-3.35/ns-3.35/
$ ./waf configure --enable-examples
$ ./waf
$ cd
```

(12) Build dan kompilasi ns-3

```
cd ns-allinone-3.35/
./build.py --enable-examples --enable-tests
Cek instalasi ns-3
cd ns-allinone-3.35/ns-3.35/
./waf --run hello-simulator
```

(13) Build dan Kompilasi netanim-3

```
$ cd ns-allinone-3.35/netanim-3.108/
$ make clean
```

```
$ qmake NetAnim.pro
$ make
$ ./NetAnim
$ cd
```

(14) Menggunakan ns-3 melakukan build, menjalankan simulasi.

a. Menjalankan program scenario dari contoh

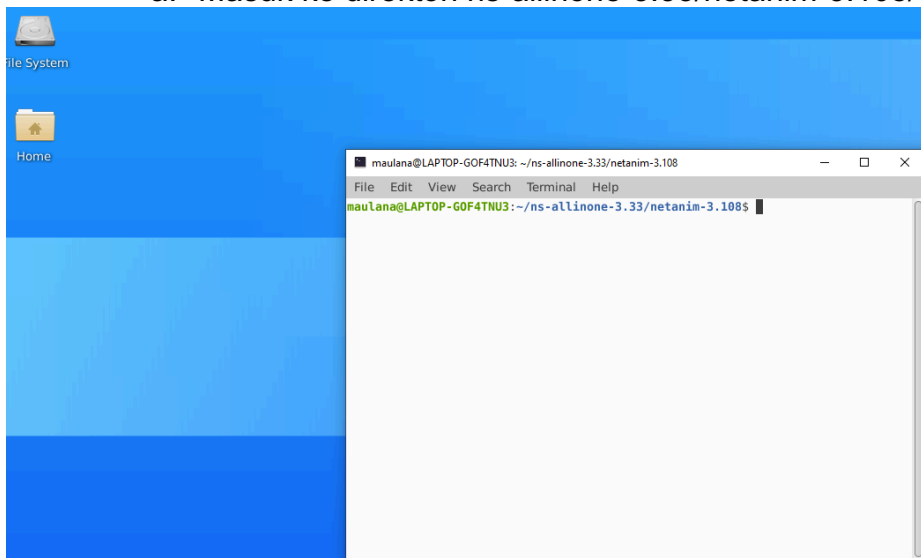
```
$ ./waf --run first
$ examples/tutorial/
$ ls -l examples/tutorial
```

b. Menjalankan program scenario dari program kustom:

```
$ nano scratch/satu.cc
```

(15) Memvisualisasikan simulasi dengan NetAnim

a. Masuk ke direktori ns-allinone-3.33/netanim-3.108/

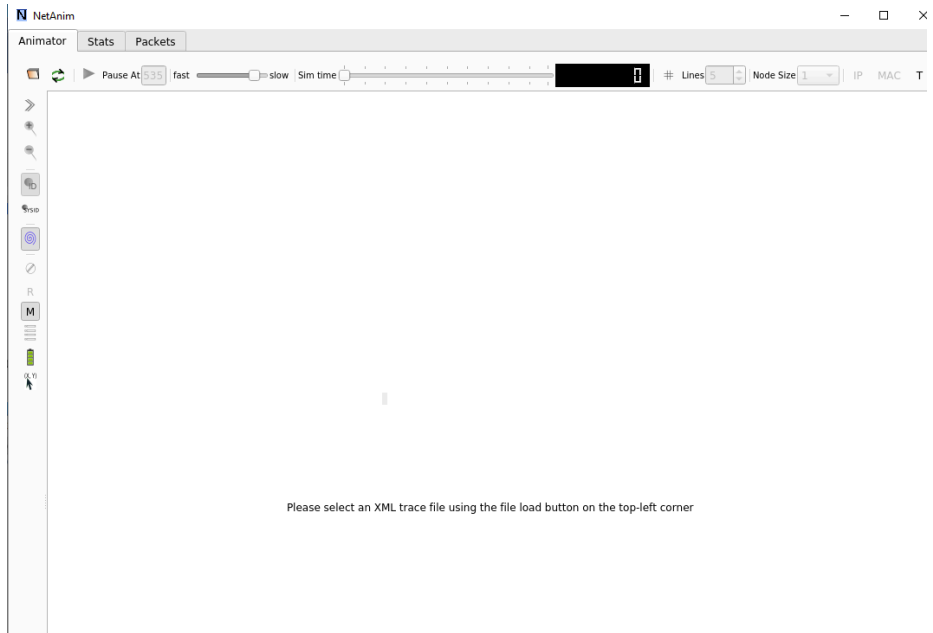


b. Jalankan perintah

```
make clean  
qmake NetAnim.pro  
make
```



A terminal window titled "maulana@LAPTOP-GOF4TNU3: ~/ns-allinone-3.33/netanim-3.108". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The prompt is "maulana@LAPTOP-GOF4TNU3:~/ns-allinone-3.33/netanim-3.108\$". The command ". /NetAnim" has been entered, and the cursor is at the end of the line.



Tampilan Animasi NetAnim

- c. Copy dan buka file first.cc ke direktori scratch

```
maulana@LAPTOP-GOF4TNU3: ~/ns-allinone-3.33/ns-3.33/scratch
File Edit View Search Terminal Help
GNU nano 4.8 first.cc
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/netanim-module.h"

// Default Network Topology
//
//      10.1.1.0
// n0 ----- n1

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit          ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line
```

- d. Tambahkan kode untuk menggenerate file animasi dalam format xml.

```
maulana@LAPTOP-GOF4TNU3: ~/ns-allinone-3.33/ns-3.33/scratch
File Edit View Search Terminal Help
GNU nano 4.8 first.cc
stack.Install (nodes);

Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer interfaces = address.Assign (devices);

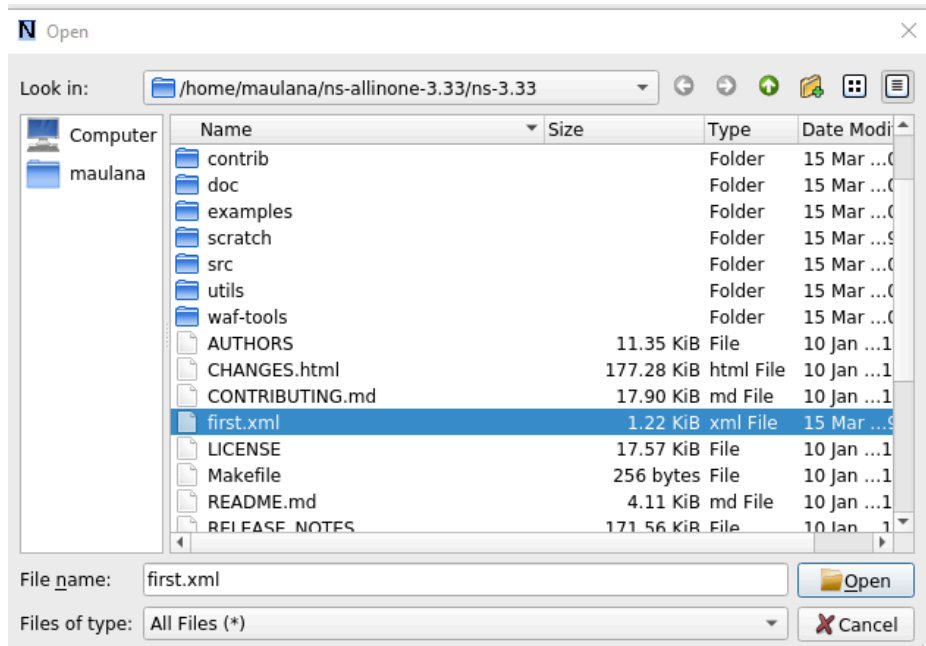
UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));

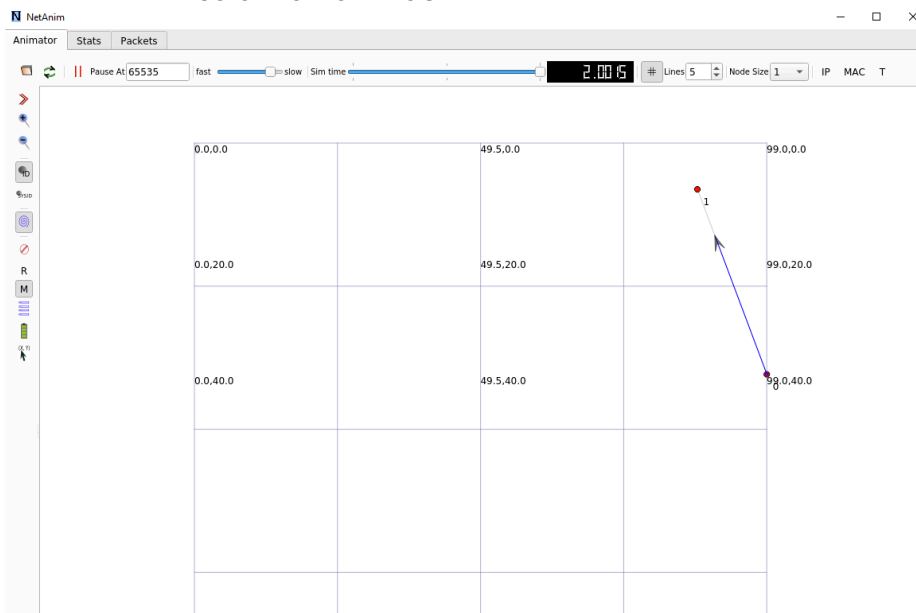
UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
echoClient.SetAttribute ("MaxPackets", UIntegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.0)));
echoClient.SetAttribute ("PacketSize", UIntegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));
AnimationInterface anim("first.xml");
Simulator::Run ();
Simulator::Destroy ();
return 0;
}
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^_ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

e. Jalankan dan Buka file hasil running program



Lokasi file xml
f. Jalankan animasi



Animasi NetAnim
(16) Program contoh yang tersedia di NS3

```
maulana@LAPTOP-GOF4TNU3: ~/ns-allinone-3.33/ns-3.33
File Edit View Search Terminal Help
maulana@LAPTOP-GOF4TNU3:~/ns-allinone-3.33/ns-3.33$ ls -l examples/tutorial/
total 76
-rw-r--r-- 1 maulana maulana 859 Jan 10 2021 examples-to-run.py
-rw-r--r-- 1 maulana maulana 6487 Jan 10 2021 fifth.cc
-rw-r--r-- 1 maulana maulana 2464 Jan 10 2021 first.cc
-rw-r--r-- 1 maulana maulana 2238 Jan 10 2021 first.py
-rw-r--r-- 1 maulana maulana 1791 Jan 10 2021 fourth.cc
-rw-r--r-- 1 maulana maulana 894 Mar 15 01:12 hello-simulator.cc
-rw-r--r-- 1 maulana maulana 3592 Jan 10 2021 second.cc
-rw-r--r-- 1 maulana maulana 3431 Jan 10 2021 second.py
-rw-r--r-- 1 maulana maulana 10001 Jan 10 2021 seventh.cc
-rw-r--r-- 1 maulana maulana 7252 Jan 10 2021 sixth.cc
-rw-r--r-- 1 maulana maulana 6048 Jan 10 2021 third.cc
-rw-r--r-- 1 maulana maulana 5649 Jan 10 2021 third.py
-rw-r--r-- 1 maulana maulana 1417 Jan 10 2021 wscript
maulana@LAPTOP-GOF4TNU3:~/ns-allinone-3.33/ns-3.33$
```

Daftar program contoh yang ada di NS-3

- (17) Menganalisa Packet dengan Wireshark
 - a. Mengaktifkan pelacakan Pcap kedalam scenario
 - b. Tambahkan fungsi EnablePcapAll
 - c. Akan dihasilkan file berekstensi *.pcap
 - d. Melihat
- (18) Membangkitkan data dan Plot data ke grafik
- (19) Menggunakan Editor kode
 - a. Visual Studio
 - b. PyCharm
 - c. Clion
 - d. Eclipse
 - e. Atom

D. Ringkasan

Jobsheet ini merupakan cara melakukan instalasi linux menggunakan WSL dan menjalankan simulator ns-3 sebagai tools simulasi ns-3.

E. Latihan:

Jawablah pertanyaan berikut ini.

1. Jelaskan cara kerja WSL!

2. Sebutkan dan Jelaskan jenis dan perbedaan simulator jaringan yang anda ketahui !

F. Tugas:

1. Buatlah sistem dashboard yang menampilkan semua sensor yang memungkinkan dalam ponsel Anda!
2. Buatlah laporan dari Modul 2 yang menjelaskan tahapan instalasi dan konfigurasi simulator NS-3 yang dijalankan pada ubuntu melalui WSL! (Menyertakan konfigurasi dan printscreen setiap tahapan praktikum, Tugas 2, dan jawaban dari pertanyaan point E.

BAB 3

SIMULASI WSN

Setelah menyelesaikan bab ini, mahasiswa diharapkan:

1. Memahami struktur pembuatan program ns-3
2. Memahami format pertukaran data
3. Menjelaskan cara kerja sensor

A. Pendahuluan

Sensor ini adalah kebiasaan umum programmer untuk mencetak pesan yang membantu mereka melacak dan mencatat apa yang dilakukan program mereka. Untuk contoh:

```
void SomeClass::DoSomething (int x)
{
std::cout << "Inside DoSomething() with x=" << x;
}
```

Dengan cara ini, kita bisa mendapatkan beberapa informasi tentang apa yang terjadi di program kita, misalnya, apakah fungsi `DoSomething` dipanggil sama sekali. Kekurangannya di sini adalah bahwa pesan ini akan selalu dicetak, dan Anda perlu mengomentarkannya jika Anda tidak ingin ditampilkan. Jika Anda memiliki banyak pernyataan 'logging' ini, maka Anda harus mengomentarkannya di banyak tempat dalam kode Anda, yang tidak nyaman. Alangkah baiknya jika kita dapat dengan mudah mengubah tombol, dan memilih apakah kita ingin mencetak pernyataan ini, dan di situlah ns3 logging dapat membantu.

Di ns3, banyak kelas C++ dirancang dengan fungsi logging, memungkinkan Anda untuk mengaktifkan dan menonaktifkannya. ns3 mendefinisikan beberapa jenis pesan log (atau level), misalnya 'info', 'peringatan', 'fungsi'

dan 'kesalahan'. Berikut ini adalah rincian dari level-level tersebut.

Tingkatan	Makna
LOG_NONE	default, no logging.
LOG_ERROR	Error messages
LOG_WARN	Warning messages
LOG_DEBUG	For use in debugging
LOG_INFO	Informational.
LOG_FUNCTION	Function tracing.
LOG_LOGIC	Control tracing.

3.1.1 Mengaktifkan Logging

Logging bisa diaktifkan pada komponen ns3. Suatu komponen ns3 mendukung fitur logging jika didalamnya di atur:

```
NS_LOG_COMPONENT_DEFINE ("NamaKomponen");
```

Di mana "NamaKomponen" adalah nama yang diberikan ke komponen. Ini tidak harus cocok dengan nama kelas C++, atau harus berada di kelas. Bahkan, nama dapat melakukannya di program ns3 yang Anda jalankan di bawah direktori awal. Untuk contoh:

```
#include "ns3/core-module.h"
using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("MyThirdProgram");
void SomeFunction(int x){
NS_LOG_FUNCTION ("x value" << x);
}
int main(int argc, char *argv[]) {
LogComponentEnable ("MyThirdProgram",
LOG_LEVEL_WARN);
NS_LOG_INFO ("Program Started!");
SomeFunction (5);
NS_LOG_WARN ("Warning message");
```



```
NS_LOG_ERROR ("An Error Happened");  
}
```

Kami mendefinisikan komponen dengan nama MyThirdProgram dan mengaktifkan logging pada LOG_LEVEL_WARN, artinya kami akan memicu pencetakan pesan dengan tingkat keparahan WARN atau lebih tinggi seperti yang ditunjukkan pada tabel sebelumnya.

Sebagai contoh, kita dapat mengaktifkan logging untuk ns3::UdpEchoClientApplication untuk mencetak pesan 'info'. Kode untuk itu adalah satu baris sederhana:

```
LogComponentEnable ("UdpEchoClientApplication",  
LOG_LEVEL_INFO);
```

Jika membuat kelas ns3 sendiri, disarankan untuk menentukan logging untuk mereka, dan biasakan untuk menambahkan NS_LOG_FUNCTION(ini) di semua fungsi yang Anda buat karena akan membantu melacak pemanggilan fungsi jika diperlukan.

B. Alat dan Bahan

Alat dan bahan yang dibutuhkan pada Praktikum ini adalah:

1. Perangkat Laptop/PC
2. Simulator NS-3

C. Simulasi dan Implementasi

Berikut tahapan praktikum simulasi WSN

(1) Modul Logging

- a. Arahkan ke direktori aktif ns-3.
- b. Melihat lebih banyak dari Log

```
$ export NS LOG=UdpEchoClientApplication=level all  
$ export 'NS LOG=UdpEchoClientApplication=level all |  
prefix func'  
$ export 'NS LOG=UdpEchoClientApplication=  
level all|prefix func :  
UdpEchoServerApplication=level all|prefix func'
```

```
$ export 'NS LOG=UdpEchoClientApplication=
level all|prefix func|prefix time:
UdpEchoServerApplication= level all|prefix
func|prefix time'
```

c. Log pada semua aplikasi

```
export 'NS LOG *=level all|prefix func|prefix time'
```

d. Menambahkan LOG khusus

```
$ vim scratch/myfirst.cc
NS LOG COMPONENT DEFINE ("FirstScriptExample");
NS LOG INFO("Creating Topology");
NodeContainer nodes;
nodes.Create (2);
```

e. Jalankan program \$. ./waf

f. Membersihkan LOG

```
$ export NS LOG=
```

g. Jalankan lagi

```
$ ./waf --run scratch/myfirst
```

h. Apakah muncul LOG?

i. Melihat LOG sendiri

```
$ export NS LOG=FirstScriptExample=info
$ ./waf --run scratch/myfirst
$ export NS LOG=FirstScriptExample=level info
$ ./waf --run scratch/myfirst
```

(2) Argument Perintah

a. Bantuan

```
$ ./waf --run "scratch/myfirst --PrintHelp"
```

b. Mencetak atribut atribut

```
$ ./waf --run "scratch/myfirst  
--PrintAttributes=ns3::PointToPointNetDevice"
```

c. Hapus 2 baris berikut

```
pointToPoint.SetDeviceAttribute ("DataRate",  
StringValue ("5Mbps"));  
pointToPoint.SetChannelAttribute ("Delay",  
StringValue ("2ms"));
```

d. Menentukan atribut-atribut

```
$ ./waf --run "scratch/myfirst  
--ns3::PointToPointNetDevice::DataRate=5Mbps"
```

e. Menggunakan nilai tertentu, sebelum cmd.Parse(argc, argv), tambahkan:

```
uint32_t nPackets = 1;  
cmd.AddValue("nPackets", "Jumlah Paket yang  
ditampilkan", nPackets);
```

f. Tambahkan

```
echoClient.SetAttribute ("MaxPackets", UIntegerValue  
(nPackets))
```

g. Jalankan lagi:

```
$ ./waf --run "scratch/myfirst --nPackets=4"
```

(3) Sistem pelacakan

a. Melihat header IP menggunakan LOG

System pelacakan memungkinkan sistem log yang lebih luasa untuk dilakukan kustomisasi sesuai kebutuhan

b. Tambahkan dua baris berikut sebelum Simulator::Run

```
AsciiTraceHelper ascii;  
pointToPoint.EnableAsciiAll (ascii.CreateFileStream  
("myfirst.tr"));
```

- c. Jalankan lagi skripnya
- d. Buka file myfirst.tr
- e. Pap Tracing, dengan menambahkan baris berikut sebelum *Simulator::Run*

```
pointToPoint.EnablePcapAll ("myfirst");
```

D. Ringkasan

Jobsheet ini merupakan cara menggunakan LOG pada simulator NS-3. Fitur logging pada ns3 memudahkan untuk mengidentifikasi pola node di jaringan dalam melakukan komunikasi.

E. Latihan:

Jawablah pertanyaan berikut ini.

1. Masukkan lebih banyak INFO LOG untuk *FirstScriptExample*
2. Contoh. Membuat Topologi, Menetapkan Alamat IP, Membuat echoServer
3. Berapa nilai penundaan saluran default?
4. Setel penundaan saluran ke 4 mdtk menggunakan CLA
5. Setel paket maksimal ke 4 menggunakan CLA
6. Tetapkan kecepatan data, penundaan saluran, dan paket maksimal secara bersamaan menggunakan CLA

F. Tugas:

1. Buatlah sistem dashboard yang menampilkan semua sensor yang memungkinkan dalam ponsel Anda!
2. Buatlah laporan dari Modul 1 yang menjelaskan tahapan pembuatan Sistem Dashboard untuk menampilkan dan menyimpan data dari sensor-sensor di ponsel Anda menggunakan Platform uBeac! (Menyertakan konfigurasi Gateway dan Layout Dashboard, Tugas 1, dan jawaban dari pertanyaan).

BAB 4

TOPOLOGI JARINGAN DAN ROUTING WSN

Setelah mengambil bab ini, siswa hendaknya:

1. Memahami konsep platform iot berbasis cloud
2. Memahami format pertukaran data
3. Menjelaskan cara kerja sensor

A. Pendahuluan

Sensor merupakan komponen terpenting yang dibutuhkan untuk pengumpulan data-data dalam proyek Internet of Things (IoT) dan Wireless Sensor Network (WSN). Namun, membeli setiap sensor per item dan menemukan tempat untuk menyimpannya merupakan masalah tersendiri, terutama saat kita mulai membuat lebih banyak proyek. Untungnya, setidaknya ada sepuluh sensor berteknologi tinggi dalam perangkat yang kita gunakan hampir setiap hari di ponsel pintar kita. Dari akselerometer, giroskop, sensor tekanan dan cahaya, hingga GPS, perangkat ini dapat digunakan untuk mengumpulkan data yang diperlukan untuk memulai berbagai proyek IoT.

Salah satu cara terbaik dan termudah untuk mengumpulkan, memvisualisasikan, dan menganalisis data ponsel pintar itu adalah dengan uBeac. uBeac merupakan platform IoT serbaguna untuk transformasi digital terpusat, integrasi, dan visualisasi data. Hub IoT uBeac memungkinkan untuk menghubungkan, memproses, dan memvisualisasikan data waktu nyata dengan cara yang aman. Sehubungan dengan aplikasi Data Collector, kita akan mengerjakan proyek IoT dalam waktu singkat..

Figure 0.1 (a) Original Image, (b) Green Channel, (c) Red Channel, (d) Blue Channel

B. Alat dan Bahan

Alat dan bahan yang dibutuhkan pada Praktikum ini adalah:

1. Perangkat Android
2. Platform IoT uBeac
3. Aplikasi Data Collector: [tautan](https://play.google.com/store/apps/details?id=tech.unismart.dc&hl=en)
<https://play.google.com/store/apps/details?id=tech.unismart.dc&hl=en>

C. Simulasi dan Implementasi

Digital image contains:

- (1) Arahkan pada direktori ns-3 aktif
- (2) Jalankan perintah

```
$ cp examples/tutorial/second.cc scratch/mysecond.cc
```

- (3) Topologi Bis

```
// Default Network Topology
//
//      10.1.1.0
// n0 ----- n1  n2  n3  n4
//      point-to-point |  |  |  |
//                      =====
//                      LAN 10.1.2.0
```

- (4) Jalankan skrip kedua

```
$ ./waf
$ export NS LOG=
$ ./waf --run scratch/mysecond
$ tcpdump -nn -tt -r second-0-0.pcap
$ tshark -n -t d -r second-0-0.pcap
```

- (5) Topologi Wireless

```
// Default Network Topology
//
//      Wifi 10.1.3.0
//                      AP
// *   *   *   *
// |   |   |   |
// n5  n6  n7  n0 ----- n1  n2  n3  n4
//                      point-to-point |  |  |  |
//                      =====
//                      LAN 10.1.2.0
```

- (6) Jalankan perintah

```
$ cp examples/tutorial/third.cc scratch/mythird.cc
$ vim scratch/mythird.cc
```

- (7) A
- (8) A
- (9) A
- (10) A
- (11) a

D. Ringkasan

Jobsheet ini merupakan cara menggunakan uBeac untuk membuat dasbor dari sensor yang terdapat pada ponsel. Sekarang kita dapat menggunakannya untuk berbagai proyek, seperti melacak ke mana dan bagaimana bepergian ke dan dari tempat kerja, dan mengoptimalkan kegunaan sensor yang ada dalam ponsel.

E. Latihan:

Jawablah pertanyaan berikut ini.

1. Jalankan mysecond dengan jumlah node cdma tambahan = 5
2. Setel MaxPackets dari klien gema ke 4 dan jalankan skrip lagi. Sekarang amati protokol ARP dan berapa banyak permintaan/balasan ARP yang Anda lihat?
3. Tambahkan satu lagi klien gema di csma dengan atribut serupa dan mulai dua klien secara bersamaan (2 detik). Amati RTT setiap node dan setiap transmisi.
4. Tambahkan argumen lain yang disebut nEchoClients untuk mengatur jumlah node yang menginstal aplikasi klien gema dan mengimplementasikannya dalam kode. Sekali lagi, amati RTT setiap node dan setiap transmisi!
5. Dapatkan file pcap dari mythird.cc dan buka semuanya. Lihat ARP dan menjelaskan kepada diri sendiri apa yang terjadi.
6. Tukar klien dan server (server yang berjalan di wifi dan klien yang berjalan di csma) dan amati perbedaan RTT.
7. Lewati nPackets dan nEchoClients (seperti yang kami lakukan di mysecond) dan amati dampak penambahan lebih banyak klien gema ke RTT.
8. Keras! Ubah model propagasi wifi menjadi acak (default adalah model propagasi jarak jauh).

F. Tugas:

1. Buatlah sistem dashboard yang menampilkan semua sensor yang memungkinkan dalam ponsel Anda!
2. Buatlah laporan dari Modul 1 yang menjelaskan tahapan pembuatan Sistem Dashboard untuk menampilkan dan menyimpan data dari sensor-sensor di ponsel Anda menggunakan Platform uBeac! (Menyertakan konfigurasi Gateway dan Layout Dashboard, Tugas 1, dan jawaban dari pertanyaan

BAB 5

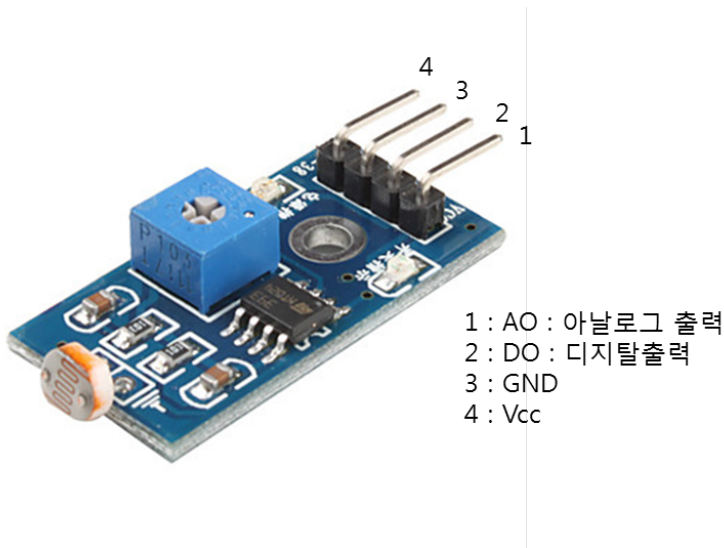
AMBIL DATA SENSOR DENGAN RASPBERRY PI

Setelah mengambil bab ini, siswa hendaknya:

1. Memahami cara kerja sensor
2. Memahami cara konversi data analog ke digital
3. Membuat program menggunakan raspberrypi

A. Pendahuluan

(1) Sensor Cahaya.



Karakteristik:

- a) Nama Model: GL5537 (5Pie Photo Resistor)
- b) Jenis: Cds Photo CELL
- c) Penyesuaian sensitivitas (penyesuaian resistor variabel biru) Mendeteksi kecerahan dan intensitas cahaya sekitar
- d) Output: output tegangan analog, output sakelar digital (0, 1)
- e) Disipasi daya: 100mW
- f) Tegangan Maks: 150Vdc atau AC
- g) Temp.Range: : -30~+70°C

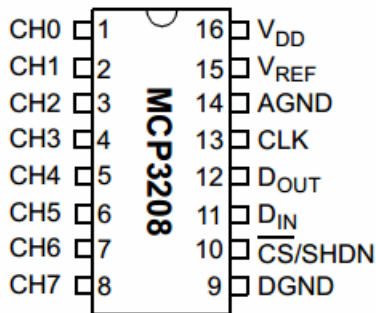
Kelistrikan:

- a) Tegangan kerja: 3.3V hingga 5V DC
- b) Keluaran: keluaran tegangan analog, keluaran sakelar digital (0, 1)
- c) Tahanan Cahaya pada 10Lux(pada 25): min 20, maks 50Kohm- Kondisi pengukuran VDD = 5V , T = 25
- d) Nilai Gamma pada 10~100Lux: 0,7 tip
- e) Perlawanan Gelap pada 0 Lux (10 detik setelah mematikan 10Lux): min 2Mohm
- f) Respon Spektrol Puncak: min 550, maks 650nm

(2) ADC

MCP3208 adalah Konverter 8-Channel 12-Bit, yaitu chip ADC yang mengubah nilai output sensor analog menjadi nilai digital 12-bit. Misalnya, mengubah nilai keluaran sensor 0 ~ 3.3v menjadi nilai digital 0 ~ 4095 (12bit). Raspberry Pi memiliki port komunikasi GPIO (General Purpose Input/Output), SPI, I2C, dan UART. Sayangnya, tidak ada fungsi ADC. Oleh karena itu, menggunakan fungsi chip eksternal ADC MCP3208. Data ADC diambil antara Raspberry Pi dan MCP3208 melalui komunikasi SPI.

MCP3208 memiliki struktur pin berikut. Di bawah ini menunjukkan fungsi Pin .



Name	Function
V _{DD}	+2.7V to 5.5V Power Supply
DGND	Digital Ground
AGND	Analog Ground
CH0-CH7	Analog Inputs
CLK	Serial Clock
D _{IN}	Serial Data In
D _{OUT}	Serial Data Out
CS/SHDN	Chip Select/Shutdown Input
V _{REF}	Reference Voltage Input

B. Alat dan Bahan

Alat dan bahan yang dibutuhkan pada Praktikum ini adalah:

1. Perangkat Android
2. Platform IoT uBeac
3. Aplikasi Data Collector: [tautan https://play.google.com/store/apps/details?id=tech.unismart.dc&hl=en](https://play.google.com/store/apps/details?id=tech.unismart.dc&hl=en)

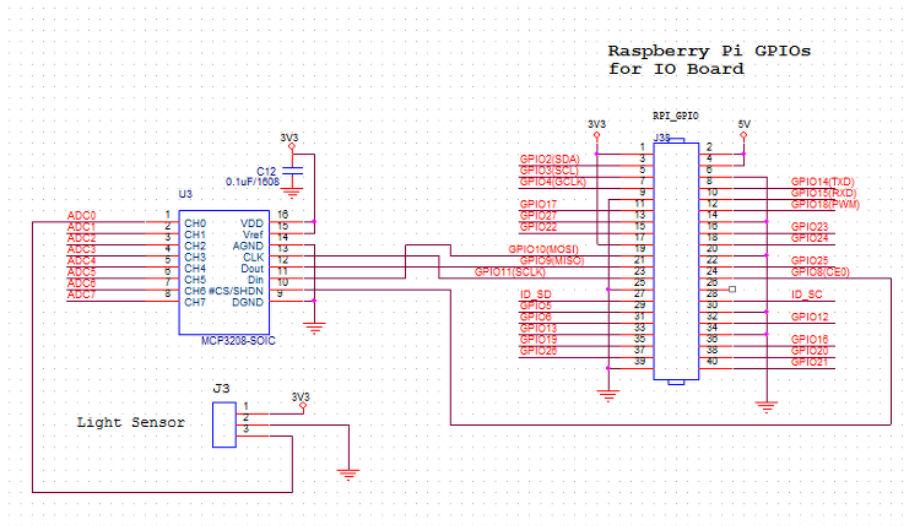
C. Simulasi dan Implementasi

Digital image contains:

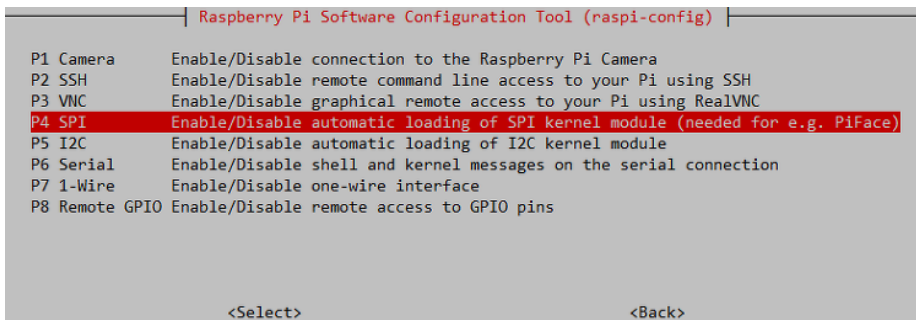
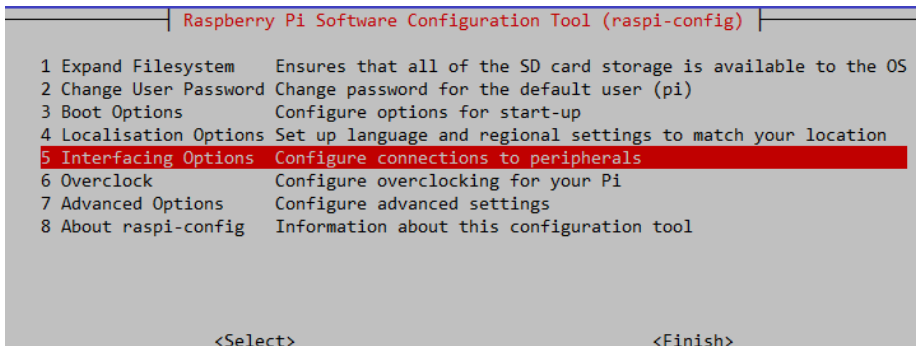
(1) Mengaktifkan fungsi SPI

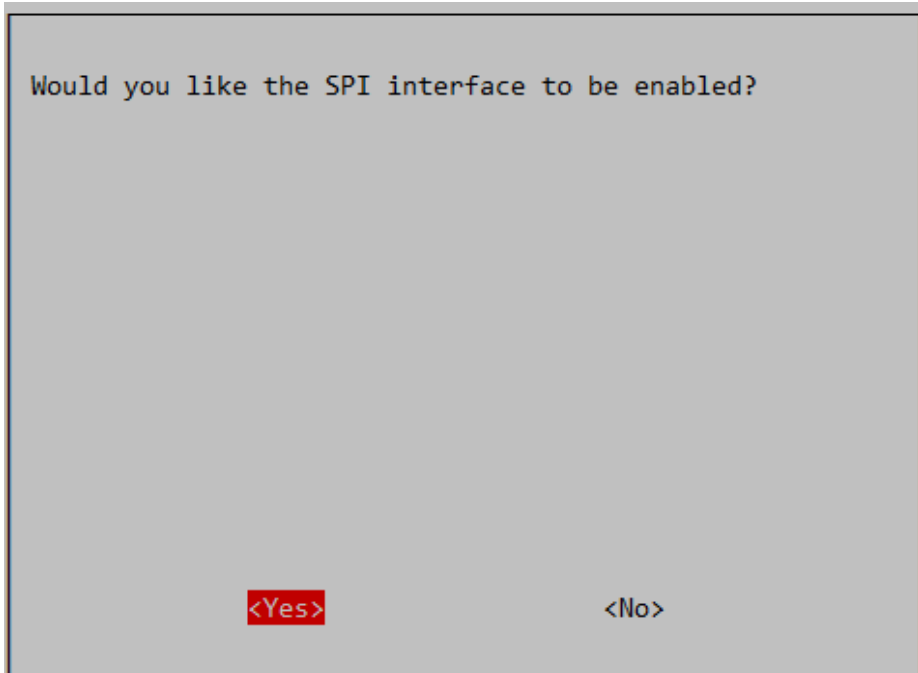
a. Rangkaian

Hubungkan ke port SPI dari Raspberry Pi seperti yang ditunjukkan pada diagram rangkaian di bawah ini. MCP3208 memiliki total 8 saluran ADC (CH0~CH7). Hubungkan output analog dari sensor cahaya ke CH0 dan uji perubahan nilai ADC sambil memutar resistor variable pada sensor. Sensor yang digunakan dapat dihubungkan ke salah satu dari CH0 hingga CH7.



b. Buka layar pengaturan:





- c. Kode Program Sensor Cahaya : buka editor nano, dan ketikkan kode berikut ini:

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <wiringPi.h>
#include <wiringPiSPI.h>

#define CS_MCP3208 8 //GPIO 8
#define SPI_CHANNEL 0
#define SPI_SPEED 1000000 //1Mhz

// komunikasi spi dengan Rpi dan mengambil data
sensor
```

```

int read_mcp3208_adc(unsigned char adcChannel)
{
    unsigned char buff[3];
    int adcValue = 0;

    buff[0] = 0x06 | ((adcChannel & 0x07) >> 2);
    buff[1] = ((adcChannel & 0x07) << 6);
    buff[2] = 0x00;

    digitalWrite(CS_MCP3208, 0);
    wiringPiSPIDataRW(SPI_CHANNEL, buff, 3);

    buff[1] = 0x0f & buff[1];
    adcValue = (buff[1] << 8 ) | buff[2];

    digitalWrite(CS_MCP3208, 1);

    return adcValue;
}

int main(void) {

    unsigned char adcChannel_light = 0;

    int adcValue_light = 0;

    float vout_light;
    float vout_oftemp;
    float percentrh = 0;
    float supsiondo = 0;

```

```

printf("mulai");

if(wiringPiSetupGpio() == -1)
{
    fprintf(stdout, "Tidak bisa menjalankan
wiringPi :%s\n", strerror(errno));
    return 1;
}

if(wiringPiSPISetup(SPI_CHANNEL, SPI_SPEED) == -1)
{
    fprintf(stdout, "wiringPiSPISetup Gagal :%s\n",
strerror(errno));
    return 1;
}

pinMode(CS_MCP3208, OUTPUT);

while(1)
{
    adcValue_light =
read_mcp3208_adc(adcChannel_light);

    //printf("Kelembaban = %u Suhu = %u\n",
adcValue_humi, adcValue_temp);
    printf("Sensor Cahaya = %u\n", adcValue_light);

    delay(100);
}

```

```
}  
    return 0;  
}
```

d. Kompilasi Program

```
$gcc -o getlightsensor getlightsensor.c -lwiringPi
```

e. Jalankan Program

```
$sudo ./getlightsensor
```

f. Tutup dan buka Sensor cahaya dan lihat tampilan layar!

```
light sensor = 436  
light sensor = 436  
light sensor = 436  
light sensor = 436  
light sensor = 436  
light sensor = 437  
light sensor = 437  
light sensor = 436  
light sensor = 437
```

(a)

```
light sensor = 1272  
light sensor = 1273  
light sensor = 1274  
light sensor = 1273  
light sensor = 1271  
light sensor = 1271  
light sensor = 1272  
light sensor = 1272  
light sensor = 1272  
light sensor = 1275
```

(b)

```
light sensor = 3113  
light sensor = 3114  
light sensor = 3113  
light sensor = 3113  
light sensor = 3114  
light sensor = 3112  
light sensor = 3112  
light sensor = 3112  
light sensor = 3114  
light sensor = 3109
```

(c)

Kondisi saat cahaya (a) sangat terang, (b) normal, dan (c) gelap.

g.

(2) A

D. Ringkasan

Jobsheet ini merupakan cara menggunakan uBeac untuk membuat dasbor dari sensor yang terdapat pada ponsel. Sekarang kita dapat menggunakannya untuk berbagai proyek, seperti melacak ke mana dan bagaimana bepergian ke dan dari tempat kerja, dan mengoptimalkan kegunaan sensor yang ada dalam ponsel.

E. Latihan:

Jawablah pertanyaan berikut ini.

1. Apakah perbedaan antara format JSON dengan CSV?

2. Sebutkan platform IoT selain uBeac!
3. Sebutkan disertai penjelasan masing-masing menu pada platform uBeac!
4. Sebutkan dan jelaskan masing-masing sensor yang terdapat pada ponsel Anda!
5. Jelaskan kelebihan dan kekurangan dari uBeac!

F. Tugas:

1. Buatlah sistem dashboard yang menampilkan semua sensor yang memungkinkan dalam ponsel Anda!
2. Buatlah laporan dari Modul 1 yang menjelaskan tahapan pembuatan Sistem Dashboard untuk menampilkan dan menyimpan data dari sensor-sensor di ponsel Anda menggunakan Platform uBeac! (Menyertakan konfigurasi Gateway dan Layout Dashboard, Tugas 1, dan jawaban dari pertanyaan

BAB 6

CHATBOT UNTUK IoT

Setelah mengambil bab ini, siswa hendaknya:

1. Memahami konsep platform iot berbasis cloud
2. Memahami format pertukaran data
3. Menjelaskan cara kerja sensor

A. Pendahuluan

Bot merupakan robot web atau bisa kita sebut juga asisten dengan kecerdasan buatan yang digunakan mulai dari web, pesan singkat dan lain-lain. Salah satu aplikasi yang memiliki sebuah penyedia membuat bot adalah telegram. Telegram merupakan aplikasi layanan pengiriman pesan, aplikasi dengan kegunaan yang multi fungsi yang sering digunakan dalam berbisnis seperti sebuah jasa penjualan pulsa prabayar untuk telepon seluler. Namun seiring perkembangan waktu bot pada telegram sendiri digunakan pada banyak hal yang tidak berhubungan dengan bisnis, melainkan dapat digunakan untuk kebutuhan sehari-hari seperti alarm, e-book dan lain-lain. Namun pembuatan bot sendiri harus melalui sebuah bot lagi yaitu BotFather yang mana kegunaannya untuk mendapatkan API token. API token yang telah didapatkan guna untuk mendapatkan hak akses kepada bot yang telah dibuat. Bot pada telegram dapat digunakan dengan fungsi-fungsi yang telah disediakan oleh telegram dan ada juga yang menggunakan manual atau menggunakan bahasa pemrograman yang lain seperti PHP, python, ruby dll. Pada pemrograman sebuah bot menggunakan PHP memang memiliki sebuah kelebihan dan kekurangan, karena pada saat pembaruan pada aplikasi telegram sebuah error pada pemrograman dapat terjadi. Bot sendiri memiliki kepintaran sendiri jadi pada saat pembaruan bot akan terus mempelajari sebuah hal yang baru. Namun hal yang penting dalam pembuatan bot ada dua yaitu membuat bot melalui BotFather yang ada pada telegram

dan yang kedua adalah membuat engine bot agar fungsi-fungsi yang telah dibuat dapat digunakan sesuai kerjanya.

B. Alat dan Bahan

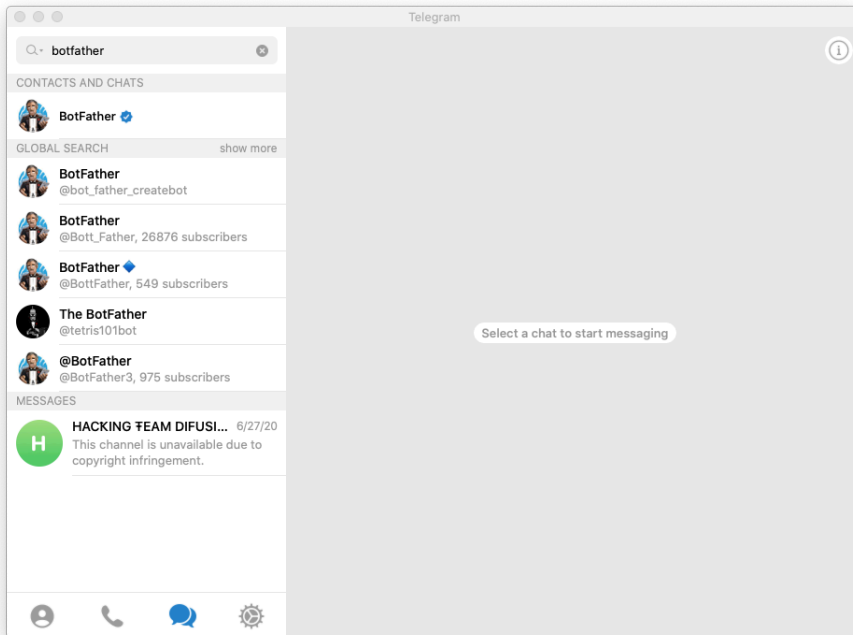
Alat dan bahan yang dibutuhkan pada Praktikum ini adalah:

1. Perangkat Android
2. Telegram
3. Aplikasi Data Collector: tautan <https://play.google.com/store/apps/details?id=tech.unismart.dc&hl=en>

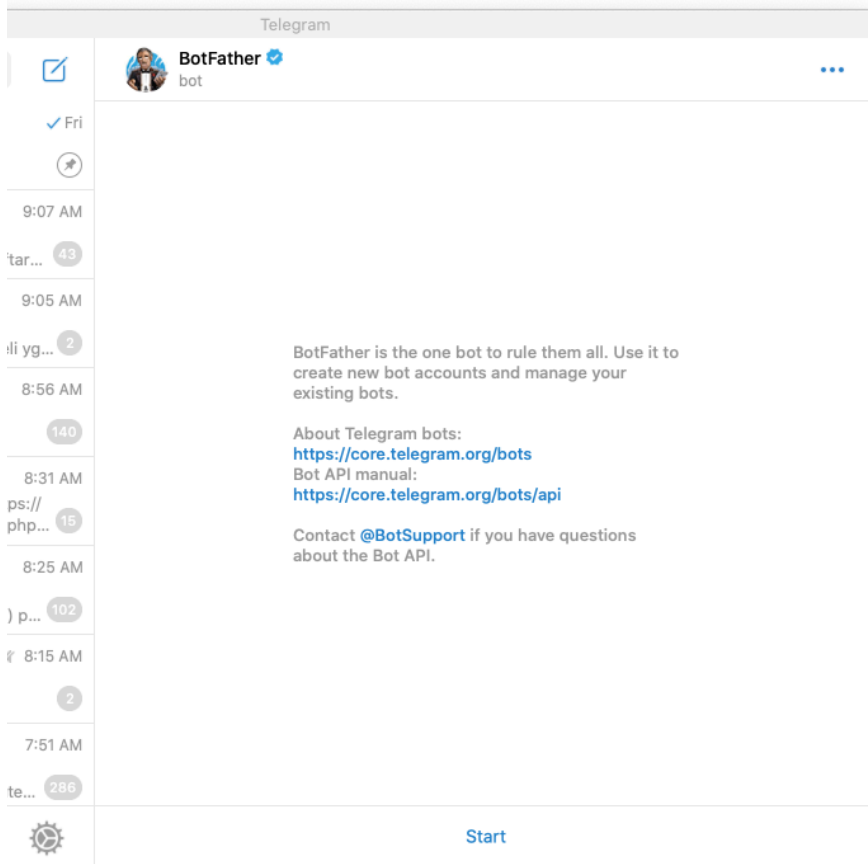
C. Simulasi dan Implementasi

Berikut cara membuat ChatBot telegram dan menghubungkannya dengan Mikrokontroler:

- (1) Cari akun @BotFather pada menu pencarian, hingga muncul tampilan chat dengan @BotFather tersebut. Kemudian user akan diminta untuk menekan tombol "START" atau "Mulai".



Gambar 6.1 Tampilan pencarian akun telegram

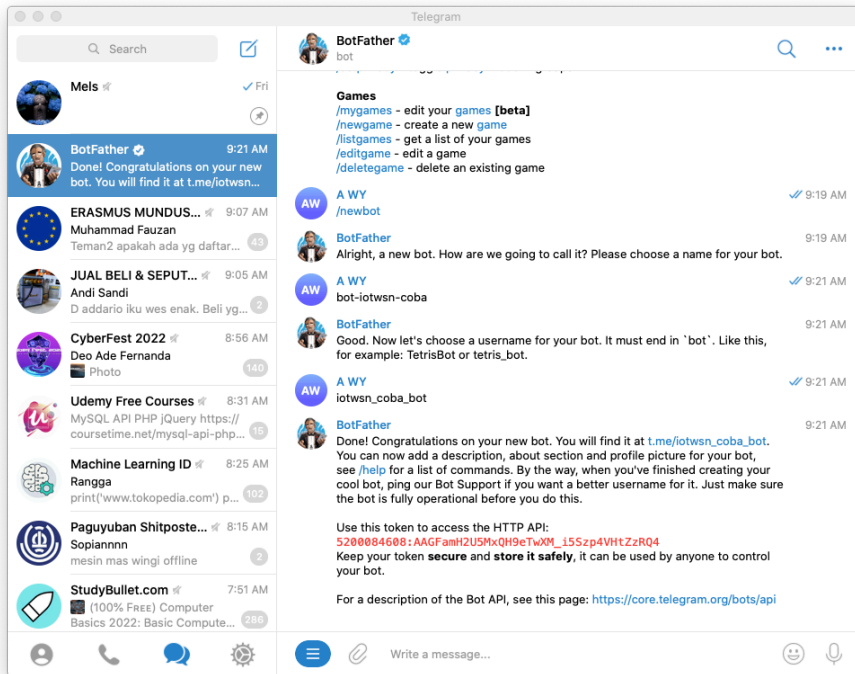


Gambar 6.2 Halaman awal BotFather

(2) Brightness

Setelah user menekan tombol "MULAI" maka akan tampil pesan balasan dari @BotFather yaitu perintah yang dapat oleh @BotFather.

(3) Kirim perintah "/newbot", kemudian @BotFather akan membalas pesan user untuk memasukkan nama bot yang akan digunakan.



Gambar 6.3 Konfirmasi Bahwa bot baru berhasil dibuat

- (4) Bot Telegram yang telah selesai dibuat akan tetapi bot Telegram tersebut masih belum bisa merespon perintah apapun karena API bot Telegram belum dimasukkan ke program Arduino IDE. Untuk tahap selanjutnya yaitu mengintegrasikan bot Telegram yang sudah dibuat dengan WemosD1, agar bot Telegram dapat mengeksekusi perintah dari user.
- (5) Buka Arduino IDE lalu download CTBot Telegram library. CTBot berfungsi sebagai penghubung untuk memprogram board ESP32 dengan bot Telegram.
- (6) Pada poin 1 kita memilih menu sketch, lalu pilih sub menu include library. Lalu akan muncul pilihan pada sub menu include library, setelah itu pilih manage libraries.
- (7) Setelah memilih manage libraries, lalu muncul seperti pada poin 6 tersebut lalu kita menuliskan CTBot pada pencarian.
- (8) Lalu akan muncul library CTBot tersebut, setelah muncul pilih version CTBot tersebut, selanjutnya klik instal.
- (9) Jika telah terinstal maka terdapat tulisan "INSTALLED".

- (10) Setting board ESP8266 sebagai board mikrokontroler yang akan kita gunakan dan untuk di program dengan cara pilih menu "tools", kemudian pilih "boards", setelah itu pilih "Wemos D1 R2 & mini,Äü.
- (11) Setelah port dan board sudah sesuai dengan setting yang ada pada aplikasi Arduino IDE. Langkah selanjutnya yaitu melakukan coding sketch program.
- (12) Setelah sketch program selesai maka sebelum kita upload sketch tersebut kita akan melakukan compile program dengan cara menekan tombol ikon centang pada aplikasi Arduino IDE seperti pada Gambar 4.4 berikut.
- (13) Langkah terakhir adalah melakukan compile program, jika tidak terjadi eror maka program tersebut dapat dinyatakan sudah benar. Jika sudah dipastikan tidak terjadi eror pada sketch program tersebut maka program tersebut bisa di upload ke Wemos D1 dengan cara menekan tombol panah.



Gambar VI 4 Tampilan Menu compile dan upload

D. Ringkasan

Jobsheet ini merupakan cara membuat chatbot telegram dan menghubungkannya ke perangkat keluarga esp32.

E. Latihan:

Jawablah pertanyaan berikut ini.

1. Jelaskan cara kerja chatbot

F. Tugas:

1. Buatlah chatbot untuk memonitoring data dari sensor!
2. Buatlah laporan dari Modul 6 yang menjelaskan tahapan pembuatan chatbot dan menghubungkannya dengan mikrokontroler esp32! (Menyertakan konfigurasi Gateway dan Layout Dashboard, Tugas 1, dan jawaban dari pertanyaan.

BAB 7

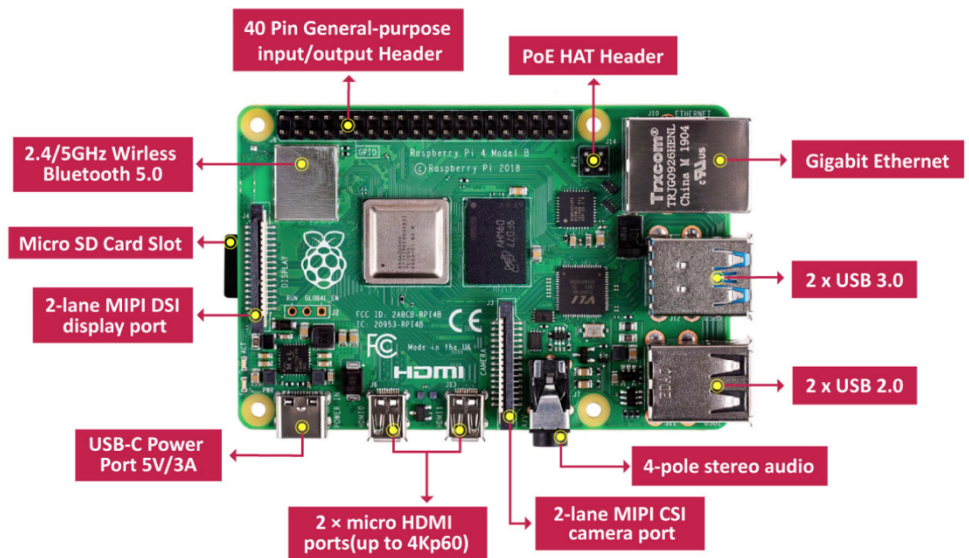
GATEWAY IoT DAN WSN

Setelah mengambil bab ini, siswa hendaknya:

1. Memahami konsep platform iot berbasis cloud
2. Memahami format pertukaran data
3. Menjelaskan cara kerja sensor

A. Pendahuluan

Raspberry Pi adalah komputer berdimensi kecil dengan mikrokontroler tunggal/ Single Board Computer. Dengan RAM yang lebih tinggi dari 256MB atau 512Mb, tergantung pada modelnya, dapat menangani tugas yang lebih kompleks dibandingkan pengontrol lain dan sebagian besar digunakan sebagai pusat unit pemrosesan untuk beberapa perangkat. Sebagian besar model baru Raspberry Pi memiliki port USB dan Ethernet, sehingga memudahkannya mengunggah data ke internet.



Gambar 7.1 Arsitektur Raspberrypi

B. Alat dan Bahan

Alat dan bahan yang dibutuhkan pada Praktikum ini adalah:

1. Raspberry Pi 4
2. Raspbian OS
3. MicroSD
4. Putty
5. Html, Php, JSON, Mysql
6. Python, Flask, Redis

C. Simulasi dan Implementasi

Digital image contains:

(1) Instalasi Instalasi

- a. Berikut ini cara instalasi Auto OS Raspbian pada Raspberry Pi di Windows:

```
Invoke-WebRequest -Uri  
https://downloads.raspberrypi.org/imager/imager_lates  
t.exe -OutFile .\imager_latest.exe  
.\imager_latest.exe
```

- b. Pilih Operating System

- c. Pemilihan Images

- i. Pilih Images yang telah didownload
- ii. Erase - Format card sebagai FAT32
- iii. Use Custom – Pilih kustom .img dari komputer.

(2) Pilih Storage

- (3) Raspberry Pi Imager akan memilih opsi tanpa merusak sistem disk. Pastikan tidak ada media penyimpanan lain yang terhubung.

(4) Advanced Options

```
CTRL + SHIFT + X
```

- (5) Advanced Options memberikan akses konfigurasi lanjut

- a. Disable overscan
- b. Set hostname

- i. .local
- c. Mengaktifkan SSH
 - i. Menggunakan password authentication
 - ii. Mengijinkan hanya public-key authentication saja
- d. Configure WiFi
 - i. SSID
 - ii. Password
 - iii. WiFi Country
- e. Set locale settings
 - i. Time Zone
 - ii. Keyboard Layout
- f. Skip first run wizard
 - i. Persistent settings
- g. Play sound ketika selesai
- h. Eject media ketika selesai
- i. Mengaktifkan telemetry
- j. [Save]

- (6) Cara instalasi Manual OS Raspbian pada Raspberry Pi.
- a. Download file ISO Raspbian pada link berikut <https://www.raspberrypi.org/downloads/raspbian/> dan extract file zip tersebut sehingga mengeluarkan file dalam bentuk .img Raspbian Jessie.
 - b. Download Win32 Disk Imager. Selanjutnya install hingga selesai.
 - c. Masukkan micro SD Card yang akan dijadikan hard disk bagi Raspberry Pi ke dalam laptop melalui card reader. Selanjutnya format micro SD Card terlebih dahulu sebelum digunakan.

(7) Konfigurasi

- (8) Setelah proses selesai, keluarkan kartu SD dari komputer Anda dan masukkan ke dalam Raspberry Pi. Hubungkan Raspberry Pi ke router Anda menggunakan kabel Ethernet, dan colokkan pengisi daya Micro USB, yang akan memulai booting Raspberry Pi.

- (9) Untuk konfigurasi awal, ada dua pilihan:

- (10) Jika memiliki keyboard USB dan monitor HDMI, kita dapat menghubungkannya ke Raspberry Pi untuk pengaturan awal.
- Pi harus mengenali perangkat ini segera setelah dicolokkan.
 - Pertama kali Pi boot, itu akan berjalan secara otomatis raspi-config. Setelah boot pertama, harus menjalankan `sudo raspi-config` sendiri untuk mengonfigurasi perangkat.
- (11) Jika tidak memilikinya, dapat terhubung ke Raspberry Pi saat menggunakan SSH:
- Pertama, perlu menemukan alamat IP Raspberry Pi Anda di jaringan lokal. Ini dapat dilakukan dengan menghubungkan ke halaman admin router, atau dengan menggunakan alat jaringan seperti nmap .
 - Setelah memiliki alamat IP perangkat, sambungkan menggunakan SSH dari terminal Anda (atau melalui Putty jika menggunakan Windows). Pengguna default adalah pi, dan kata sandi default adalah raspberry. Jadi, misalnya, jika alamat IP adalah 192.168.1.16, jalankan `ssh pi@192.168.1.16` dan masukkan kata sandi saat diminta.
 - Saat terhubung, jalankan `sudo raspi-config`.
 - Raspi-config akan memandu Anda melalui pengaturan akhir. Anda dapat mengonfigurasi semua opsi tetapi yang paling penting adalah dua yang pertama: untuk memperluas sistem file, memastikan bahwa semua penyimpanan kartu SD tersedia untuk OS, dan untuk mengubah kata sandi untuk pengguna Pi default, sehingga server Anda akan terlindungi dari penyusup.
- (12) Instalasi Web Server
- Selanjutnya, akan menginstal server web. Menggunakan Nginx karena memiliki jejak memori yang kecil, dan karena ia berfungsi baik dengan Node.js (yang akan siap nanti). Server web lain, seperti Apache atau lighttpd , akan

bekerja dengan baik, tetapi kami akan menggunakan Nginx untuk demonstrasi ini.

- a. Sebelum Anda mulai menginstal apa pun, Anda harus memastikan semuanya mutakhir dengan menjalankan perintah ini di Pi:

```
sudo apt-get update  
sudo apt-get upgrade
```

- b. Kemudian dapat menginstal Nginx menggunakan apt-get:

```
sudo apt-get install nginx
```

- c. Setelah instalasi selesai, mulai server dengan menjalankan:

```
sudo service nginx start
```

- d. Jika tidak perlu mengetahui IP lokal Raspberry Pi di langkah sebelumnya, saatnya untuk mengetahuinya dengan menjalankan ifconfig. Output untuk adaptor ethernet akan berada di bawah eth0, dan dengan alamat IP lokalnya diberi label inet addr.
- e. Setelah mengetahui alamat IP, dapat mengarahkan browser komputer Anda ke sana, di mana akan melihat pesan default "Welcome to Nginx".

(13) Langkah 4: Instalasi Halaman Server

- a. Instalasi PHP
- b. Instalasi Python

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install Python
sudo apt-get install Python-pip
sudo pip install Flask
```

(14) Script REST Server

D. Ringkasan

Jobsheet ini merupakan cara Eksperimen lab ini menunjukkan kepada Mahasiswa bagaimana Anda dapat membuat Gateway Sistem IoT dan WSN.

E. Latihan:

Jawablah pertanyaan berikut ini.

1. Sebutkan dan jelaskan spesifikasi perangkat dalam bentuk tabel!
2. Sebutkan dan jelaskan perintah dasar Linux Raspbian!
3. Jelaskan tentang perangkat Raspberry dan emulatoarnya!
4. Jelaskan dasar pemrograman Python!

F. Tugas:

1. Buatlah sistem dashboard yang menampilkan semua sensor yang memungkinkan dalam ponsel Anda!
2. Buatlah laporan dari Modul 7 yang menjelaskan tahapan pembuatan Gateway! (Menyertakan konfigurasi Gateway dan Layout Dashboard, Tugas 7, dan jawaban dari pertanyaan

BAB 8

APLIKASI IoT M5STACK KIT

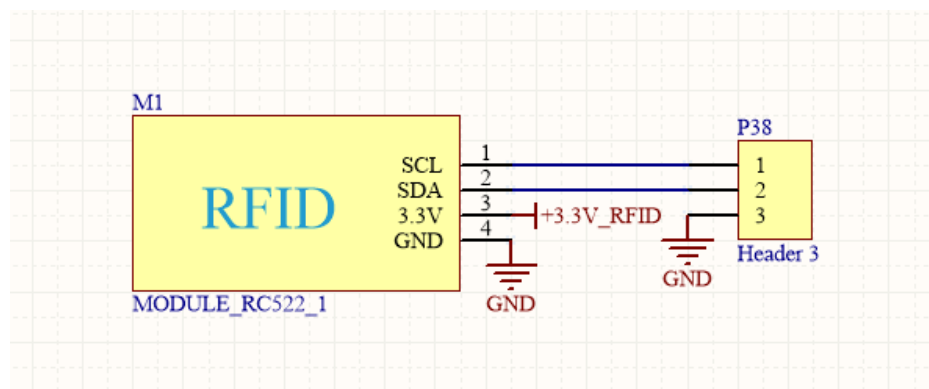
Setelah mengambil bab ini, siswa hendaknya:

1. Memahami konsep platform iot berbasis cloud
2. Memahami format pertukaran data
3. Menjelaskan cara kerja sensor

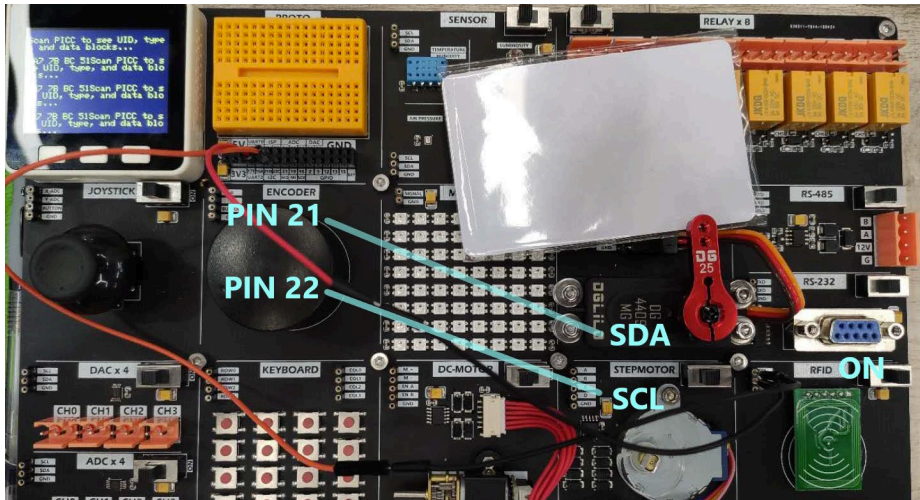
A. Pendahuluan

Modul RFID mengintegrasikan chip identifikasi frekuensi radio RC522 dan menggunakan protokol I2C untuk transmisi data. Alamat I2C adalah (0x28)

Frekuensi kerja adalah 13.56MHz. Mendukung beberapa fungsi seperti membaca kartu, menulis kartu, identifikasi, perekaman, pengkodean, dan otorisasi kartu RF.



Rangkaian Modul RFID RC522



B. Alat dan Bahan

Alat dan bahan yang dibutuhkan pada Praktikum ini adalah:

1. Perangkat PC/Laptop
2. Papan M5Stack IoT Developer Kit

C. Simulasi dan Implementasi

Digital image contains:

(1) Matrix Led

```
#include <M5Stack.h>
#include <FastLED.h>
/**
 * Catatan: Install library Fastled terlebih dahulu.
 *
 ***/
#define DATA_PIN 15
#define LED_TYPE WS2811
#define COLOR_ORDER GRB
#define NUM_LEDS 64
CRGB leds[NUM_LEDS];
```



```

#define BRIGHTNESS 5
void setup() {
M5.begin();
M5.Lcd.setCursor(120, 110, 4);
M5.Lcd.println("MATRIX");
FastLED.addLeds<LED_TYPE,DATA_PIN,COLOR_ORDER>(leds,
NUM_LEDS).setCorrection(TypicalLEDStrip);
FastLED.setBrightness(BRIGHTNESS);
}
void loop() {
for(int i = 0; i < 64; i++){
leds[i] = CRGB::White;
FastLED.show();
}
delay(500);
// Now turn the LED off, then pause
for(int i = 0; i < 64; i++){
leds[i] = CRGB::Black;
FastLED.show();
}
delay(500);
}

```

(2) RFID

```

#include <Wire.h>
#include "MFRC522_I2C.h"
#include <M5Stack.h> // 0x28 is i2c address on SDA.
Check your address with i2cscanner if not match.
MFRC522 mfrc522(0x28); // Create MFRC522 instance.
void setup() {

```

```

M5.begin();
Wire.begin();
M5.Lcd.setCursor(140, 0, 4);
M5.Lcd.println("RFID");
mfr522.PCD_Init(); // Init MFRC522
ShowReaderDetails(); // Show details of PCD - MFRC522
Card Reader details
Serial.println(F("Scan PICC to see UID, type, and
data blocks..."));
M5.Lcd.setCursor(0,30,2);
M5.Lcd.println("Scan PICC to see UID, type, and data
blocks...");
}
void loop() {
// Look for new cards, and select one if present
if (!mfr522.PICC_IsNewCardPresent() || !
mfr522.PICC_ReadCardSerial()) {
delay(50);
return;
}
// Now a card is selected. The UID and SAK is in
mfr522.uid.
// Dump UID
Serial.print(F("Card UID:"));
M5.Lcd.println(" ");
for (byte i = 0; i < mfr522.uid.size; i++) {
Serial.print(mfr522.uid.uidByte[i] < 0x10 ? " 0" : "
");
Serial.print(mfr522.uid.uidByte[i], HEX);
M5.Lcd.print(mfr522.uid.uidByte[i] < 0x10 ? " 0" : "

```

```
");  
M5.Lcd.print(mfrc522.uid.uidByte[i], HEX);  
}  
Serial.println();  
}
```

D. Ringkasan

Jobsheet ini merupakan cara M5Stack untuk membangun aplikasi IoT.

E. Latihan:

Jawablah pertanyaan berikut ini.

1. Apakah perbedaan antara format JSON dengan CSV?
2. Sebutkan kelebihan dan kekurangan penggunaan M5Stack

F. Tugas:

1. Buatlah sistem dashboard yang menampilkan semua sensor yang memungkinkan dalam ponsel Anda!
2. Buatlah laporan dari Modul 8 yang menjelaskan tahapan pembuatan Sistem Dashboard untuk menampilkan dan menyimpan data dari sensor-sensor di ponsel Anda menggunakan Platform uBeac! (Menyertakan konfigurasi Gateway dan Layout Dashboard, Tugas 1, dan jawaban dari pertanyaan

BAB 9

PLATFORM NODE-RED dan MQTT pada Raspberry Pi

Setelah mengambil bab ini, siswa hendaknya:

1. Memahami cara instalasi dan kerja Node-Red
2. Memahami cara instalasi dan kerja MQTT
3. Membuat konfigurasi dan menjalankan pertukaran data menggunakan Node-red , mosquito pada Raspberry Pi

A. Pendahuluan

NodeRed merupakan piranti lunak untuk membangun Internet of Things. NodeRed memiliki fitur untuk menyediakan API dan pemrograman berbasis visual. NodeRed memungkinkan untuk pembuatan antar muka grafis untuk keperluan pembuatan aplikasi dashboard.

Node-Red adalah perangkat lunak IBM yang memungkinkan kita membuat aplikasi melalui sistem visual (dengan blok). Jika ingin informasi lebih lanjut tentang perangkat lunak ini, node-red memiliki situs webnya sendiri <http://nodered.org/> .

Jika ingin mencoba perangkat lunak ini dapat menginstalnya melalui layar baris perintah dengan perintah:

```
sudo apt-get install nodered
```

MQTT

Messaging Queuing Telemetry Protocol, atau MQTT, adalah protokol pesan ringan yang dirancang untuk perangkat dan lingkungan bandwidth rendah. Ini adalah solusi yang cepat, mudah diterapkan, dan hemat energi untuk menghubungkan dan bertukar informasi antara beberapa perangkat IoT. Ini membutuhkan broker/server MQTT dan topik yang pesannya diterbitkan dan dilanggan oleh perangkat atau pengguna untuk mengirim atau menerima informasi.

B. Alat dan Bahan

Alat dan bahan yang dibutuhkan pada Praktikum ini adalah:

1. Perangkat PC/Laptop/Raspberry
2. Node-red
3. MQTT Broker
4. Aplikasi klien MQTT
5. Koneksi Internet

C. Simulasi dan Implementasi

Untuk menggunakan nodered, maka langkah pertama adalah instalasi

(1) Langkah instalasi node-red

```
pi@raspberrypi:~ $ sudo apt-get update && sudo apt-get upgrade
pi@raspberrypi:~ $ sudo apt-get install nodered
pi@raspberrypi:~ $ sudo systemctl enable nodered.service
pi@raspberrypi:~ $ sudo reboot
```

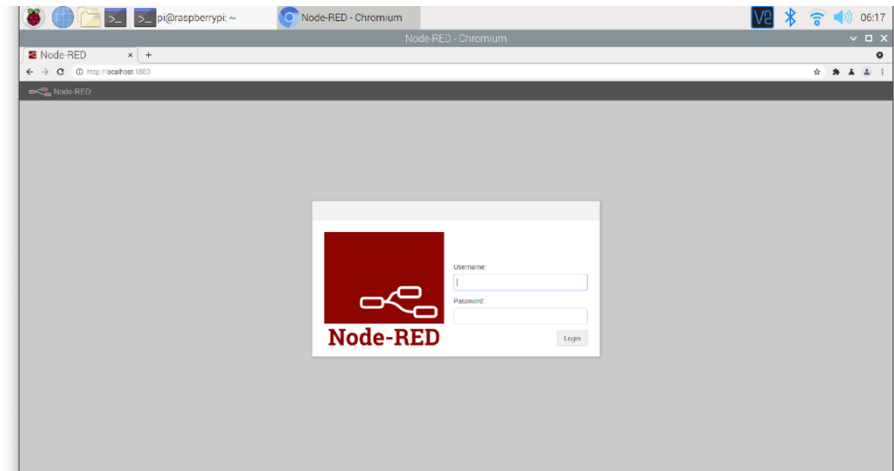
(2) Buat file pengaturan node-red dengan menjalankan perintah berikut ini.

```
node-red admin init
```

- a. Tekan Enter
- b. Jika akan mengaktifkan fitur keamanan maka tekan **Yes**. Masukkan username dan password
- c. Pilih Full Access
- d. Bisa untuk menambahkan user lainnya atau jika tidak ingin menambah user lain maka pilih No.
- e. Pilih No jika ditanya tentang mengaktifkan fitur project.
- f. Tekan Enter
- g. Ketikkan katafrasa dan simpan
- h. Pilih default theme sebagai Monaco(default)
- i. Pilih No Ketika ditanya tentang pemanggilan node external.
- j. Restart node-red dengan perintah berikut:

```
node-red restart
```

- (3) Node-red telah berhasil di install , selanjutnya buka web browser dan masukkan alamat IP dari raspberry disertai port 1880 untuk membuka halaman panel admin seperti gambar berikut



Gambar 9. Tampilan Node-Red berhasil di install

Instalasi MQTT Broker

- (4) Install MQTT Broker dengan menjalankan perintah berikut ini

```
sudo apt install mosquitto mosquitto-clients -y
```

Perintah ini akan melakukan instalasi baik Mosquitto broker maupun layanan pada klien nya.

- (5) Setelah proses instalasi selesai lakukan aktivasi layanan Mosquitto menggunakan perintah berikut ini:

```
sudo systemctl enable mosquitto.service
```

Kemudian diikuti perintah:

```
mosquitto -v
```

- (6) Buat akun user untuk mengizinkan akses secara remote ke gateway dengan menjalankan perintah berikut ini

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd  
namauser
```

Disini **namauser** merupakan username (sesuai username yang dikehendaki). Ketika muncul dialog memasukkan password maka masukkan password. Catat username dan password untuk digunakan kemudian.

- (7) Buka file **mosquito.conf** menggunakan editor nano dengan menjalankan perintah berikut ini:

```
sudo nano /etc/mosquitto.conf
```

Salin kode berikut kedalam file

```
per_listener_settings true  
pid_file /run/mosquitto/mosquitto.pid  
persistence true  
persistence_location /var/lib/mosquitto/  
log_dest file /var/log/mosquitto/mosquitto.log  
include_dir /etc/mosquitto/conf.d  
allow_anonymous false  
listener 1883  
password_file /etc/mosquitto/passwd
```

Tekan Ctrl+X selanjutnya Y untuk menyimpan perubahan file.

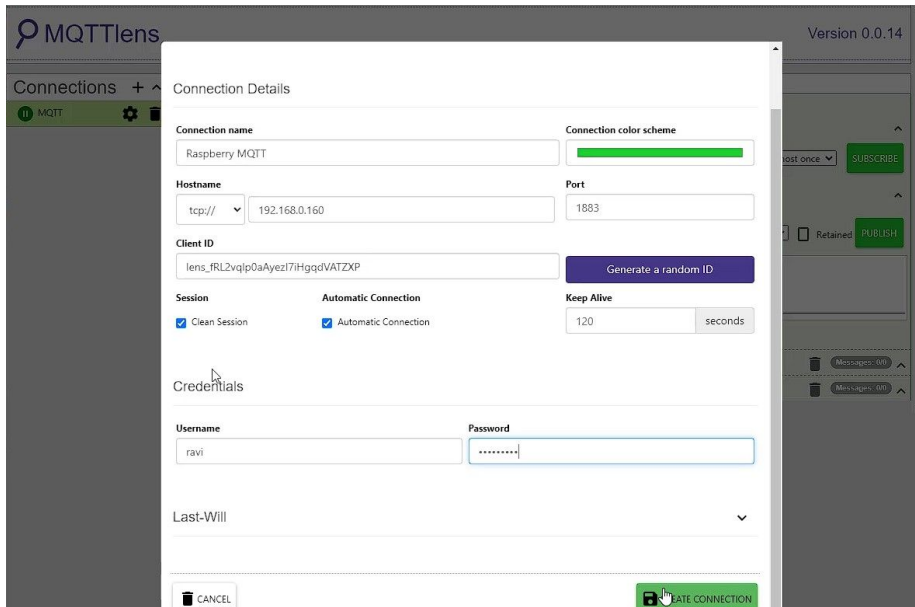
- (8) Lakukan Restart broker MQTT dengan menjalankan perintah berikut ini:

```
sudo systemctl restart mosquitto
```

- (9) Cek status broker MQTT dengan menjalankan perintah berikut:

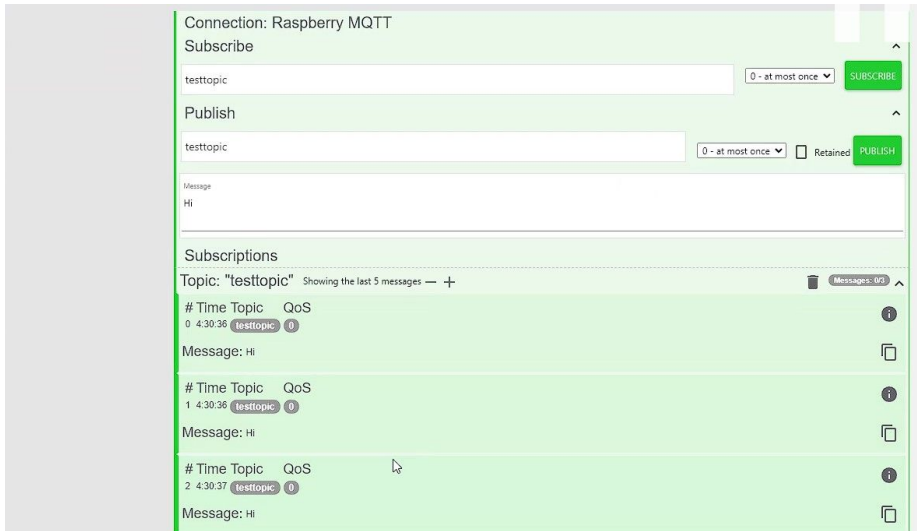
```
sudo systemctl status mosquitto
```


- (10) Uji Broker MQTT dengan melakukan instalasi MQTT app di chrome salah satunya bisa menggunakan **MQTTLens**.
- (11) Masukkan alamat IP dari raspberry dan username password MQTT yang sudah dibuat sebelumnya. Klik **Create Connection**



Gambar 9. Aplikasi MQTTLens

- (12) Masukkan topik pada kolom Subscribe dan juga topik yang sama pada kolom Publish. Kirimkan pesan Hi atau pesan apapun dan klik Publish.



Gambar 9 Tampilan Pesan MQTT

D. Ringkasan

Jobsheet ini membahas mengenai penggunaan platform Node-Red dengan konfigurasi MQTT untuk membangun aplikasi IoT. Dalam panduan ini, telah dibahas penggunaan broker MQTT dan piranti lunak Node-RED di Raspberry Pi untuk menghubungkan perangkat IoT pintar yang mendukung MQTT dan menggunakannya untuk implementasi IoT.

E. Latihan:

Jawablah pertanyaan berikut ini.

1. Apakah perbedaan antara MQTT server dengan klien?
2. Port berapakah yang digunakan dalam komunikasi MQTT?

F. Tugas:

Jawablah pertanyaan berikut ini.

1. Buatlah konfigurasi MQTT dan node red baru dengan akun nama depan anda diikuti tanggal lahir.
2. Buatlah laporan Modul 9 disertai hasil Analisa dan kesimpulan.

BAB 10

LoRA

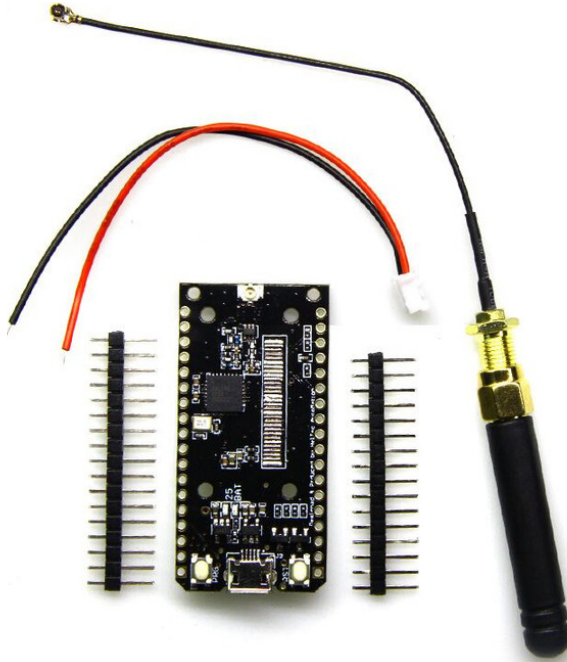
Setelah mengambil bab ini, mahasiswa hendaknya:

1. Memahami cara instalasi dan kerja LoRa
2. Memahami format pertukaran data menggunakan LoRa
3. Mampu membuat system komunikasi menggunakan LoRa

A. Pendahuluan

LoRa, adalah teknik modulasi RF yang memungkinkan sistem komunikasi nirkabel berdaya sangat rendah. Karena frekuensi yang digunakan adalah orde besarnya lebih kecil daripada metode komunikasi nirkabel umum lainnya, seperti Wi-Fi, jangkauan transmisi jauh lebih besar; hingga 15km atau lebih di daerah pedesaan dengan garis pandang yang jelas/ Line of Sight. Menggabungkan transmisi jarak jauh ini dengan kebutuhan daya yang sangat rendah menjadikan LoRa pilihan ideal untuk perangkat Internet of Things (IoT) bertenaga baterai dengan aplikasi yang memerlukan komunikasi dengan sistem yang didistribusikan secara luas.

LoRaWAN® adalah protokol berdasarkan spesifikasi open-source. Singkatnya, LoRaWAN adalah contoh jaringan area luas berdaya rendah yang didasarkan pada modulasi radio LoRa. Standar ini memungkinkan koneksi nirkabel perangkat (node) berbasis LoRa ke internet dan membantu memastikan kompatibilitas perangkat di antara produsen di seluruh dunia. Bahkan mungkin, seperti yang akan kami uraikan dalam panduan ini, untuk membangun simpul akhir LoRa kustom Anda sendiri yang dapat mengirimkan informasi ke perangkat LoRa lain dan internet melalui server jaringan LoRaWAN (LNS).



Gambar 10. TTGO LoRa ESP32 SX1276 Bluetooth WIFI Antenna (Whitout LCD) - 915MHz

TTGO LoRa32

Dicetak dalam PCB warna hitam dengan memiliki beberapa varian diantaranya:

- LoRa (tanpa 32): dengan antena PCB Wifi/Bluetooth terpasang di atasnya. Tampaknya hanya tersedia untuk 433MHz. Juga tersedia tanpa layar.
- LoRa32 V1: dengan antena Wifi/Bluetooth logam terpasang di bagian bawah. Konektor I-Pex terletak di atas, memiliki modul transceiver SX1276 LoRa, dan layar OLED yang dikendalikan oleh SSD1306. Ini akan menjadi pilihan khusus kami.
- LoRa32 V2: dengan antena Wifi/Bluetooth logam terpasang di bagian bawah (di lokasi berbeda).

Menggunakan ESP32-Pico-D4 (dengan memori flash terintegrasi) alih-alih ESP32, menggunakan modul LoRa (terlindung), konektor I-Pex terletak di bagian bawah, konektor micro-USB diputar 90 derajat, selain itu memiliki

kartu micro-SD slot di bagian bawah dan tombol on/off untuk baterai di sebelah konektor micro-USB. Mengalihkan baterai saja sehingga tidak memungkinkan untuk mematikan papan saat terhubung ke USB untuk mengisi daya baterai.

LED 'Dapat Diprogram' pada pin 22 alih-alih pin 25 tetapi tidak berguna karena terhubung ke SCL dan ketiga LED berada di sisi bawah di mana Anda tidak dapat melihatnya. DIO1 dan DIO2 masing-masing memiliki pin papan terpisah tetapi keduanya tidak terhubung ke port GPIO sehingga harus disambungkan secara eksplisit. Ini juga berarti bahwa TTGO V2 memiliki dua port GPIO lebih sedikit yang seharusnya dapat digunakan untuk tujuan lain

B. Alat dan Bahan

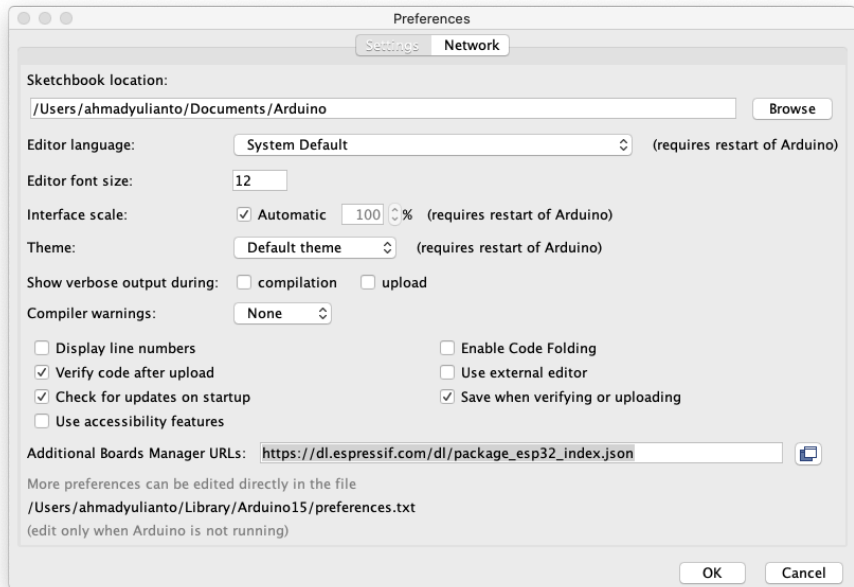
Alat dan bahan yang dibutuhkan pada Praktikum ini adalah:

1. Perangkat PC/Laptop
2. SX1276/1278

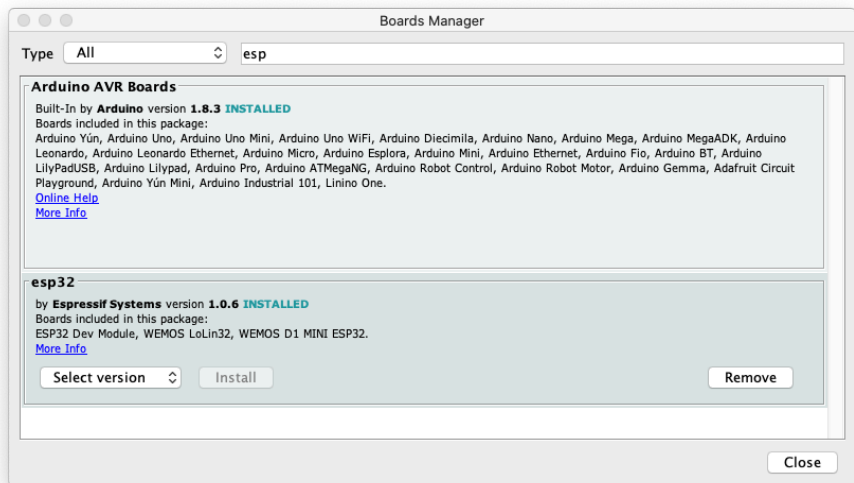
C. Simulasi dan Implementasi

Konfigurasi Arduino IDE:

- (1) Download dan install Arduino IDE (untuk keperluan penulisan ini digunakan versi 1.8.19)
- (2) Buka Arduino IDE, masukkan link berikut pada Preferences
https://dl.espressif.com/dl/package_esp32_index.json

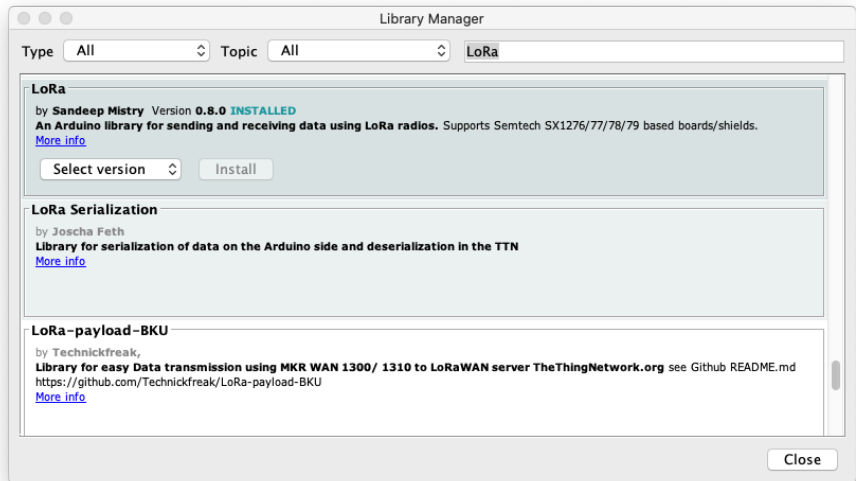


- Gambar 10.1 Preferences Arduino IDE
- (3) Pada menu Arduino IDE pilih Sketch kemudian Include Library , lalu Manage Libraries.



Gambar 10. Tampilan Boards Manager

- (4) Pada jendela Library Manager yang terbuka, masukkan kata “Lora” (tanpa tanda kutip)
- (5) Pilih library Lora versi terakhir, kemudian klik Install.



Gambar 10. Tampilan Library Manager

Konfigurasi LoRa

Bagian ini mendemokan pengiriman dan pembacaan pesan menggunakan LoRa dengan interval waktu tertentu.

- (6) Pada menu Arduino IDE , pilih **File** kemudian **New**
- (7) Salin kode berikut:

```
#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>

#define SCK      5    // GPIO5   -- SX1278's SCK
#define MISO     19   // GPIO19  -- SX1278's MISO
#define MOSI     27   // GPIO27  -- SX1278's MOSI
#define SS       18   // GPIO18  -- SX1278's CS
#define RST      14   // GPIO14  -- SX1278's RESET
#define DI0      26   // GPIO26  -- SX1278's
IRQ(Interrupt Request)
```

```

/*
 * Band frequencies:
 * 433 Hz - 433E6
 * 868 Hz - 868E6
 * 915 Hz - 915E6
 */

#define BAND    915E6

unsigned int counter = 0;

String rssi = "RSSI --";
String packSize = "--";
String packet ;
String msg = "";

void setup() {
  Serial.begin(115200);
  while (!Serial);
  Serial.println();
  Serial.println("Uji Pengiriman LoRa ");

  SPI.begin(SCK,MISO,MOSI,SS);
  LoRa.setPins(SS,RST,DI0);
  if (!LoRa.begin(BAND)) {
    Serial.println("Memulai LoRa: Gagal!");
    while (1);
  }
}

```



```

Serial.println("init ok");
delay(1500);
}

void loop() {

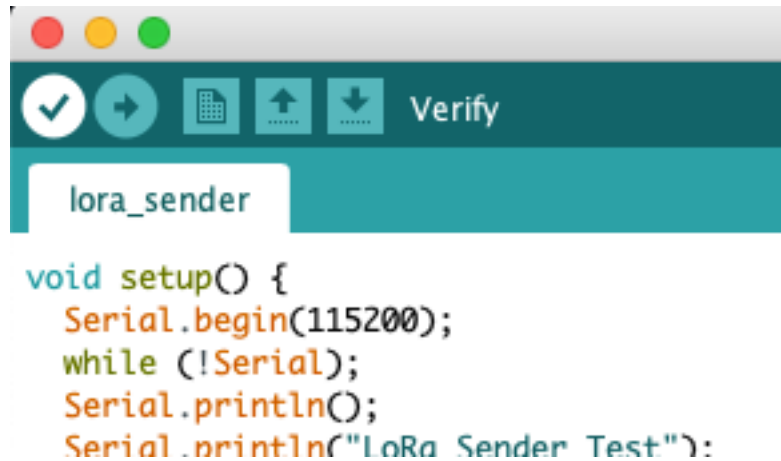
// if(Serial.available() > 0){
  msg = Serial.readString();
  // send packet
  msg = "("+String(counter)+"): "+ msg;
  Serial.print(msg);

  LoRa.beginPacket();
  LoRa.print(msg);
  LoRa.endPacket();
  counter++;
  delay(1000);

// }
//   Serial.print("Tidak ada serial");
}

```

- (8) Hubungkan Arduino IDE ke port USB di PC/Laptop hingga muncul nyala LED.
- (9) Pada Menu Arduino IDE , pilih Tools kemudian Port. Pilih Port aktif sesuai dengan hasil deteksi otomatis dari PC/Laptop.
- (10) Tekan tombol Verify , hal ini untuk memferifikasi kode program untuk mengecek kesalahan kode.



Gambar 10. Verifikasi Kode

- (11) Setelah dipastikan proses verifikasi berhasil, selanjutnya lakukan upload program
- (12) Selanjutnya pada Device TTGO Lilygo penerima, lakukan tahapan percobaan mulai dari Langkah 6 hingga 11 dengan kode program berikut ini

```

#include <SPI.h>
#include <LoRa.h>
#include <Wire.h>

#define SCK      5    // GPIO5  -- SX1278's SCK
#define MISO     19   // GPIO19 -- SX1278's MISO
#define MOSI     27   // GPIO27 -- SX1278's MOSI
#define SS       18   // GPIO18 -- SX1278's CS
#define RST      14   // GPIO14 -- SX1278's RESET
#define DI0      26   // GPIO26 -- SX1278's
                    IRQ(Interrupt Request)

/*
 * Band frequencies:
 * 433 Hz - 433E6
 * 868 Hz - 868E6

```

```

* 915 Hz - 915E6
*/

#define BAND      868E6

void loraData();
void cbk(int);

String rssi = "RSSI --";
String packSize = "--";
String packet ;

void loraData(){
    Serial.print(packet);
}

void cbk(int packetSize) {
    packet = "";
    packSize = String(packetSize,DEC);
    for (int i = 0; i < packetSize; i++) { packet +=
(char) LoRa.read(); }
    rssi = "RSSI " + String(LoRa.packetRssi(), DEC) ;
    loraData();
}

void setup() {
    Serial.begin(115200);
    while (!Serial);
    Serial.println();
}

```

```

Serial.println("LoRa Receiver Callback");
SPI.begin(SCK,MISO,MOSI,SS);
LoRa.setPins(SS,RST,DI0);
if (!LoRa.begin(868E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
}
//LoRa.onReceive(cbk);
LoRa.receive();
Serial.println("init ok");

delay(1500);
}

void loop() {
    int packetSize = LoRa.parsePacket();
    if (packetSize) { cbk(packetSize); }
    delay(10);
}

```

- (13) Dengan posisi LoRa penerima terhubung ke PC/Laptop selanjtkan buka Serial monitor untuk mengamati nilai RSSI.
- (14) Lakukan pengujian dengan kondisi LoS dan Non-LoS

D. Ringkasan

Jobsheet ini membahas mengenai penggunaan Teknologi Modulasi LoRA untuk pengiriman data jarak jauh dengan daya yang rendah.

E. Latihan:

Jawablah pertanyaan berikut ini.

1. Jelaskan cara kerja LoRa?
2. Sebutkan kelebihan dan kekurangan LoRa dibandingkan teknologi akses di IoT yang lain!

F. Tugas:

Kerjakan tugas berikut ini:

1. Lakukan pengukuran RSSI di sisi penerima dengan jarak kelipatan 10m!
2. Bandingkan pengukuran antara kondisi LoS dan Non-Los, berikan Analisa dan kesimpulannya!

DAFTAR PUSTAKA

Quinn, M.J. 2015. Ethics for the Information Age, 6th Edition. Seattle: Pearson.

B. Rajkumar, D.A.Vahid (2016). Internet of Things – Principles and Paradigms. Cambridge: Morgan Kaufmann

G.C. Hillar.(2017). MQTT Essentials – A Lightweight IoT Protocol. Packt Publishing

S. Ciraini, G. Ferrari, M. Picone, L. Veltri (2019). Internet of Things: Architectures, Protocols and Standards. Wiley

O. Hersent, D. Boswarthick, O. Elloumi (2012). The Internet of Things: Key Applications and Protocols. Wiley

Hanes et al., IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things

Sarosa, Mooechammad, Isnomo, YHP , Zakaria, M.N (2019). Internet of things: tuntunan praktis dan contoh aplikasi. Malang: Polinema Press.

https://lora-developers.semtech.com/uploads/documents/files/Building_a_LoRa-based_Device_End-to-End_with_Arduino_Dnld_Final.pdf

<https://www.lilygo.cn/>
<https://github.com/LilyGO/TTGO-LORA32-V2.0>

<https://www.makeuseof.com/>

Wikipedia. Raspberry Pi.
https://www.wikipedia.com/wiki/Raspberry_Pi

<https://www.thingsverse.com>

<http://www.randomnerdtutorials.com/>

GLOSARIUM

IoT : Internet Of Things yaitu system yang memungkinkan semua piranti terhubung langsung ke Internet

WSN : Wireless Sensor Network: Jaringan yang terdiri atas sensor-sensor yang berkomunikasi satu sama lain untuk proses pengiriman data hingga ke gateway

JSON: format pertukaran data dengan penulisan berbasis Javascript namun bisa digunakan untuk semua Bahasa pemrograman

Gateway: computer yang berfungsi sebagai penghubung antara jaringan local dengan jaringan Internet.

Server: computer yang berfungsi melayani permintaan dari klien.

INDEKS

akselerometer, 1
aplikasi, 2

CSV, 11

Dasbor, 9

gateway, 6
giroskop, 1
GPS, 1

Internet of Things (IoT), 1

JSON, 11

platform, 11

sensor cahaya, 1
sensor tekanan, 1
server, 8

uBeac, 2

Virtual Machine Platform, 13

Wireless Sensor Network (WSN), 1

BIODATA PENULIS

Ahmad Wilda Yulianto, ST., MT. Penulis menyelesaikan pendidikan sarjana dan pascasarjana di Institut Teknologi Sepuluh Nopember. Saat ini penulis bekerja sebagai dosen politeknik Negara Malang, Jurusan Teknik Elektro, Program Studi Jaringan Telekomunikasi Digital sejak tahun 2019. Penulis dapat dihubungi melalui email ahmadwildan@polinema.ac.id.

M. Nanak Zakaria, ST., MT. Penulis menyelesaikan pendidikan sarjana dan pascasarjana di Universitas Brawijaya Malang. Saat ini penulis bekerja sebagai dosen politeknik Negeri Malang, Jurusan Teknik Elektro, Teknik Telekomunikasi sejak tahun 1999. Penulis dapat dihubungi melalui email nanak.zakaria@polinema.ac.id.

Rieke Adriati Wijayanti, ST., MT. Penulis menyelesaikan pendidikan sarjana dan pascasarjana di Universitas Brawijaya, Fakultas Teknik, Departemen Teknik Elektro. Saat ini penulis bekerja sebagai dosen politeknik Negeri Malang, Jurusan Teknik Elektro, Program Studi Jaringan Telekomunikasi Digital sejak tahun 2016. Penulis dapat dihubungi melalui email riekeaw@polinema.ac.id.

Dr. Mochammad Junus , S.T., M.T. Penulis menyelesaikan pendidikan sarjana dan pascasarjana di Institut Teknologi Sepuluh Nopember Surabaya dan Universitas Brawijaya Malang. Saat ini penulis bekerja sebagai dosen Politeknik Negeri Malang, Jurusan Teknik Elektro, Teknik Telekomunikasi sejak tahun 1999. Penulis dapat dihubungi melalui email mochammad.junus@polinema.ac.id.