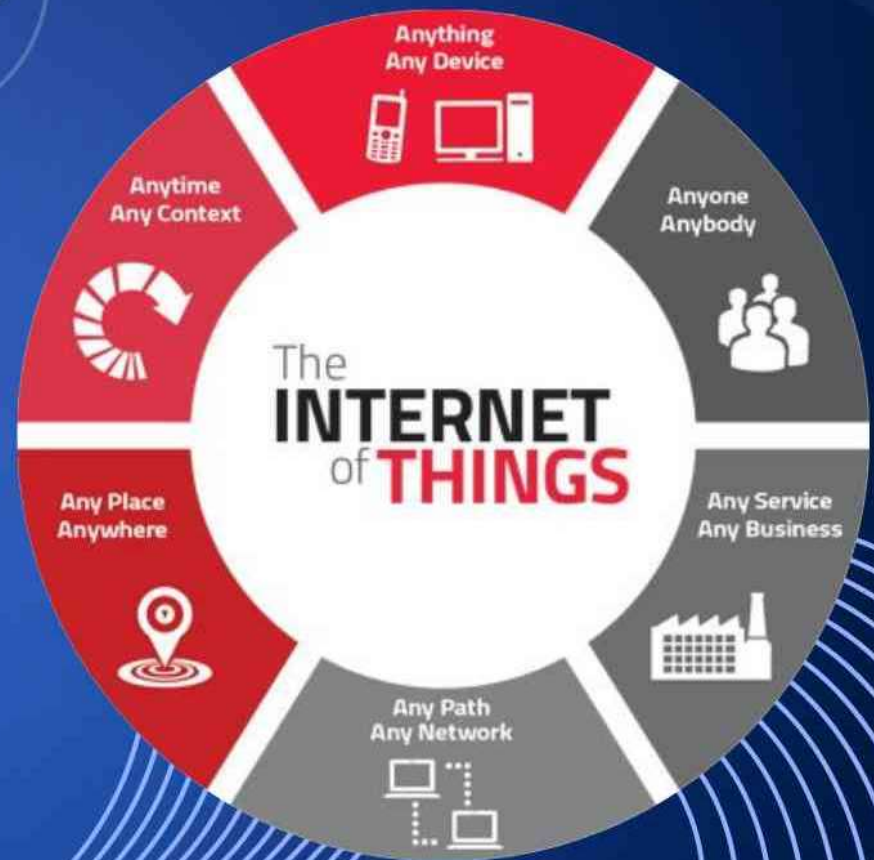


Industrial Automation & IoT Training

Presented by: ID-Networkers (IDN.ID)

APA ITU IOT

Pengertian IoT
Sejarah IoT
Macam-Macam IoT



Pengertian IoT

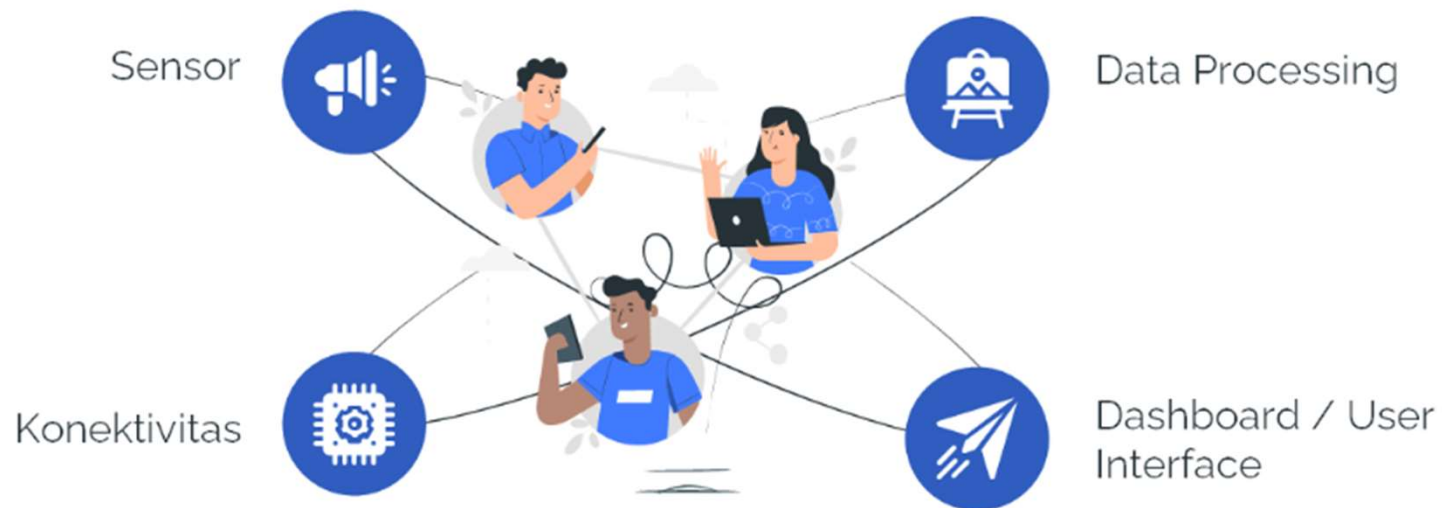
- Internet of things (IoT) diartikan dengan sistem yang menghubungkan semua perangkat komputasi dengan internet. Semua perangkat ini memiliki sensor, alat scan, maupun software sehingga memungkinkan perangkat-perangkat ini untuk terhubung dan bisa digunakan untuk melakukan transfer informasi.
- Cara sederhana untuk memahami internet of things (IoT) ini adalah usaha untuk menghubungkan semua alat komputasi fisik ke internet. Contoh yang paling mudah dipahami adalah smart TV dimana sebuah TV sekarang sudah bisa terhubung dengan internet dan bisa menampilkan apapun yang dibutuhkan.
- Menghubungkan perangkat fisik ke internet ini memiliki peran besar untuk kehidupan manusia, dimana peralatan penting seperti peralatan kesehatan, peralatan produksi bisa terkoneksi di internet sehingga bisa melakukan hal-hal yang sebelumnya tidak bisa dilakukan. Misalnya identifikasi, pertukaran data dan lain sebagainya.

Sejarah IoT



Unsur Unsur IoT

Unsur Internet Of Things



IoT Solution Architecture



Perkenalan

- Nama :
- Asal dari :
- Saat ini kerja di :
- Posisi sebagai :
- Pengalaman di dunia Automation :
- Pengalaman di IoT :
- Harapan :

Materi Training

- Gambar Electrical
- Modbus RS-485 to Device (Direct)
- Modbus RS-485 to HMI
- PLC - Ladder
- Modbus PLC
- IoT for Industrial
- Data Logging
- Else :

Preparation Before Training

www.idn.id

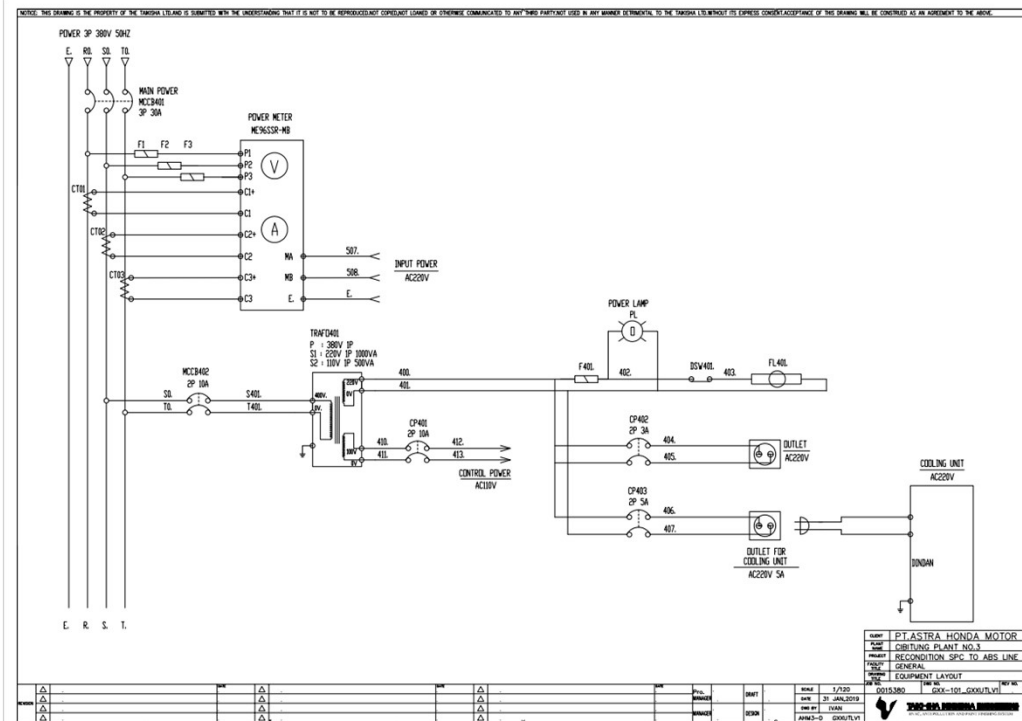
Gambar Electrical

NOTE: THIS DRAWING IS THE PROPERTY OF THE TANGKAS LILAND IS SHOWN WITH THE UNDERSTANDING THAT IT IS NOT TO BE REPRODUCED OR COPIED WITHOUT THE WRITTEN CONSENT OF TANGKAS LILAND. ANY UNAUTHORIZED REPRODUCTION OR COPIING OF THIS DRAWING WILL BE CONSIDERED AS AN INFRINGEMENT TO THE ABOVE.

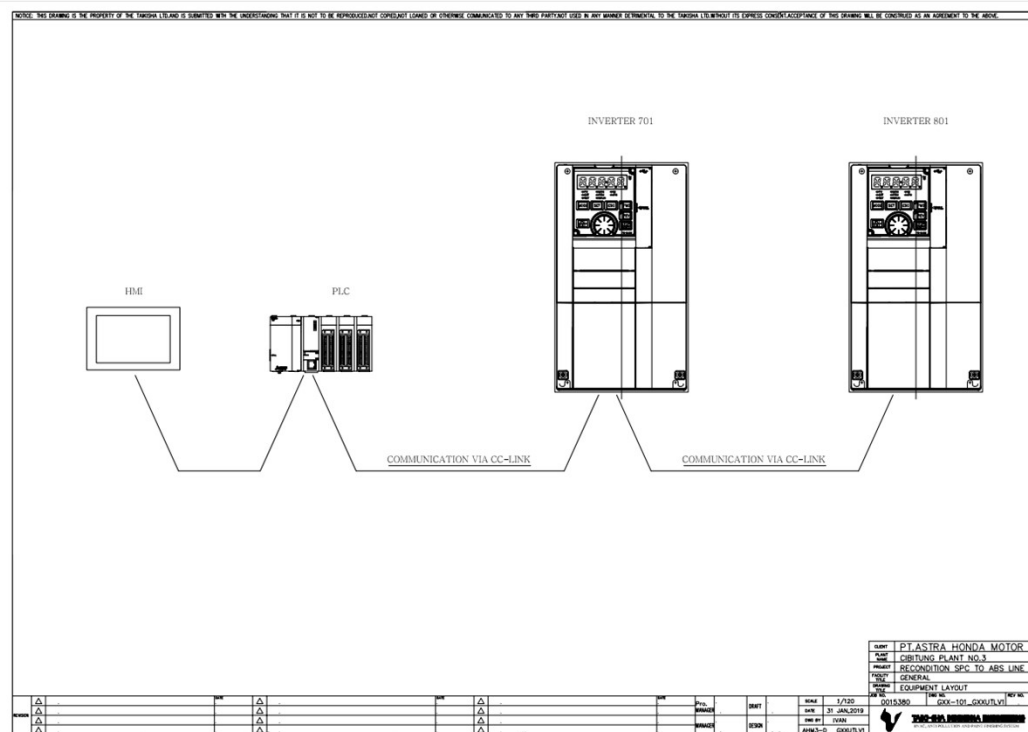
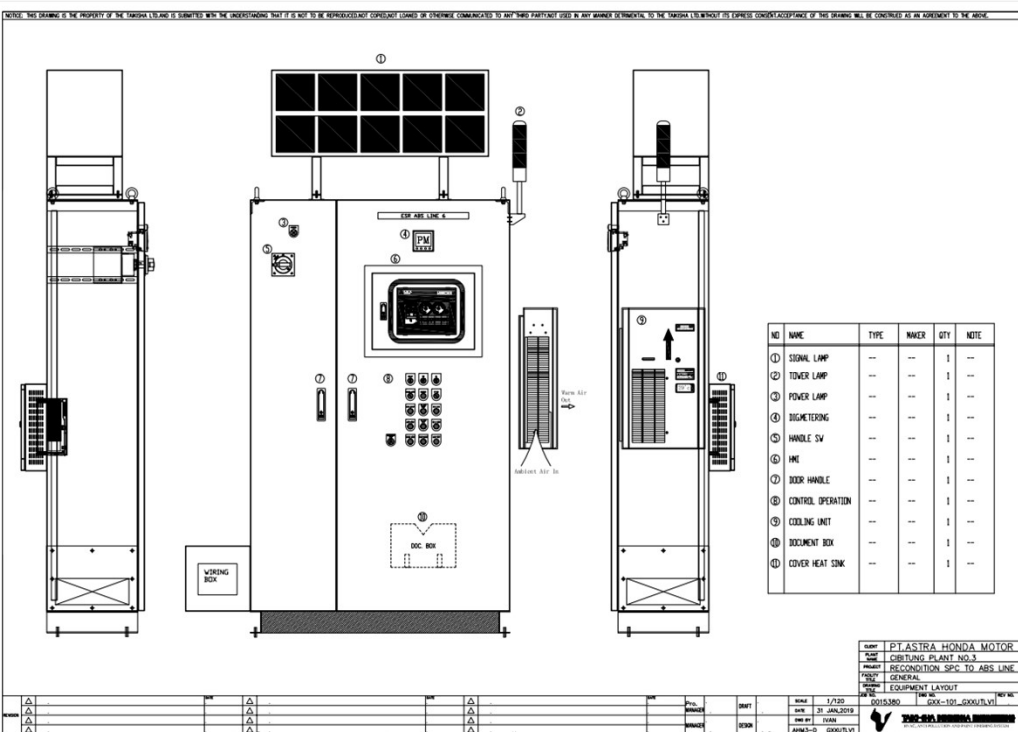
NAME OF PANEL : LP-OFFICE	
1. TYPE	FREESTANDING
2. INCOMING	SIDE ENTRY
3. OUTGOING	SIDE
4. PLATE THICKNESS	2.0 mm
5. PAINT	NOYON PLATONE 8000 NP-020 NP - 020 (DUST BLUE)
6. NOTE :	

NO	QTY	SYMBOL	DESCRIPTION	TYPE	MARK
1	1	MCB 01	MCB 3P 30A 7.5kA	PA20SAG	FILE
2	2	MCB 02, 03	MCB 3P 10A 7.5kA	PA20SAG	FILE
3	2	MCB 04, 05	MCB 3P 10A 7.5kA	PA20SAG	FILE
4	7	CP 01, 02, 03, 04, 05, 06, 07	CP 2P 20, 3A, 5A	CP20S	FILE
5	1		PILOT LIGHT AC220V	PI2000A-MC0	FILE
6	4	F1 - F4	FUSE CARRIER 2A	LS001	MARKER
7	1	MCM	METAL MOUNTING 56 X 56	ME2000-MB	FILE
8	1	POWER 3V	ROTARY SWITCH	TR-000	FILE
9	3	CT 01, 02, 03	CURRENT TRANSFORMER	CT1000	SCHEMATIC
10	1	CT01	CT 100V / 5A	CT01	FILE
11	1	TR-01	TRANSFORMER P.V. AC220V 1 Ph, 5kVA AC220V 1 Ph, 1000VA	TR-01	FILE
12	1	TR-02	TRANSFORMER P.V. AC220V 1 Ph, 5kVA AC220V 1 Ph, 1000VA	TR-02	FILE
13	2	INV 700, 800	FR-400 3P 300V 2.2 kW	FR-400-2-60	MITOBI
14	2	CC-LINK 700, 800	CC-LINK CABLE FOR FR-400 SERIES	FR-400C	MITOBI
15	1	PLC SET	MELSEC Q SERIES	Q-001	MITOBI
1	1		BASE UNIT	Q-001	MITOBI
1	1		POWER SUPPLY UNIT	Q-001	MITOBI
1	1		CPU UNIT	Q-001	MITOBI
1	1		CONTROL & COMMUNICATION LINK UNIT	Q-001	MITOBI
1	1		DIGITAL INPUT MODULE	Q-001	MITOBI
1	1		DIGITAL OUTPUT MODULE	Q-001	MITOBI
4	4	D1, D2, D3, D4	TERMINAL BLOCK UNIT	ASTB-2Y36	MITOBI
4	4	CAB	CABLE TO TERMINAL BLOCK UNIT	AC-010	MITOBI
1	1	HR	HR TERMINAL 1P	CC-010 M100	MITOBI
1	1	PS	POWER SUPPLY AC220V 240V, 100 V	SRV1000A	OMRON
1	1	NR	NOISE FILTER SA AC220V	NRV1000	OMRON
2	2	MC001, MC002	MAGNETIC CONTACTOR 3P 5A	SC-5-1	FILE ELECTRIC
3	3	CR001, CR002, CR003	RELAY 1P 5A AC220V	LY-2N AC220V	OMRON
16	16	CR001 - CR016	RELAY 1P 5A AC220V	LY-2N AC220V	OMRON
1	1	TL-01	TIMER LAMP RED, YELLOW, GREEN	CP25-2-08	SHIMADZU
1	1	KS-01	KEYLOCK SWITCH	AKS0-10A-10A	FILE ELECTRIC
3	3	PL 500A, PL 500V	PILOT LIGHT AC220V & DC24V	PI2000A-MB, PI2000A-LS	FILE ELECTRIC
7	7	ES001, ES002, ES003, ES004, ES005, ES006, ES007, ES008, ES009, ES010, ES011, ES012	EMERGENCY STOP PB WITH LAMP	AKS0-10A-10A	FILE ELECTRIC
7	7	ES001, ES002, ES003, ES004, ES005, ES006, ES007, ES008, ES009, ES010, ES011, ES012	EMERGENCY STOP PB WITH LAMP	AKS0-10A-10A	FILE ELECTRIC
2	2	ES001, ES002	EMERGENCY STOP PB WITH LAMP	AKS0-10A-10A	FILE ELECTRIC
2	2	ES001, ES002	EMERGENCY STOP PB WITH LAMP	AKS0-10A-10A	FILE ELECTRIC
1	1	PS	PUSH BUTTON GREEN	AKS0-10A-10A	FILE ELECTRIC
14	14	PS	PUSH BUTTON RED	AKS0-10A-10A	FILE ELECTRIC
16	16	PS	ILLUMINATED PUSH BUTTON GREEN	AKS0-10A-10A	FILE ELECTRIC
2	2	SS	SELECTOR SWITCH	AKS0-10A-10A	FILE ELECTRIC
2	2	SS	ILLUMINATED SELECTOR SWITCH GREEN	AKS0-10A-10A	FILE ELECTRIC
2	2	PS	PUSH BUTTON SELECTOR SWITCH	AKS0-10A-10A	FILE ELECTRIC
1	1	BZ	BELL	AKS0-10A-10A	FILE ELECTRIC
2	2	TRC	RELAY TRIGGER	AKS0-10A-10A	FILE ELECTRIC
2	2	BR	RELAY BARRIER	AKS0-10A-10A	FILE ELECTRIC

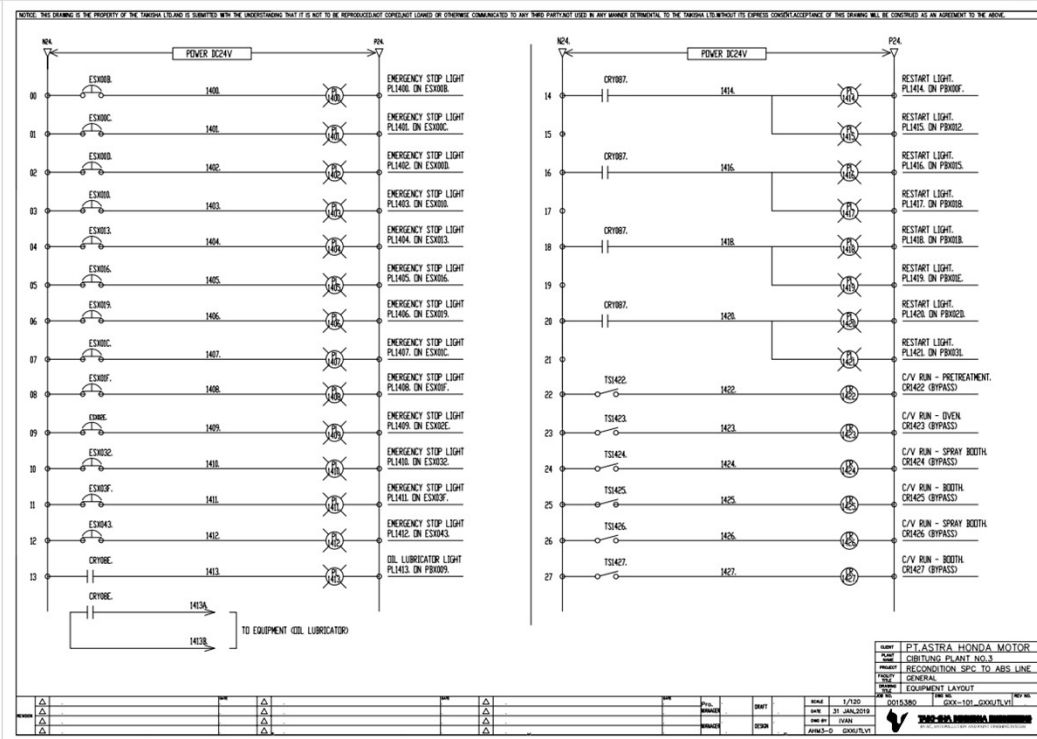
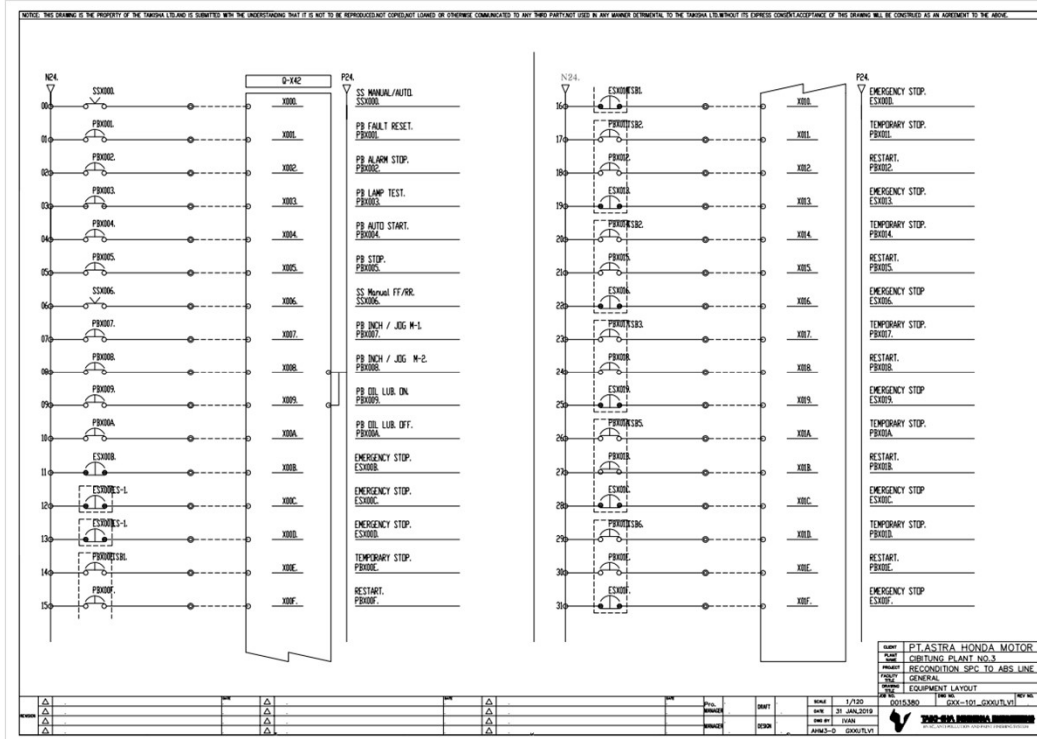
NAME	DATE	BY	CHK	APP	REV	DATE	DESCRIPTION
PT ASTRA HONDA MOTOR	1/2020	0015380	0015380	0015380	0015380	0015380	0015380
CHITUNG PLANT NO.3	1/2020	0015380	0015380	0015380	0015380	0015380	0015380
RECONSTRUCTION SPEC. TO ABS. LINE	1/2020	0015380	0015380	0015380	0015380	0015380	0015380
GENERAL	1/2020	0015380	0015380	0015380	0015380	0015380	0015380
EQUIPMENT LAYOUT	1/2020	0015380	0015380	0015380	0015380	0015380	0015380



Gambar Electrical



Gambar Electrical



Modbus

Modbus adalah protokol komunikasi serial yang dipublikasikan oleh Modicon pada tahun 1979 untuk diaplikasikan ke dalam programmable logic controllers (PLCs). Modbus sudah menjadi standar protokol yang umum digunakan untuk menghubungkan peralatan elektronik industri.

Adapun Function Code yang ada di dalam Modbus adalah seperti berikut.

Modbus Function Codes Recognized by CSI Gateways

Function Code	Register Type
1	Read Coil
2	Read Discrete Input
3	Read Holding Registers
4	Read Input Registers
5	Write Single Coil
6	Write Single Holding Register
15	Write Multiple Coils
16	Write Multiple Holding Registers

Macam Macam Modbus

Modbus RTU

Modbus RTU (Remote Terminal Unit) merupakan varian Modbus yang umum digunakan pada komunikasi serial dengan bentuk yang ringkas. Format RTU menggunakan mekanisme Cyclic Redundancy Check (CRC) untuk memastikan data yang dikirim lengkap dan benar. Dalam setiap 8-byte data yang dikirim berisi karakter 4-bit heksa desimal.

Modbus TCP

Modbus TCP (dikenal juga sebagai Modbus-TCP/IP) adalah protokol Modbus RTU menggunakan antarmuka TCP yang berjalan pada media Ethernet. Modbus TCP adalah protokol yang berjalan pada layer aplikasi TCP/IP.

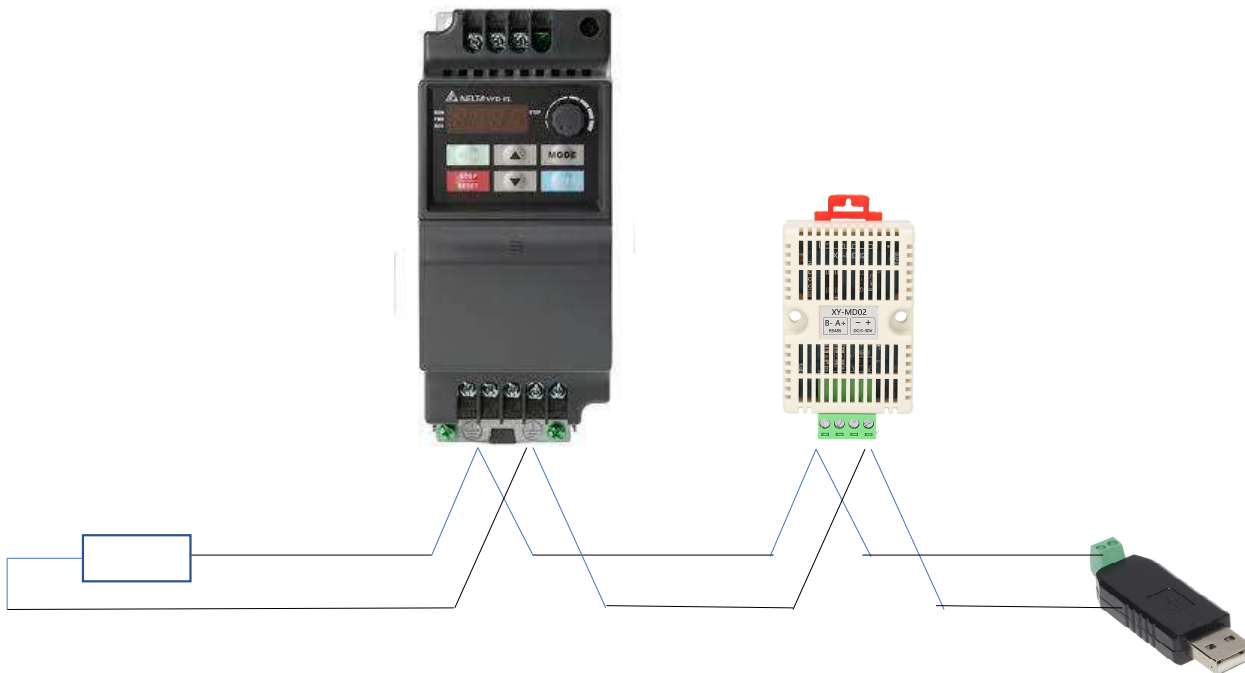
Modbus ASCII

Digunakan pada komunikasi serial dengan memanfaatkan karakter ASCII. Format ASCII menggunakan mekanisme *longitudinal redundancy check* (LRC). Setiap frame data Modbus ASCII diawali dengan titik dua (":") dan baris baru yang mengikuti (CR/LF).

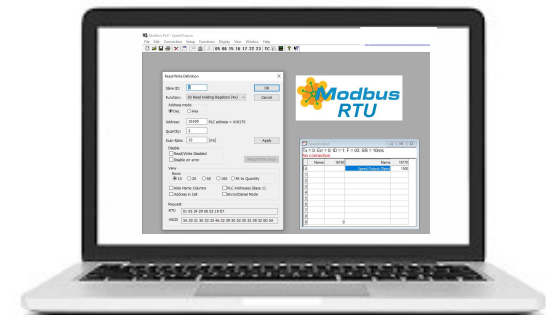
Istilah Error Message pada ModBuss Poll

1	Illegal Function	The function code received in the query is not recognized by the slave or is not allowed by the slave.
2	Illegal Data Address	The data address (register number) received in the query is not an allowed address for the slave, i.e., the register does not exist. If multiple registers were requested, at least one was not permitted.
3	Illegal Data Value	The value contained in the query's data field is not acceptable to the slave.
4	Slave Device Failure	An unrecoverable error occurred while the slave was attempting to perform the requested action
6	Slave Device Busy	The slave is engaged in processing a long-duration command. The master should try again later.
10	Gateway Path Unavailable	Specialized use in conjunction with gateways, usually means the gateway is misconfigured or overloaded
11	Gateway Target Device Failed to Respond	Specialized use in conjunction with gateways, indicates no response was received from the target device.

Konfigurasi / Wiring ModBus RS-485



1. ModBus Poll
2. Modscan



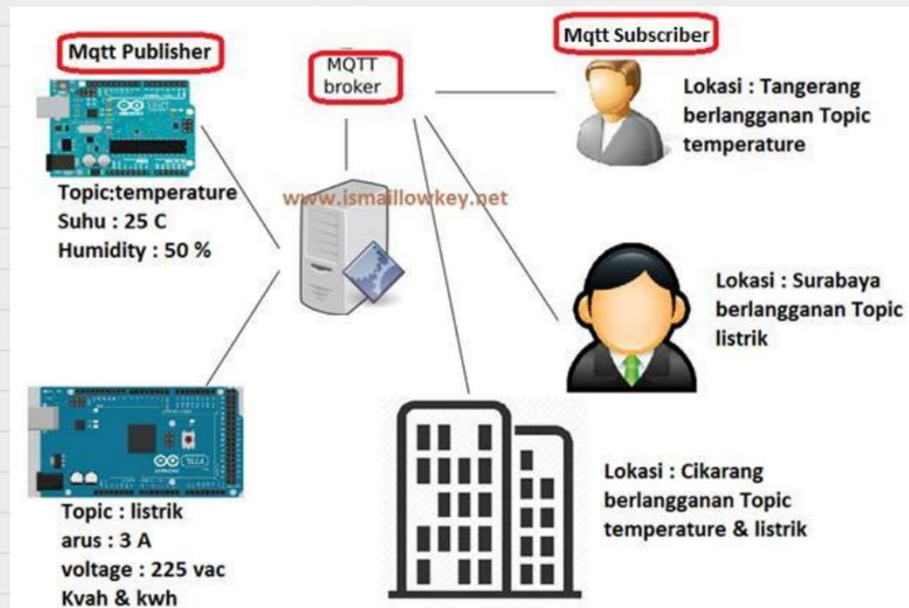
Perkenalan MQTT

“**MQTT** merupakan sebuah protokol yang diterapkan pada IOT. Protokol ini sangat mendukung untuk jaringan WAN, karena WAN mencakup area yang luas. Protokol **MQTT** mempunyai kelebihan yaitu dapat bekerja dengan energi dan media penyimpanan yang minimum.”

(<http://e-journals.unmul.ac.id>)

IoT Haiwell

Contoh Skema MQTT



IoT Haiwell

Istilah-istilah di MQTT

Publisher

Pihak yang **mengirim** topic disertai dengan message/pesan. Publisher dapat mengirim topic disertai message/pesan secara realtime(terus menerus) maupun secara event(hanya sesekali). Publisher dapat mempublish berbagai topic sekaligus secara realtime

Subscriber

Subscriber adalah pihak yang **berlangganan** topic. Subscriber dapat berlangganan satu topic maupun banyak topic sekaligus dari berbagai publisher. Subscriber tidak akan tahu darimana sumber topic tersebut dipublish, Subscriber hanya tahu message/pesan dari topic yang ada di broker

Broker

Broker adalah pihak penyedia yang berfungsi layaknya penerbit pada koran. Broker berfungsi menjembatani antara **publisher** dan **subscriber**.

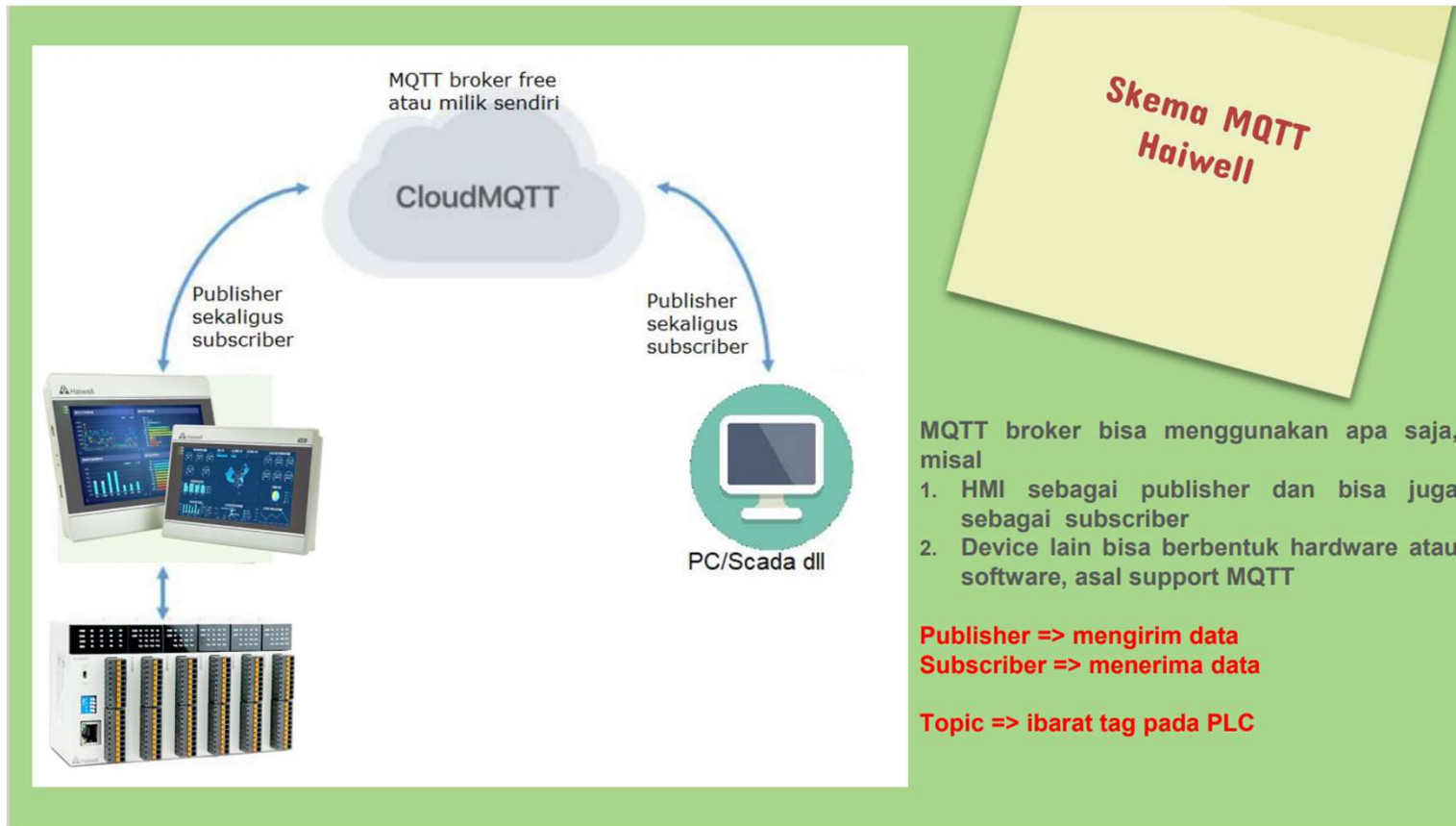
Topic

Topic semacam variable yang akan dikirim oleh **publisher** dan dibaca oleh **subscriber**, topic sendiri dapat dianalogikan topic pada majalah/berita seperti topic tentang makanan, game,topic ekonomi dan sebagainya. Setiap topic akan terdapat **message/pesan** . contoh topic **officejakarta/Room1/suhu** berisi message/pesan suhu di office jakarta pada room1

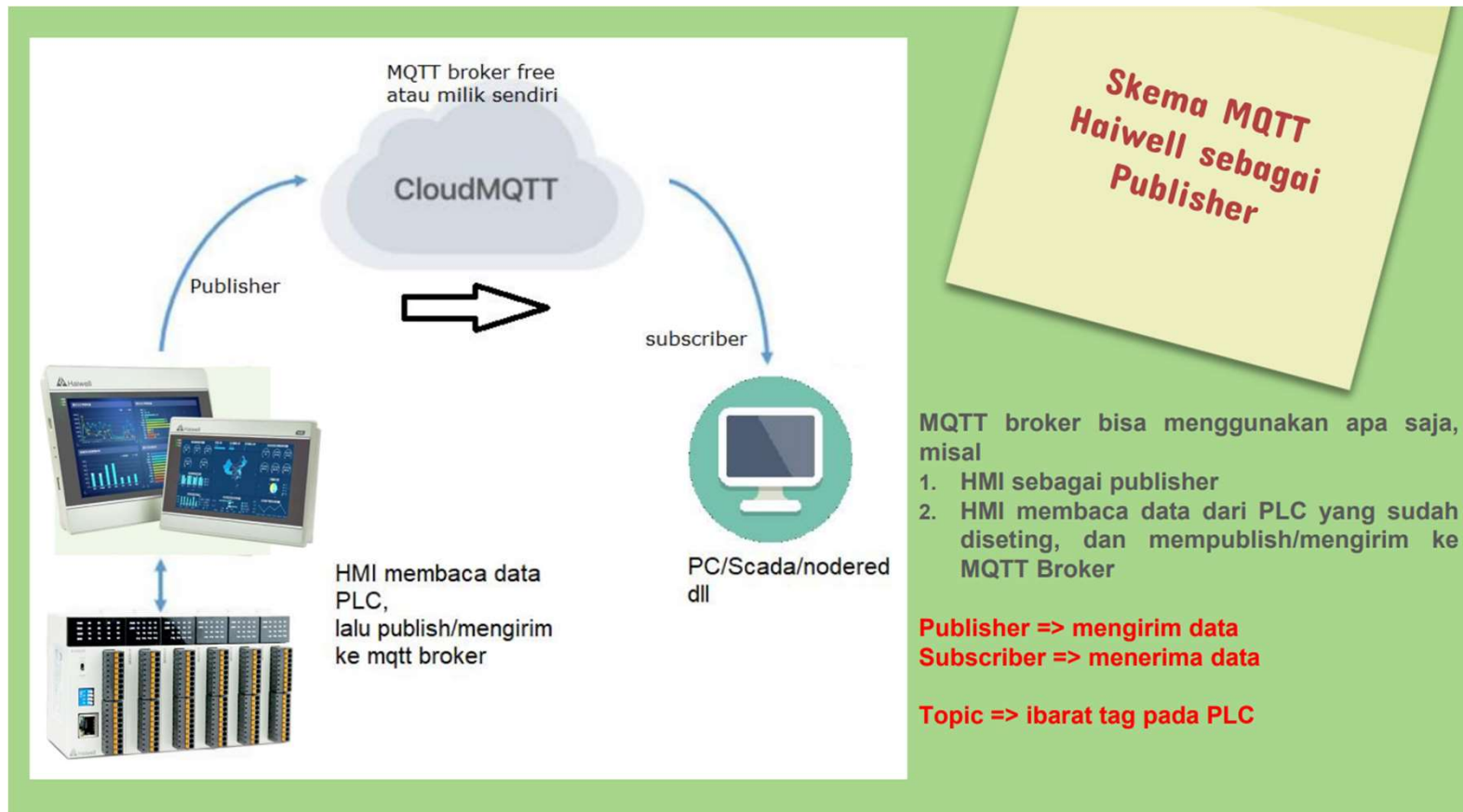
Message

Message adalah isi dari topic yang dikirim oleh publisher, message/pesan dapat berisi data secara langsung,misal suhu ruangan maupun dapat berupa format **JSON (JavaScript Object Notation)**. Didalam format JSON dapat berisi berbagai macam data,sehingga satu topic dapat memuat banyak berbagai data

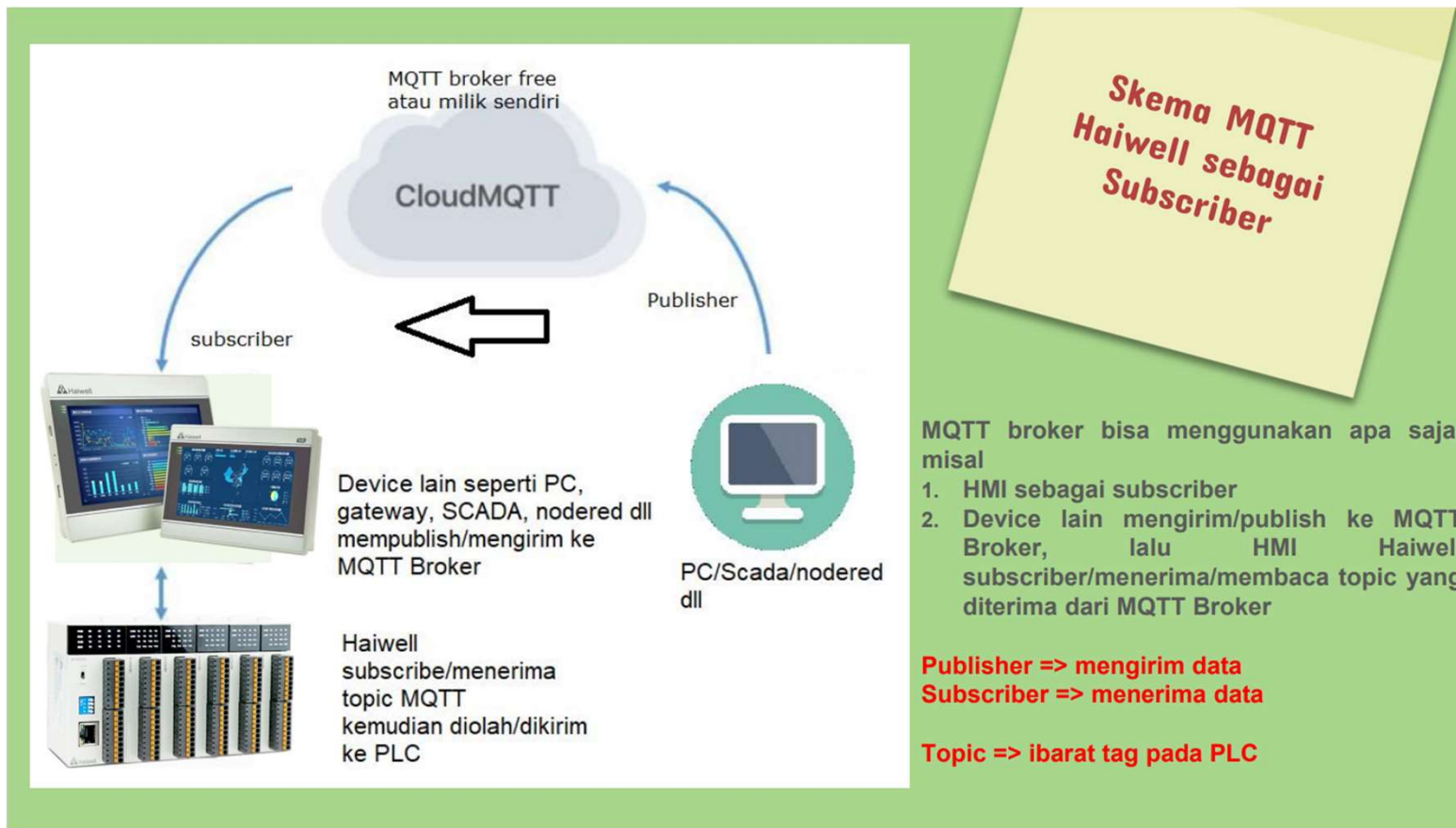
IoT Haiwell



IoT Haiwell



IoT Haiwell



IoT Haiwell

Contoh JSON Haiwell sebagai Publisher

```
{
  "_terminalTime": "2021-11-13 22:25:18.733",
  "_groupName": "group1",
  "data1": "111",
  "data2": "0",
  "data3": "0"
}
```

Pada sebelah kiri adalah **key**, terdapat **_terminalTime** sampai **data3**, Sedangkan sebelah kanan adalah **value** pada **key** tersebut.

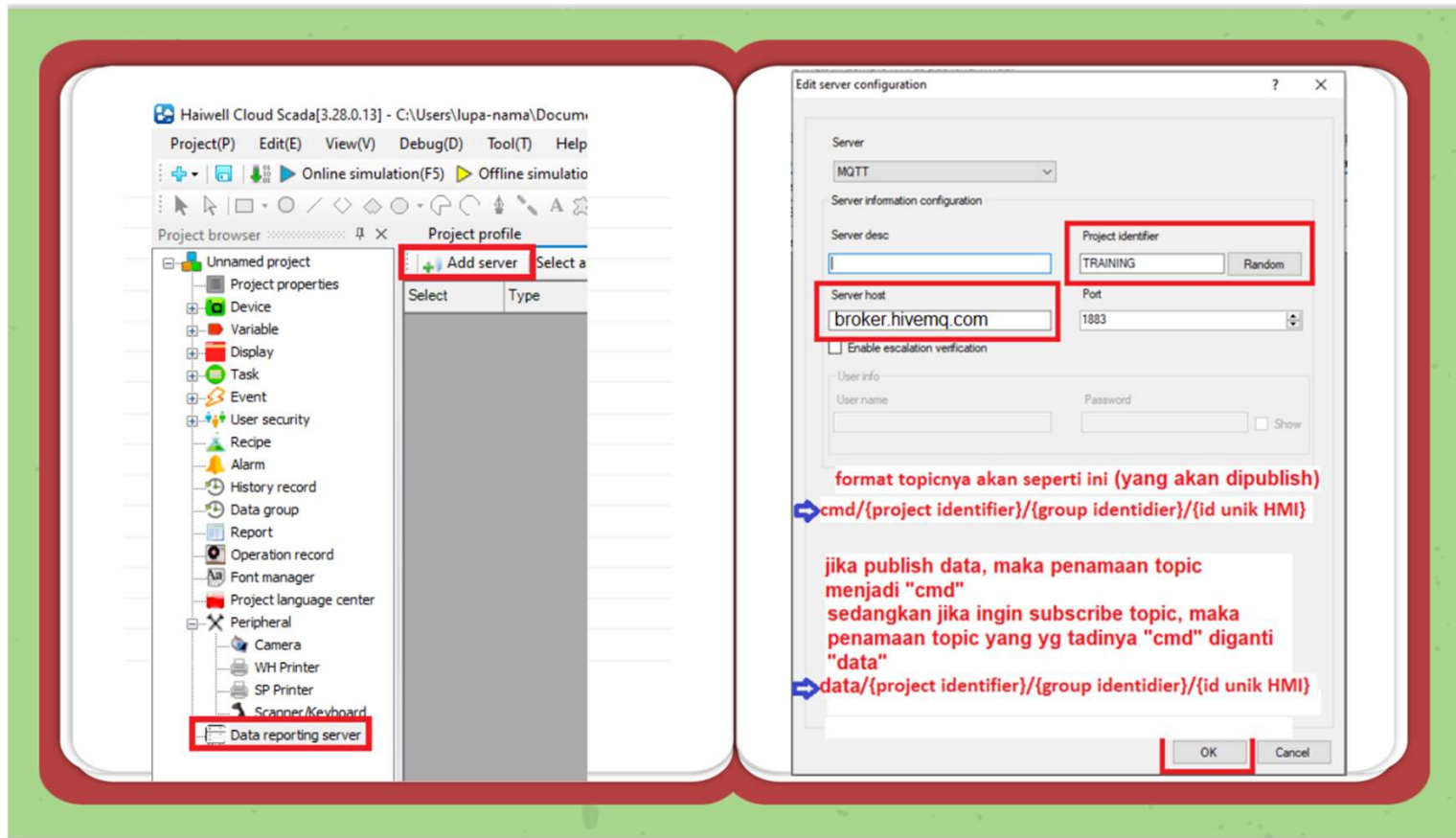
_terminalTime dan **_groupName** merupakan key JSON bawaan haiwell, sedangkan **data1, data2, data3** merupakan key yg bisa ditambah/diganti nama sesuai kebutuhan

IoT Haiwell

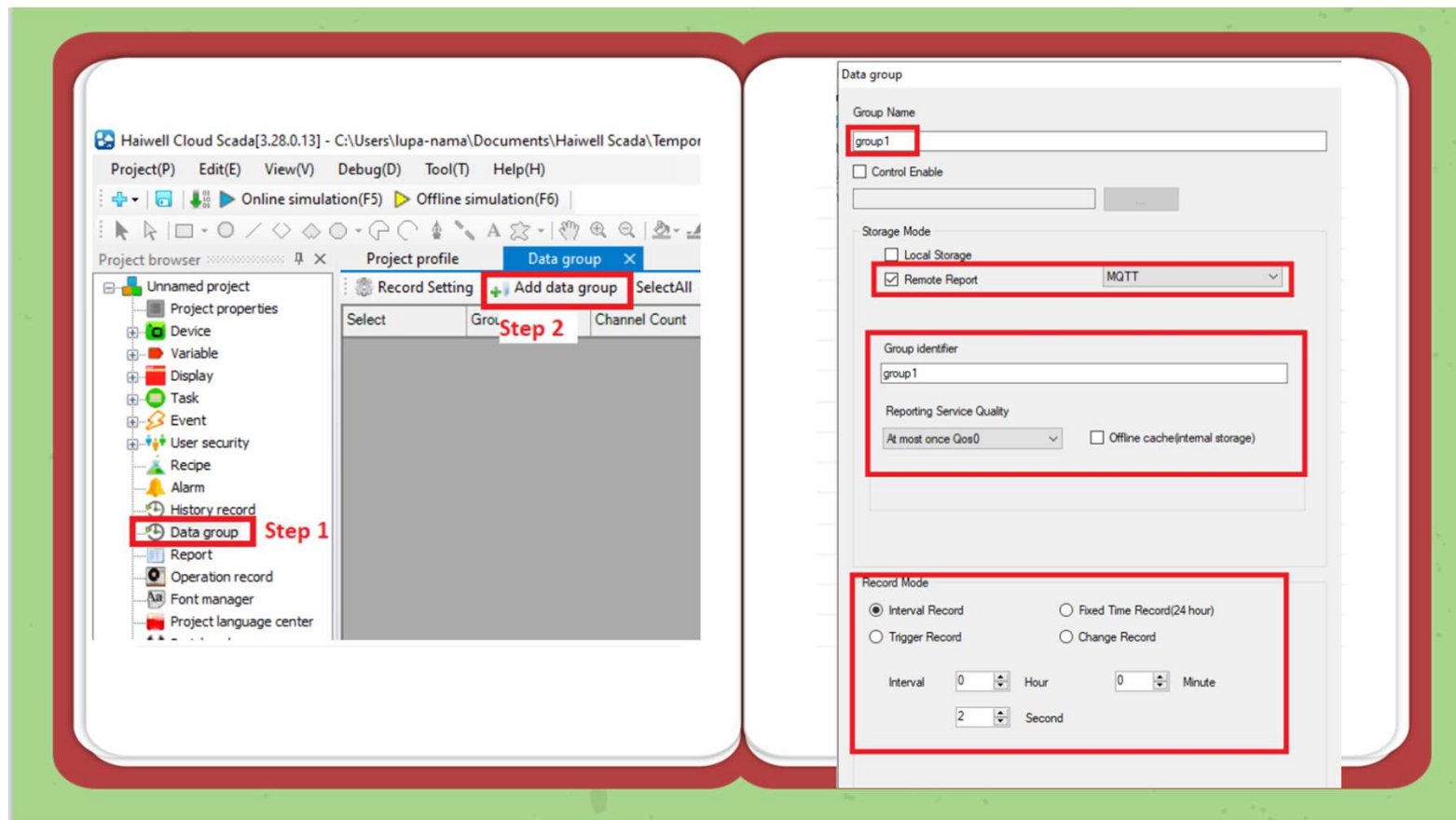
[Tutorial] HMI Haiwell sebagai MQTT Publisher

Dijelaskan cara publish data dari
HMI Haiwell ke MQTT
Termasuk nama topic dan format
JSON

IoT Haiwell



IoT Haiwell



IoT Haiwell

pilih 3 jika ingin kirim 3 data dalam 1 topic MQTT (format JSON)

Channel Count(1-100)

Record Channel Variables List

Index	Channel Variable	Channel identifier
1		Flag1
2		Flag2
3		Flag3

Isi Data JSON
HMI Haiwell

Channel Count(1-100) Start Index

Record Channel Variables List

Index	Channel Variable	Channel identifier	Channel Title	Enable remote wr	Integer	Digits
1	InternalVariable_1.data1	data1	data1	☒	5	0
2	InternalVariable_1.data2	data2	data2	☒	5	0
3	InternalVariable_1.data3	data3	data3	☒	5	0

centang

topic

payload atau isi topic

terminal time = timestamp saat data dipublish

```
{
  "_terminalTime": "2021-11-13 22:25:18.733",
  "_groupName": "group1",
  "data1": "111",
  "data2": "0",
  "data3": "0"
}
```

Channel Variable

Merupakan nama tag hmi yang akan dipublish ke MQTT

Semua value key dari JSON berbentuk string

IoT Haiwell

```
JSON
{
  "_groupName": "group1",
  "data1": "0",
  "data2": "0",
  "data3": "0"
}

2022-03-17 11:05:28:963
jika simulasi software biasanya berbentuk angka random, jika
download ke HMI, biasanya menjadi 123
Topic: data/TRAINING/group1/55285886242332169
95 QoS: 0

{
  "_terminalTime": "2022-03-17 11:05:2
9.924",
  "_groupName": "group1",
  "data1": "0",
  "data2": "0",
  "data3": "0"
}

2022-03-17 11:05:29:958
```

**Test
Subscribe
data dengan
software
MQTTx**

MQTTx Software

Merupakan software untuk mengetest mqtt antara lain melihat message saat subscribe, maupun test publish topic

IoT Haiwell

Contoh

Server: MQTT

Server information configuration

Server desc: broker.hivemq.com

Project identifier: TRAINING

Port: 1883

Enable escalation verification: ☐

User info

User name:

Password:

Show: ☐

Data group

Group Name: data

Control Enable: ☐

Storage Mode

☒ Local Storage

☒ Remote Report

MQTT

Group identifier

sendTemp

Reporting Service Quality

At least once Qos1

☒ Offline cache(internal storage)

publish topic : **data/{Project identifier}/{Group identifier}/{id unik}**

data/TRAINING/sendTemp/+

+ => artinya baca semua id unik HMI

note:

- huruf besar dan kecil dibedakan
- tidak boleh ada spasi

IoT Haiwell

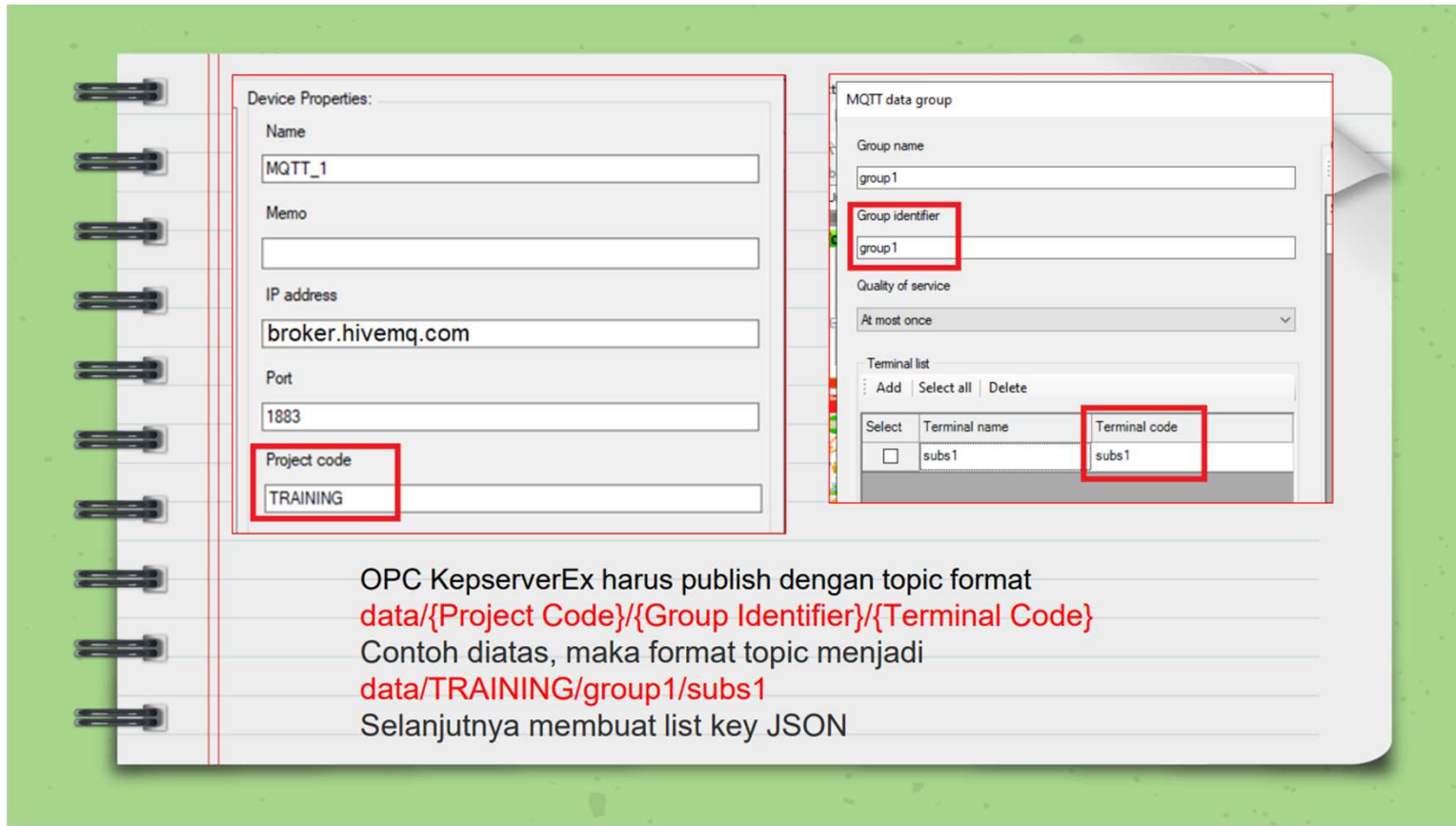
Setting HMI Haiwell sebagai subscriber

Untuk Dapat publish topic dari OPC KepserverEx ke HMI Haiwell, harus mengikuti format topic HMI Haiwell, Adapun format topic HMI Haiwell seperti berikut

data/{Project Code}/{Group Identifier}/{Terminal Code}

*** Sangat disarankan Project code untuk publisher dan subscriber dibuat berbeda**

IoT Haiwell



The screenshot displays the 'Device Properties' and 'MQTT data group' configuration windows. In the 'Device Properties' window, the 'Name' field is 'MQTT_1', 'IP address' is 'broker.hivemq.com', 'Port' is '1883', and 'Project code' is 'TRAINING'. In the 'MQTT data group' window, the 'Group name' is 'group1', 'Group identifier' is 'group1', 'Quality of service' is 'At most once', and the 'Terminal list' contains one entry: 'subs1' with 'Terminal code' 'subs1'. Red boxes highlight the 'Project code', 'Group identifier', and 'Terminal code' fields.

Device Properties:

Name: MQTT_1

Memo:

IP address: broker.hivemq.com

Port: 1883

Project code: TRAINING

MQTT data group

Group name: group1

Group identifier: group1

Quality of service: At most once

Terminal list

Select	Terminal name	Terminal code
☐	subs1	subs1

OPC KepserverEx harus publish dengan topic format
data/{Project Code}/{Group Identifier}/{Terminal Code}
Contoh diatas, maka format topic menjadi
data/TRAINING/group1/subs1
Selanjutnya membuat list key JSON

IoT Haiwell

Channel list

... Add Select all Delete

Select	Channel name	Channel identifier	Data type	Data length	Read-write mode
☐	tagopc1	tagopc1	Integer	1	Read and write
☒	tagopc2	tagopc2	Integer	1	Read and write

Di haiwell, dibuat subscribe dengan key **tagopc1** dan **tagopc2**, maka di OPC KepserverEx harus publish dengan topic **data/TRAINING/group1/subs1** dan format JSON

```
{  
  "tagopc1" : "111",  
  "tagopc2" : "222"  
}
```

- * Value harus string (jika integer, maka ubah ke string dahulu)
- * Jika value Boolean, nilainya "1" atau "0", tidak boleh true atau false

IoT Haiwell

Test publish menggunakan MQTTx

Payload: JSON QoS: 0 ☐ Retain
data/TRAINING/group1/subs1
{
 "tagopc1" : "123",
 "tagopc2" : "456"
}

untuk send


HMI as Subscriber

123

456

Di sebelah kiri, dengan software MQTTx, testh publish data, sedangkan sebelah kanan adalah layar HMI

* Jika HMI haiwell diset sebagai subscriber, maka message mqtt akan masuk ke tag/variable haiwell mqtt, untuk mengirim dari tag/variable mqtt haiwell ke PLC, butuh script di program HMI haiwell

| IoT Haiwell