



PROYEK AKHIR - VE 180626

**RANCANG BANGUN *HUMAN MACHINE INTERFACE*
(HMI) PADA SISTEM *TELEMONITORING* JARINGAN
PENERANGAN JALAN UMUM (PJU)**

Devi Dwi Ariyani
NRP 10311500000050

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng
Ciptian Weried P., S.ST, MT.

Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - VE 180626

***DESIGN OF HUMAN MACHINE INTERFACE (HMI) ON
SOLAR STREET LIGHTING NETWORK TELEMONITOR-
ING SYSTEM***

Devi Dwi Ariyani
NRP 10311500000050

Supervisor
Ir. Josaphat Pramudijanto, M.Eng
Ciptian Wered P., S.ST, MT.

*Electrical and Automation Engineering Department
Vocational Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2018*


PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul "**Rancang Bangun Human Machine Interface pada Sistem Telemonitoring Jaringan Penerangan Jalan Umum (PJU)**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 17 Januari 2019



Devi Dwi Ariyani
10311500000050

-----Halaman ini sengaja dikosongkan-----



**RANCANG BANGUN *HUMAN MACHINE INTERFACE*
(HMI) PADA SISTEM *TELEMONITORING* JARINGAN
PENERANGAN JALAN UMUM (PJU)**



PROYEK AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember



Menyetujui :

Dosen Pembimbing 1

Dosen Pembimbing 2

Ir. Josaphat Pramudijanto, M.Eng.
NIP. 1962.10.05.1990.03.1/003

Ciptian Weried P., S.ST. MT.
NIP. 1990.20171.1.060



Handwritten signature and date: 21/12/2019

**SURABAYA
JANUARI, 2019**

-----Halaman ini sengaja dikosongkan-----

**RANCANG BANGUN *HUMAN MACHINE INTERFACE*
(HMI) PADA SISTEM *TELEMONITORING* JARINGAN
PENERANGAN JALAN UMUM (PJU)**

Nama : Devi Dwi Ariyani
Pembimbing : Ir. Josaphat Pramudijanto, M.Eng
Ciptian Weried P., S.ST. MT

ABSTRAK

Penerangan Jalan Umum merupakan salah satu sarana penting dalam setiap pembangunan infrastruktur. Maraknya penggunaan *Photovoltaic* (PV) sebagai sumber daya alternatif PJU, belum diimbangi dengan metode perawatan peralatan yang menunjang. Perawatan PJU masih dilakukan secara konvensional, sehingga petugas harus memeriksa kondisi tiap PJU secara bergantian dan memakan waktu yang lama. Hal tersebut dinilai kurang efektif dan efisien.

Berdasarkan permasalahan tersebut, dibuat sebuah *Human Machine Interface* (HMI) menggunakan *software* Microsoft Visual Studio 2017, modul Ethernet Shield untuk pengiriman datanya dan Arduino Mega 2560 sebagai mikrokontrolernya. Data yang dikirim masing-masing PJU adalah data dari komponen *Photovoltaic* (PV), Baterai, dan LED. Untuk mengetahui apabila komponen rusak, maka dibuatlah indikator status normal/*error* dalam tampilan HMI. Status normal/*error* ditentukan dari nilai data yang diterima oleh Microsoft Visual Studio 2017 yang telah diprogram agar dapat membedakan nilai yang diterima termasuk kedalam *range* komponen aman atau rusak.

Hasil yang diperoleh dari pembuatan dan pengujian HMI ini adalah HMI dapat menampilkan status normal/*error* berdasarkan nilai yang diterima dengan persentase keberhasilan program sebesar 100%, penerimaan data sebesar 99,95% dengan jarak 40 meter secara *wireless*, dan keberhasilan seluruh sistem sebesar 88%.

Kata Kunci :

Ethernet Shield, *Human Machine Interface*, Microsoft Visual Studio, PJU

-----Halaman ini sengaja dikosongkan-----

**DESIGN OF HUMAN MACHINE INTERFACE (HMI) ON
SOLAR STREET LIGHTING NETWORK TELEMONITOR-
ING SYSTEM**

Name : Devi Dwi Ariyani
Supervisor : Ir. Josaphat Pramudijanto, M.Eng
Ciptian Weried P., S.ST. MT

ABSTRACT

Public street lighting is one of the important means in any infrastructure development. Rampant use of Photovoltaics (PV) as alternative resources PJU, yet balanced with method of care equipment support. PJU care still carried out konvensional, so the officer should check the conditions each PJU interchangeably and take a long time. It is judged less effective and efficient.

Based on these problems, created a Human Machine Interface (HMI) software using Microsoft Studio Visual Studio 2017, module Ethernet Shield for the delivery of Dathan-ya and Arduino Mega 2560 as mikrokontrolernya. The data sent in each of the component data is PJU Photovoltaics (PV), batteries and LEDs. To find out if components are damaged, then a single normal status indicator/error in display HMI. Status normal/error is determined from the value of the data received by a Microsoft Visual Studio 2017 which has been programmed to be able to distinguish the value received is included into the range of safe components or damaged.

The results obtained from the manufacture and testing of HMI HMI can display this is the normal status/error based-kan accepted values with percentage of success amounting to 100% of the data and the acceptance of 99.95% at a distance of 40 metres in Wireless and percentage of all system successful amounting 88%.

Keywords :

Ethernet Shield, Human Machine Interface, Microsoft Visual Studio, PJU

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Proyek Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Proyek Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN *HUMAN MACHINE INTERFACE* (HMI) PADA SISTEM *TELEMONITORING* JARINGAN PENERANGAN JALAN UMUM (PJU)

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Josaphat Pramudijanto, M.Eng dan Bapak Ciptian Wered P., S.ST, MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Proyek Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Proyek Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Proyek Akhir ini. Akhir kata, semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 17 Januari 2019

Devi Dwi Ariyani

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN JUDUL	iii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PENGESAHAN	vi
ABSTRAK.....	v
<i>ABSTRACT</i>	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Batasan Masalah.....	2
1.3 Tujuan.....	2
1.4 Sistematika Laporan	3
1.5 Relevansi.....	3
BAB II TEORI PENUNJANG.....	5
2.1 Penerangan Jalan Umum (PJU).....	5
2.2 <i>Human Machine Interface</i> (HMI).....	6
2.3 Arduino Mega 2560.....	6
2.3.1 Spesifikasi Arduino Mega 2560.....	7
2.3.2 <i>Input dan Output</i>	8
2.3.3 Catu Daya.....	10
2.3.4 Memori.....	11
2.3.5 Komunikasi	11
2.4 <i>Ethernet Shield</i>	12
2.5 Arduino IDE	13
2.6 <i>Modbus</i>	14
2.7 Microsoft Visual Studio 2017	17
BAB III PERANCANGAN DAN PEMBUATAN ALAT	19
3.1 Perancangan Sistem.....	19
3.2 Perancangan <i>Hardware</i>	20
3.2.1 Rangkaian Modul <i>Dummy</i>	20
3.2.2 Rangkaian <i>Ethernet Shield</i>	21
3.3 Perancangan <i>Software</i>	22

3.3.1	Perancangan Komunikasi <i>Wireless</i>	23
3.3.2	Perancangan Pengiriman Data	24
3.3.3	Perancangan <i>Human Machine Interface</i>	26
BAB IV	HASIL DAN PEMBAHASAN	35
4.1	Pengujian Koneksi <i>Router</i> dan PC	35
4.2	Pengujian Tes <i>Ping Network</i>	39
4.2.1	Pengujian Tes <i>Ping</i> PJU001	39
4.2.2	Pengujian Tes <i>Ping</i> PJU002	41
4.2.3	Pengujian Tes <i>Ping</i> PJU003	42
4.2.4	Pengujian Tes <i>Ping</i> PJU004	43
4.3	Pengujian Jarak Koneksi <i>Ethernet</i>	44
4.3.1	Pengujian Jarak 10 Meter	45
4.3.2	Pengujian Jarak 20 Meter	46
4.3.3	Pengujian Jarak 30 Meter	47
4.3.4	Pengujian Jarak 40 Meter	48
4.4	Pengujian Tampilan Status Normal/ <i>Error</i>	49
4.4.1	Pengujian Status Normal/ <i>Error</i> Pada Komponen PJU001	50
4.4.2	Pengujian Status Normal/ <i>Error</i> Pada Komponen PJU002	52
4.4.3	Pengujian Status Normal/ <i>Error</i> Pada Komponen PJU003	57
4.4.4	Pengujian Status Normal/ <i>Error</i> Pada Komponen PJU004	62
4.5	Keberhasilan dan Efektifitas HMI	68
4.5.1	Keberhasilan HMI	68
4.5.2	Efektifitas HMI	69
BAB V	PENUTUP	71
5.1	Kesimpulan	71
5.2	Saran	71
LAMPIRAN A	HASIL PENGUJIAN HMI	A-1
A.1	Pengujian PJU002	A-1
A.2	Pengujian PJU003	A-2
A.3	Pengujian PJU004	A-3
LAMPIRAN B	<i>SURVEY</i> EFEKTIFITAS HMI	B-1
LAMPIRAN C	<i>DATASHEET</i>	C-1
C.1	<i>Ethernet Shield</i>	C-1

C.2 Arduino Mega 2560..... C-6

LAMPIRAN D RIWAYAT PENULIS D-1

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

Gambar 2. 1	Tampilan Depan Arduino Mega 2560.....	7
Gambar 2. 2	Tampilan Belakang Arduino Mega 2560	7
Gambar 2. 3	<i>Ethernet Shield</i>	13
Gambar 2. 4	Tampilan Awal <i>Software</i> Arduino IDE.....	14
Gambar 2. 5	Contoh Arsitektur <i>Modbus</i>	16
Gambar 2. 6	Tampilan Awal <i>Microsoft Visual Studio 2017</i> .	17
Gambar 3. 1	Skema Proses Pengiriman Data.....	19
Gambar 3. 2	<i>Wiring</i> Arduino Mega 2560	20
Gambar 3. 3	Rancangan <i>Box</i> Modul <i>Dummy</i>	21
Gambar 3. 4	Realisasi <i>Box</i> Modul <i>Dummy</i>	21
Gambar 3. 5	Rangkaian <i>Ethernet Shield</i>	22
Gambar 3. 6	Ethernet Shield Dengan Arduino Mega 2560...	22
Gambar 3. 7	<i>Flowchart</i> Komunikasi <i>Wireless</i>	24
Gambar 3. 8	<i>Flowchart</i> Pengiriman Data	25
Gambar 3. 9	<i>Flowchart</i> Perancangan HMI	26
Gambar 3.10	Rancangan <i>Form Login</i>	27
Gambar 3.11	Realisasi <i>Form Login</i>	28
Gambar 3.12	Tampilan Notifikasi Saat Login Tanpa <i>Username</i> dan <i>Password</i>	29
Gambar 3.13	Tampilan Notifikasi Saat <i>Username/Password</i> Salah	29
Gambar 3.14	Tampilan Saat Berhasil <i>Login</i>	29
Gambar 3.15	Rancangan <i>Main Menu</i>	30
Gambar 3.16	Realisasi <i>Form Main Menu</i>	30
Gambar 3.17	Rancangan <i>Form Details</i>	32
Gambar 3.18	Realisasi <i>Form Details</i>	32
Gambar 4. 1	Tampilan <i>Login Website</i> TP-LINK.....	35
Gambar 4. 2	Tampilan Menu Informasi dan Status <i>Router</i> ...	36
Gambar 4. 3	Tampilan Menu <i>Operation Mode Router</i>	36
Gambar 4. 4	Tampilan Menu <i>Wireless Setting</i>	37
Gambar 4. 5	Tampilan Menu <i>DHCP Settings</i>	37
Gambar 4. 6	Koneksi Dengan <i>Router</i>	38
Gambar 4. 7	<i>Ping IP Address</i> PC	38
Gambar 4. 8	<i>Ping IP Address Router</i>	39
Gambar 4. 9	Program Tes <i>Ping</i> PJU001 Pada VB.....	39
Gambar 4.10	Program Tes <i>Ping</i> PJU001 Saat Berhasil Terkoneksi.....	40

Gambar 4.11	Program Tes <i>Ping</i> PJU001 Saat Gagal Terkoneksi	40
Gambar 4.12	Program Tes <i>Ping</i> PJU002 Saat Berhasil Terkoneksi	41
Gambar 4.13	Program Tes <i>Ping</i> PJU002 Saat Berhasil Terkoneksi	41
Gambar 4.14	Program Tes <i>Ping</i> PJU002 Saat Gagal Terkoneksi	41
Gambar 4.15	Program Tes <i>Ping</i> PJU003 Pada VB	42
Gambar 4.16	Program Tes <i>Ping</i> PJU003 Saat Berhasil Terkoneksi	42
Gambar 4.17	Program Tes <i>Ping</i> PJU003 Saat Gagal Terkoneksi	43
Gambar 4.18	Program Tes <i>Ping</i> PJU004 Pada VB	43
Gambar 4.19	Program Tes <i>Ping</i> PJU004 Saat Berhasil Terkoneksi	44
Gambar 4.20	Program Tes <i>Ping</i> PJU004 Saat Gagal Terkoneksi	44
Gambar 4.21	Pengujian Jarak Koneksi <i>Ethernet</i>	45
Gambar 4.22	Tampilan Status Normal/ <i>Error</i> Pada <i>Main Menu</i>	49
Gambar 4.23	Pengujian Status Normal/ <i>Error</i>	50
Gambar 4.24	Pengujian Status PJU001 Pada <i>Main Menu</i>	51
Gambar 4.25	Pengujian Status PJU001 Pada <i>Menu Details</i> ..	51
Gambar 4.26	Status Normal PJU002 Pada <i>Main Menu</i>	53
Gambar 4.27	Status Normal PJU002 Pada <i>Menu Details</i>	54
Gambar 4.28	Status <i>Error</i> PJU002 Pada <i>Main Menu</i>	55
Gambar 4.29	Status PV PJU002 Saat <i>Error</i>	56
Gambar 4.30	Status Baterai PJU002 Saat <i>Error</i>	56
Gambar 4.31	Status LED PJU002 Saat <i>Error</i>	57
Gambar 4.32	Status Normal PJU003 Pada <i>Main Menu</i>	58
Gambar 4.33	Status Normal PJU003 Pada <i>Menu Details</i>	59
Gambar 4.34	Status <i>Error</i> PJU003 Pada <i>Main Menu</i>	60
Gambar 4.35	Status PV PJU003 Saat <i>Error</i>	61
Gambar 4.36	Status Baterai PJU003 Saat <i>Error</i>	61
Gambar 4.37	Status LED PJU003 Saat <i>Error</i>	62
Gambar 4.38	Status Normal PJU004 Pada <i>Main Menu</i>	63
Gambar 4.39	Status Normal PJU004 Pada <i>Menu Details</i>	64
Gambar 4.40	Status <i>Error</i> PJU004 Pada <i>Main Menu</i>	65

Gambar 4.41	Status PV PJU004 Saat <i>Error</i>	66
Gambar 4.42	Status Baterai PJU004 Saat <i>Error</i>	66
Gambar 4.43	Status LED PJU004 Saat <i>Error</i>	67

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

Tabel 2. 1	Pin Serial RX dan TX	8
Tabel 2. 2	Pin Serial Eksternal Interupsi	9
Tabel 2. 3	Pin Serial SPI.....	9
Tabel 3. 1	<i>Mind Mapping</i> Arduino Mega 2560.....	20
Tabel 3. 2	Tampilan <i>Form Login</i>	28
Tabel 3. 3	Tampilan <i>Main Menu</i>	31
Tabel 3. 4	Tampilan <i>Form Details</i>	33
Tabel 4. 1	Pengukuran Jarak 10 Meter	45
Tabel 4. 2	Pengukuran Jarak 20 Meter	46
Tabel 4. 3	Pengukuran Jarak 30 Meter	47
Tabel 4. 4	Pengukuran Jarak 40 Meter	48
Tabel 4. 5	Keberhasilan HMI.....	68
Tabel 4. 6	Efektifitas HMI.....	69

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penerangan Jalan Umum (PJU) merupakan salah satu fasilitas umum yang sering dijumpai di jalan besar maupun kecil. Pentingnya pemasangan penerangan jalan umum selain sebagai penerangan, juga untuk mengurangi resiko kecelakaan. Mahalnya biaya pemasangan dan pemeliharaan PJU konvensional membuat banyak pihak beralih menggunakan PJU dengan tenaga surya. Biaya pemasangan dan pemeliharaan PJU tenaga surya lebih murah dibandingkan dengan PJU konvensional.

Penerangan Jalan umum tenaga surya (PJU TS) telah digunakan secara massal sebagai penerangan pengganti PJU konvensional. Dapat diketahui bahwa PJU TS memiliki beberapa keunggulan yaitu mampu menghasilkan listrik dari sinar matahari, mempunyai tempat penyimpanan energi dan ramah lingkungan. PJU TS biasa digunakan di jalan tol atau di jalan antar kota/provinsi yang tidak dapat teraliri listrik dari PLN. Namun sayangnya, penggunaan PJU TS tidak dapat optimal dikarenakan PJU TS memiliki kekurangan yang belum bisa di atasi dengan baik. Terdapat komponen utama yaitu solar panel sebagai sumber energi, baterai sebagai penyimpan daya, *solar charge controler* sebagai penghubung dan pemutus apabila baterai penuh/kosong dan lampu sebagai penerang jalan.

PJU TS memiliki komponen utama yang rawan mengalami kerusakan ringan, pada *photovoltaic* memiliki kondisi *short circuit* dan *open circuit* apabila kondisi tersebut terjadi tanpa di ketahui oleh teknisi maka baterai tidak akan terisi penuh dan lampu tidak akan menyala di malam hari. Selain itu baterai juga memiliki karakteristik yaitu baterai dapat mengalami kenaikan temperatur apabila baterai terlalu lama mengalami proses *charging/depth discharging*, lalu baterai juga memiliki nilai tertentu untuk berapa kali mengalami *charge discharge* agar dapat di ketahui apakah baterai masih layak pakai atau tidak.

Sekira 60 persen penerangan jalan umum tenaga surya (PJU-TS) di Tanjung Selor dalam kondisi rusak. Banyak masyarakat yang mengeluhkan kondisi tersebut. Keadaan ini dikhawatirkan bisa

mengakibatkan kecelakaan lalu lintas. Termasuk rawan tindakan kriminalitas pada malam hari. Namun, Dinas Perumahan Rakyat dan Kawasan Permukiman (DPR-KP) Kabupaten Bulungan belum dapat melakukan perbaikan secara maksimal karena keterbatasan anggaran. Hal tersebut mengabarkan tentang kerusakan PJU-TS, kerusakan di sebabkan perawatan terhadap PJU-TS yang tidak maksimal [1]. Sehingga dibutuhkan suatu alat untuk membantu perawatan pada PJU-TS . Perawatan dilakukan secara berkala dengan cara *memonitoring* kondisi PJU-TS. Oleh karena itu dibutuhkan suatu perangkat yang dapat mendiagnosis kerusakan pada PJU TS secara keseluruhan, agar PJU TS dapat berfungsi secara optimal.

Dibutuhkan pemantauan secara khusus dan terus menerus untuk mengetahui keadaan PJU TS secara efisien. Pemantauan dilakukan berdasarkan pengiriman data *self diagnosis* kepada *server* secara *wireless* dengan menggunakan *ethernet*. Data yang dikirim akan muncul di *server* berupa status dan nilai. *Server* menggunakan *Human Machine Interface* dengan Microsoft Visual Studio 2017.

1.2 Batasan Masalah

Pada proyek akhir ini, memiliki batasan-batasan masalah yang diambil, diantaranya :

1. Jumlah PJU yang dibuat hanya berjumlah 4 buah, dengan rincian 1 PJU *real*, dan 3 simulasi menggunakan modul *dummy*.
2. Menggunakan satu *power supply* sebagai catu daya untuk modul *dummy* PJU002,PJU003, dan PJU004.
3. Jumlah *port* PJU yang dapat terkoneksi pada *router* hanya 4 buah saja.

1.3 Tujuan

Tujuan dari pengerjaan Proyek Akhir ini adalah :

1. Mendesain *Human Machine Interface* (HMI) untuk memudahkan pemantauan kondisi tiap PJU.
2. Merancang komunikasi untuk mengirim data dari kondisi PJU agar dapat ditampilkan melalui HMI.

1.4 Sistematika Laporan

Pembahasan Proyek Akhir ini akan dibagi menjadi lima bab dengan sistematika sebagai berikut:

- Bab I Pendahuluan**
Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan dan relevansi.
- Bab II Teori Penunjang**
Bab ini menjelaskan tentang tinjauan pustaka dari Penerangan Jalan Umum (PJU), *Human Machine Interface* (HMI), Arduino Mega2560, Arduino IDE, Ethernet Shield, *Modbus*, dan Microsoft Visual Studio 2017.
- Bab III Perancangan dan Pembuatan Alat**
Bab ini membahas perencanaan dan pembuatan perangkat keras (*Hardware*) yang meliputi rangkaian Arduino Mega 2560, rangkaian Ether-net Shield dan pembuatan perangkat lunak (*Soft-ware*) yang meliputi program pada Arduino IDE untuk menjalankan alat tersebut, serta pembuatan program HMI menggunakan Microsoft Visual Studio 2017.
- Bab IV Hasil dan Pembahasan**
Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian jarak yang dapat dijangkau oleh modul ethernet shield, kesesuaian data yang dikirimkan, serta pembacaan data pada HMI.
- Bab V Penutup**
Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.5 Relevansi

Hasil yang diperoleh dari Proyek Akhir ini diharapkan dapat mempermudah pemantauan kondisi tiap PJU dari jarak jauh, sehingga tidak memakan banyak waktu untuk memeriksa kondisi masing-masing PJU.

-----Halaman ini sengaja dikosongkan-----

BAB II

TEORI PENUNJANG

Pada bab ini membahas mengenai teori yang dapat menunjang dalam proses pembuatan Proyek Akhir. Adapun pada bab ini terdiri dari 6 subbab, yaitu Penerangan Jalan Umum (PJU), *Human Machine Interface* (HMI), Arduino Mega 2560, Arduino IDE, Ethernet Shield, *Modbus*, dan Microsoft Visual Studio 2017.

2.1 Penerangan Jalan Umum (PJU) [2]

Penggunaan tenaga surya dengan solar panel merupakan solusi sumber daya listrik yang baru, awalnya sumber daya listrik yang sebagian besar kita gunakan berasal dari PLN yang menggunakan sumber batu bara untuk *supply* sumber energinya. Penggunaan panel surya ini merupakan energi yang ramah lingkungan dikarenakan sumber dayanya adalah energi cahaya matahari yang diubah menjadi energi listrik. Energi listrik yang didapat dari solar panel.

Solar panel digunakan pada Penerangan Jalan Umum (PJU), sudah banyak PJU yang menggunakan solar panel sebagai sumber dayanya sehingga PJU yang menggunakan solar panel ini disebut Penerangan Jalan Umum - Tenaga Surya (PJU-TS). Cara kerja PJU-TS, solar panel menangkap cahaya matahari untuk dijadikan energi listrik dan disimpan pada baterai. Fungsi baterai pada hal ini ialah menyimpan energi listrik digunakan untuk menyalakan lampu.

Solar panel menangkap cahaya untuk dijadikan sebagai energi listrik, namun ketidakstabilan cahaya berpengaruh pada energi listrik yang dihasilkan. Energi listrik yang dihasilkan akan disimpan pada baterai. Baterai mendapatkan masukan yang tidak stabil akan merusak baterai. Baterai memiliki karakteristik berupa kondisi untuk mulai pengisian dan pengosongan, kondisi tersebut ialah *state of charge* (SOC) dan *depth of discharge* DOD. Sehingga dibutuhkan perangkat lain untuk mengatur pengisian dan pengosongan, perangkat tersebut ialah *solar charge controller* selain berfungsi mengatur pengisian dan pengosongan, *solar charge controller* mengatur mati/hidupnya lampu.

2.2 *Human Machine Interface (HMI)*

HMI (*Human Machine Interface*) adalah sistem yang menghubungkan antara manusia dan teknologi mesin. HMI dapat berupa pengendali dan visualisasi status baik dengan manual maupun melalui visualisasi komputer yang bersifat *real time*. Sistem HMI biasanya bekerja secara *online* dan *real time* dengan membaca data yang dikirimkan melalui *I/O port* yang digunakan oleh sistem kontroler-nya. *Port* yang biasanya digunakan untuk kontroler dan akan dibaca oleh HMI antara lain adalah *port com*, *port USB*, *port RS232* dan ada pula yang menggunakan *port serial*. Tugas dari HMI (*Human Machine Interface*) yaitu membuat visualisasi dari teknologi atau sistem secara nyata. Sehingga dengan desain HMI dapat disesuaikan sehingga memudahkan pekerjaan fisik. Tujuan dari HMI adalah untuk meningkatkan interaksi antara mesin dan operator melalui tampilan layar komputer dan memenuhi kebutuhan pengguna terhadap informasi sistem [3].

HMI dalam dunia industri berupa suatu tampilan *GUI* (*Graphic User Interface*) pada suatu tampilan layar komputer yang akan dihadapi oleh operator mesin maupun pengguna yang membutuhkan data kerja mesin. HMI terdapat berbagai macam visualisasi untuk monitoring dan data mesin yang terhubung secara *online* dan *real time*. Pada HMI terdapat visualisasi pengendali mesin berupa tombol, *slider*, dan sebagainya yang dapat difungsikan untuk mengendalikan mesin sebagaimana mestinya. Sebagai tambahan, HMI juga menampilkan data-data rangkuman kerja mesin termasuk secara grafik [4].

2.3 *Arduino Mega 2560* [5]

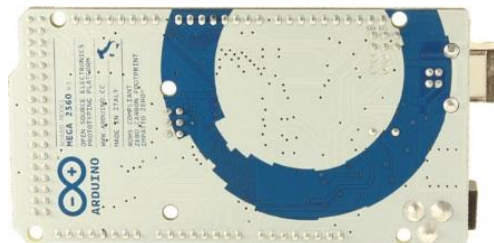
Arduino Mega 2560 adalah papan *microcontroller* berbasis Atmega 2560. Arduino Mega 2560 memiliki 54 pin digital *input/output*, dimana 15 pin dapat digunakan sebagai *output PWM*, 16 pin sebagai *input* analog, dan 4 pin sebagai *UART* (*port serial hardware*), 16 MHz kristal osilator, koneksi USB, *jack power*, *header ICSP*, dan tombol *reset*. Ini semua yang diperlukan untuk mendukung *microcontroller*. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau *power* dihubungkan dengan adaptor AC – DC atau baterai untuk mulai mengaktifkannya. Arduino Mega 2560 kompatibel dengan sebagian besar *shield* yang dirancang untuk Arduino Duemilanove atau

Arduino Diecimila. Arduino Mega 2560 adalah versi terbaru yang menggantikan versi Arduino Mega.

Arduino Mega 2560 berbeda dari papan sebelumnya, karena versi terbaru sudah tidak menggunakan *chip driver* FTDI USB-to-serial. Tapi, menggunakan *chip* Atmega 16U2 (Atmega 8U2 pada papan Revisi 1 dan Revisi 2) yang diprogram sebagai konverter USB-to-serial. Arduino Mega 2560 Revisi 2 memiliki resistor penarik jalur HWB 8U2 ke *Ground*, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU. Adapun tampilan fisik Arduino dari depan dapat dilihat pada Gambar 2.1 dan tampilan Arduino dari belakang pada Gambar 2.2.



Gambar 2.1 Tampilan Depan Arduino Mega 2560 [5]



Gambar 2.2 Tampilan Belakang Arduino Mega 2560 [5]

2.3.1 Spesifikasi Arduino Mega 2560

Arduino Mega 2560 Revisi 3 memiliki fitur-fitur baru seperti berikut:

- Pinout : Ditambahkan pin SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, IOREF memungkinkan *shield* untuk beradaptasi dengan tegangan yang tersedia pada

papan. Di masa depan, *shield* akan kompatibel baik dengan papan yang menggunakan AVR yang beroperasi dengan 5 Volt dan dengan Arduino Due yang beroperasi dengan tegangan 3.3 Volt. Dan ada dua pin yang tidak terhubung, yang disediakan untuk tujuan masa depan.

- Sirkuit RESET.
- *Chip* ATmega16U2 menggantikan *chip* Atmega 8U2.

2.3.2 *Input dan Output*

Masing-masing dari 54 digital pin pada Arduino Mega 2560 dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Arduino Mega 2560 beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up* internal (yang terputus secara *default*) sebesar 20 – 50 kilo ohms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

- Serial yang digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel Pin Serial RX dan TX [5]

Nomor Pin	Nama Pin	Peta Nama Pin
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
45	PD2 (RXD1/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)

- **Eksternal Interupsi:** Pin ini dapat dikonfigurasi untuk memicu sebuah *interupsi* pada nilai yang rendah, meningkat atau menurun, atau berubah nilai. Pin eksternal interupsi dapat dilihat pada Tabel 2.2.

Tabel 2.2 Pin Serial Eksternal Interupsi [5]

Nomor Pin	Nama Pin	Peta Nama Pin
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1(SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3(TXD1/INT3)	Digital pin 18 (TX1)

- **SPI:** Pin ini mendukung komunikasi *SPI* menggunakan *SPI library*. Pin *SPI* juga terhubung dengan *header ICSP*, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove dan Arduino Diecimila. Adapun jenis pin *SPI* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Pin Serial *SPI* [5]

Nomor Pin	Nama Pin	Peta Nama Pin
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MIS)

- **LED:** Pin 13. Tersedia secara *built-in* pada papan Arduino ATmega LED terhubung ke pin digital 13. Ketika pin diset bernilai *HIGH*, maka LED menyala (*ON*), dan ketika pin diset bernilai *LOW*, maka LED padam (*OFF*).
- **TWI:** Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi *TWI* menggunakan *Wire library*. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin *TWI* pada Arduino Duemilanove atau Arduino Diecimila.

Arduino Mega 2560 memiliki 16 pin sebagai analog *input*, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara *default* pin ini dapat diukur/diatur dari mulai *Ground* sampai dengan 5 Volt, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan

pin AREF dan fungsi *Analog Reference()*. Ada beberapa pin lainnya yang tersedia, antara lain:

- **AREF**: Referensi tegangan untuk *input* digunakan dengan fungsi *Analog Reference()*.
- **RESET**: Jalur LOW ini digunakan untuk me-*reset* (menghidupkan ulang) *microcontroller*. Jalur ini biasanya digunakan untuk menambahkan tombol *reset* pada *shield* yang menghalangi papan utama Arduino.

2.3.3 Catu Daya

Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Sumber daya eksternal (non-USB) dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke ke *jack* sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui header pin Gnd dan pin Vin dari konektor POWER.

Papan Arduino ATmega 2560 dapat beroperasi dengan pasokan daya eksternal 6 Volt sampai 20 Volt. Jika diberi tegangan kurang dari 7 Volt, maka, pin 5 Volt mungkin akan menghasilkan tegangan kurang dari 5 Volt dan ini akan membuat papan menjadi tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 Volt, regulator tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 Volt sampai 12 Volt.

Pin tegangan yang tersedia pada papan Arduino adalah sebagai berikut:

- **VIN**: Adalah *input* tegangan untuk papan Arduino ketika menggunakan sumber daya eksternal (sebagai ‘saingan’ tegangan 5 Volt dari koneksi USB atau sumber daya ter-*regulator* lainnya). Anda dapat memberikan tegangan melalui pin ini, atau jika memasok tegangan untuk papan melalui *jack power*, kita bisa mengakses/mengambil tegangan melalui pin ini.
- **5V**: Sebuah pin yang mengeluarkan tegangan ter-*regulator* 5 Volt, dari pin ini tegangan sudah diatur (ter-*regulator*) dari *regulator* yang tersedia (*built-in*) pada papan. Arduino dapat diaktifkan dengan sumber daya baik berasal

dari *jack power* DC (7-12 Volt), konektor USB (5 Volt), atau pin VIN pada *board* (7-12 Volt). Memberikan tegangan melalui pin 5V atau 3,3V secara langsung tanpa melewati *regulator* dapat merusak papan Arduino.

- **3V3:** Sebuah pin yang menghasilkan tegangan 3,3 Volt. Tegangan ini dihasilkan oleh *regulator* yang terdapat pada papan (*on-board*). Arus maksimum yang dihasilkan adalah 50 mA.
- **GND:** Pin *Ground* atau *Massa*.
- **IOREF:** Pin ini pada papan Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada *microcontroller*. Sebuah perisai (*shield*) dikonfigurasi dengan benar untuk dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan (*voltage translator*) pada *output* untuk bekerja pada tegangan 5 Volt atau 3,3 Volt.

2.3.4 Memori

Arduino ATmega 2560 memiliki 256 KB *flash memory* untuk menyimpan kode (8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

2.3.5 Komunikasi

Arduino Mega 2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan *microcontroller* lainnya. Arduino ATmega 328 menyediakan 4 *hardware* komunikasi serial UART TTL (5 Volt). Sebuah *chip* ATmega 16U2 (ATmega 8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi serial melalui USB dan muncul sebagai *COM Port Virtual* (pada *Device* komputer) untuk berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi Windows masih tetap memerlukan file inf, tetapi untuk sistem operasi OS X dan Linux akan mengenali papan sebagai *port* COM secara otomatis. Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui *chip* USB-to-serial yang terhubung melalui

USB komputer (tetapi tidak untuk komunikasi serial seperti pada pin 0 dan 1).

Sebuah *Software Serial library* memungkinkan untuk komunikasi serial pada salah satu pin digital Mega 2560. ATmega 2560 juga mendukung komunikasi TWI dan SPI. Perangkat lunak Arduino termasuk *Wirelibrary* digunakan untuk menyederhanakan penggunaan bus TWI. Untuk komunikasi SPI, menggunakan *SPI library*.

2.4 *Ethernet Shield* [6]

Ethernet Shield adalah modul yang digunakan untuk mengkoneksikan arduino dengan internet menggunakan kabel (*Wired*). *Ethernet Shield* seperti pada Gambar 2.3 dibuat berdasarkan pada Wiznet W5100 *Ethernet chip*. Wiznet W5100 menyediakan IP untuk TCP dan UDP, yang mendukung hingga 4 *socket* secara simultan. Untuk menggunakannya dibutuhkan *library Ethernet* dan SPI. Dan *Ethernet Shield* ini menggunakan kabel RJ-45 untuk mengkoneksikanya ke Internet, dengan *integrated line transformer* dan juga *Power over Ethernet*.

Ethernet Shield bekerja dengan cara memberikan layanan IP pada arduino dan PC agar dapat terhubung ke internet. Cara menggunakan cukup mudah yaitu hanya dengan menghubungkan Arduino *Ethernet Shield* dengan *board* Arduino lalu akan disambungkan ke jaringan internet. Cukup memasukkan modul ini ke *board* Arduino, lalu menghubungkannya ke jaringan internet dengan kabel RJ-45, maka Arduino akan terkoneksi langsung ke internet. Dan untuk menggunakannya, tentu saja kita harus men-*setting* IP pada modul dan PC internet agar dapat terhubung satu sama lain.

Selain itu modul ini juga terdapat sebuah *onboard micro-SD* slot, yang dapat digunakan untuk menyimpan *file* dan data. Modul *Ethernet Shield* bisa digunakan dengan *board* Arduino Uno dan Mega. Dan dapat bekerja dengan baik pada kedua Arduino tersebut. Untuk menggunakan akses *microSD card reader onboard* ini dapat dengan menggunakan *library SD card*. Saat menggunakan *library* ini, SS ditempatkan pada Pin 4. Modul ethernet juga terdapat pula *reset controller*, untuk memastikan bahwa modul W5100 *Ethernet* dapat *reset on power-up*.

Agar *board* Arduino dapat berkomunikasi baik dengan modul W5100 dan SD card menggunakan SPI bus melalui ICSP

header, yang ada pada *board* Arduino Uno di pin digital 10, 11, 12, dan 13, sedangkan pada *board* Arduino Mega pada pin digital 50, 51, dan 52. Di kedua *board* Arduino tersebut, pin digital 10 digunakan untuk memilih mode W5100 dan pin digital 4 untuk SD card, dimana pin tersebut tidak dapat digunakan untuk pin I/O biasa. Dalam *board* Arduino Mega, pin digital 53 (SS) tidak digunakan sama sekali, baik untuk memilih antara modul W5100 atau SD card, namun harus tetap ditetapkan sebagai *output* agar *interface* SPI dapat bekerja dengan baik.

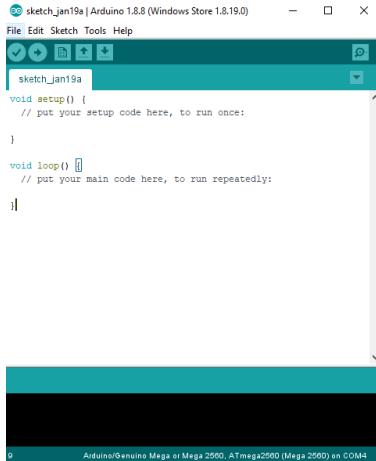


Gambar 2.3 Ethernet Shield [6]

2.5 Arduino IDE [7]

Arduino IDE adalah perangkat lunak yang digunakan untuk memrogram, *monitoring* dan *debugging* mikrokontroler *Arduino*. IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan.

Disebut sebagai lingkungan karena melalui *software* inilah *Arduino* dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. *Arduino* menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman *Arduino* (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler *Arduino* telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* *Arduino* dengan mikrokontroler. Pada Gambar 2.4 merupakan tampilan awal untuk membuat program pada *software* *Arduino IDE*.



Gambar 2.4 Tampilan Awal *Software* Arduino IDE [7]

2.6 *Modbus* [8]

Modbus adalah salah satu protokol untuk komunikasi serial yang di publikasikan oleh Modicon pada tahun 1979 untuk di gunakan pada PLC Modicon (PLC pertama di dunia yang di kembangkan oleh Schneider). Secara sederhana, *modbus* merupakan metode yang digunakan untuk mengirimkan data/informasi melalui koneksi serial antar perangkat elektronik. Perangkat yang meminta informasi disebut *Modbus Master* dan perangkat penyediaan informasi disebut *Modbus Slave*. Pada jaringan *Modbus* standar, terdapat sebuah *master* dan *slave* sampai dengan 247, masing-masing mempunyai Alamat *Slave* yang berbeda mulai dari 1 sampai 247. *Master* juga dapat menulis informasi kepada *Slave*.

Modbus merupakan sebuah *open* protokol, yang berarti bahwa dapat digunakan dalam peralatan tanpa harus membayar royalti. *Modbus* telah menjadi protokol komunikasi standar dalam industri, dan sekarang paling banyak dipakai untuk menghubungkan perangkat elektronik industri. *Modbus* digunakan secara luas oleh banyak produsen di banyak industri.

Protokol ini menjadi standard komunikasi dalam industri dan menjadi yang paling banyak dipakai untuk komunikasi antar peralatan elektronik pada industri. Alasan utama mengapa *Modbus* Protokol banyak di gunakan adalah:

1. Di publikasikan secara terbuka tanpa *royalty fee* untuk penggunaannya.
2. Relatif mudah untuk di aplikasikan pada *industrial network*.
3. *Modbus* mempunyai struktur bit tanpa memiliki banyak larangan bagi vendor lain untuk mengaksesnya.

Modbus memungkinkan adanya komunikasi dua-jalur antar perangkat yang terhubung ke jaringan yang sama, misalnya suatu sistem yang mengukur suhu, tekanan, kelembaban dsb, kemudian mengkomunikasikan hasilnya ke komputer (*HMI/Human Machine Interface*).

Berdasarkan media transfernya, *Modbus* dikategorikan ke dalam *modbus* serial (RS232/485) dan *Modbus* Ethernet (TCP/IP). Jika dirujuk dari bentuk datanya, *Modbus* dibagi ke dalam *modbus* RTU (serial) dan *modbus* ASCII. Pada *modbus* serial digunakan istilah *Master/Slave* sedangkan *Modbus* Ethernet biasanya memakai terminologi *Server/Client*.

Protokol *modbus* memungkinkan komunikasi yang mudah di semua jenis arsitektur jaringan. Setiap jenis perangkat (PLC, HMI, Kontrol Panel, Driver, kontrol *Motion*, I/O *Device* dll) dapat menggunakan protokol *Modbus* untuk operasi remot. Komunikasi yang sama dapat dilakukan juga pada serial line seperti pada Ethernet TCP / IP. *Gateway* memungkinkan komunikasi antara beberapa jenis bus atau jaringan dengan menggunakan protokol *modbus*.

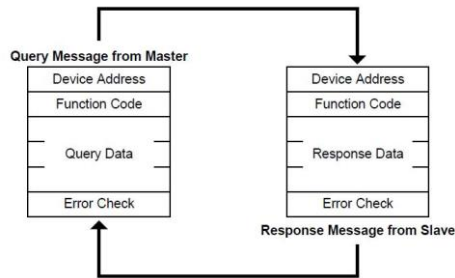
Perangkat *modbus* berkomunikasi menggunakan teknik *master-slave*, dimana hanya satu perangkat (*master*) yang dapat melakukan transaksi atau melakukan perintah atau permintaan yang disebut "*queries*". Perangkat lain (*slave*) merespon dengan menyediakan data yang diminta untuk *master*, atau dengan melakukan aksi sesuai dengan yang diminta dalam *query*.

Master dapat mengirimkan *request* ke satu *slave* secara individu, atau dapat mengirim pesan *broadcast* ke semua *slave*. *Slave* memberikan respon untuk pertanyaan yang ditujukan kepada mereka secara individu. Sedangkan *query broadcast* dari *master* tidak akan diberikan respon oleh *slave*.

Protokol *modbus* mempunyai format tertentu untuk setiap *query* dari master yang berisi alamat dari perangkat (*slave*) yang dituju, kode fungsi yang mendefinisikan aksi yang diminta, data

yang akan dikirim, dan pemeriksaan kesalahan. Pesan respon *slave* juga mempunyai format tertentu dalam protokol *Modbus*. Format ini berisi tentang konfirmasi tindakan yang dilakukan, data yang akan dikirim, dan bidang pemeriksaan kesalahan.

Proses *Query-Response* pada protokol *modbus* mempunyai format seperti Gambar 2.5.



Gambar 2.5 Contoh Arsitektur *Modbus* [8]

Format untuk *query master*, *device address* merupakan alamat *slave* yang akan diambil datanya, *Function Code* merupakan kode fungsi yang mendefinisikan aksi yang diminta, dimana beberapa contoh kode fungsi tersebut adalah sebagai berikut:

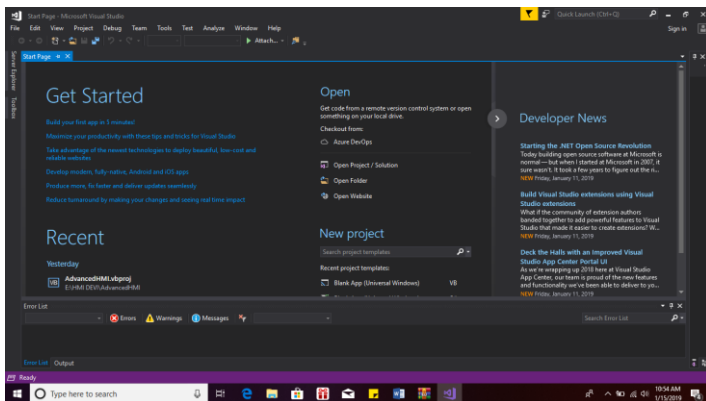
- 01 : *read DO (Digital Output)*
- 02 : *read DI (Digital Input)*
- 03 : *read AO (Analog Output)*
- 04 : *read AI (Analog Input)*
- 05 : *write single DO (Digital Output)*
- 06 : *write single AO (Analog Output)*
- 15 : *write multiple DO (Digital Output)*
- 16 : *write multiple AO (Analog Output)*.

Query data merupakan blok data informasi dan *Error Check* merupakan pemeriksaan kesalahan (cek data dari kesalahan komunikasi). Respon *slave* pada protokol *Modbus* juga mempunyai format yang sama, hanya pada *function code* berisi konfirmasi tindakan yang dilakukan, *Respond* merupakan data yang akan dikembalikan, dan *error check* sebagai bidang pemeriksaan kesalahan.

2.7 Microsoft Visual Studio 2017 [9]

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi windows, ataupun aplikasi web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic.NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe .

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, visual studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework). Tampilan Awal Microsoft Visual Studio 2015 ditunjukkan pada Gambar 2.6.



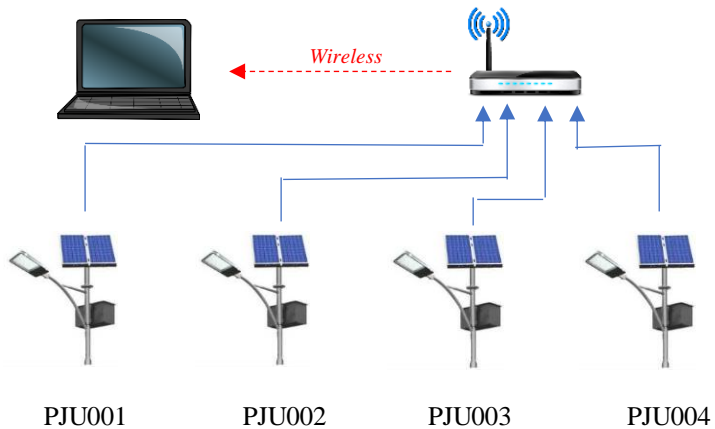
Gambar 2.6 Tampilan Awal Microsoft Visual Studio 2017 [9]

-----Halaman ini sengaja dikosongkan-----

BAB III PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini membahas mengenai perancangan *Human Machine Interface* (HMI) pada Sistem *Telemonitoring* Jaringan Penanganan Jalan Umum (PJU). Perancangan proyek akhir ini terbagi menjadi dua, yaitu perancangan sistem, *hardware* dan *software*.

3.1 Perancangan Sistem



Gambar 3.1 Skema Proses Pengiriman Data

Pada sistem ini terdapat empat *Client* yang terdiri dari 1 PJU *real* (PJU001) serta PJU simulasi (Modul *dummy* PJU002, PJU003, PJU004). Proses mengirim data dari *Client* ke *Server* dapat dilihat pada Gambar 3.1.

Proses pengiriman data antara setiap *Client* dengan *Master* menggunakan konsep protokol *Modbus* TCP/IP. Sehingga, masing-masing *client* harus tersambung dengan *router* yang akan mengirimkan data menuju *Server*.

Tiap *client* wajib memiliki IP yang berbeda. Data yang dikirim berjumlah 10 data dari masing-masing PJU. Masing-masing data diberi alamat berbeda yang bertujuan untuk memudahkan dalam penampilan data di HMI.

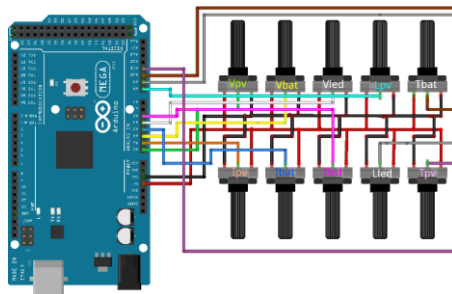
3.2 Perancangan Hardware

Pada perancangan *hardware* dibagi menjadi beberapa sub bab yang akan dijelaskan per sub bab nya, antara lain :

3.2.1 Rangkaian Modul *Dummy*

Modul *dummy* digunakan sebagai rangkaian *client*. Modul *dummy* menggunakan arduino mega 2560 untuk mengolah data yang akan dikirimkan *Ethernet Shield* menuju *server*.

Berikut merupakan perincian pin Arduino mega 2560 yang digunakan pada *hardware client*, dapat dilihat pada Gambar 3.2 dan Tabel 3.1.

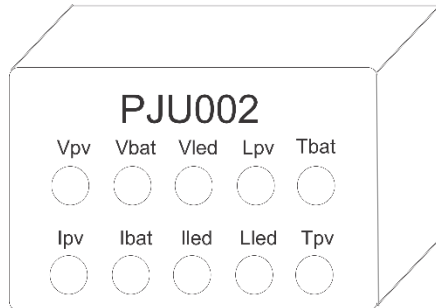


Gambar 3.2 Wiring Arduino Mega 2560

Tabel 3.1 Mind Mapping Pin Arduino Mega 2560

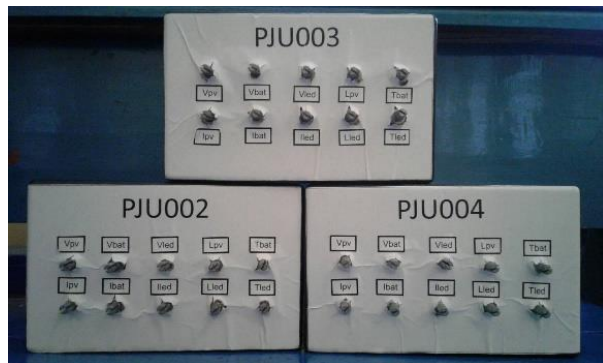
Nomor Pin	Keterangan
5V	Vcc
Gnd	Ground
A0	Vpv
A1	Ipv
A2	Vbat
A3	Ibat
A4	Vled
A5	Iled
A8	Lpv
A9	Lled
A10	Tbat
A11	Tpv

Agar modul *dummy* terlihat lebih rapi, maka diperlukan *box* untuk meletakkan rangkaian modul *dummy* adapun rancangan *box* yang akan digunakan untuk modul *dummy* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Rancangan *Box* Modul *Dummy*

Adapun realisasi dari *box* Modul *dummy* dapat dilihat ada Gambar 3.4. Modul *dummy* yang dibuat sebanyak 3 buah untuk mewakili 3 PJU.

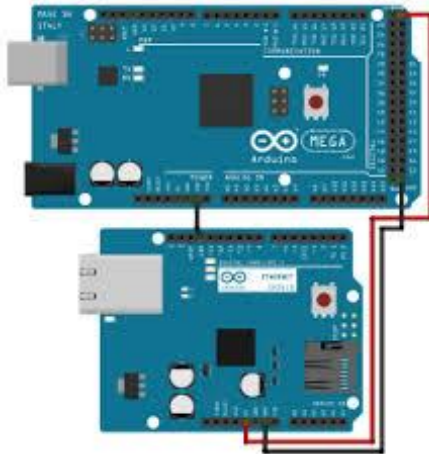


Gambar 3.4 Realisasi *Box* Modul *Dummy*

3.2.2 Rangkaian *Ethernet Shield*

Ethernet Shield digunakan sebagai komunikasi pengiriman data dari *slave* menuju *master*. Rangkaian *Ethernet Shield* hanya digunakan pada bagian *slave* saja.

Berikut merupakan rangkaian *Ethernet Shield* dengan Arduino Mega 2560 yang dapat dilihat pada Gambar 3.5.



Gambar 3.5 Rangkaian *Ethernet Shield*

Namun, rangkaian *Ethernet Shield* dapat disederhanakan lagi dengan cara meletakkannya diatas Arduino Mega 2560 seperti pada Gambar 3.6.



Gambar 3.6 *Ethernet Shield* dengan Arduino Mega 2560

3.3 Perancangan *Software*

Berikut merupakan perancangan perangkat lunak untuk monitoring kondisi Penerangan Jalan Umum (PJU), yang terdiri

dari perancangan komunikasi *wireless*, perancangan pembacaan data, dan perancangan HMI.

3.3.1 Perancangan Komunikasi *Wireless*

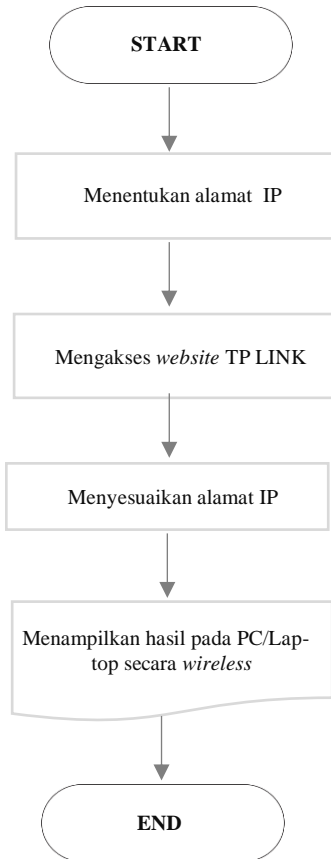
Agar keseluruhan fungsi dalam alat ini dapat berjalan dengan baik mulai dari *hardware* hingga *software*, maka diperlukannya komunikasi *wireless* dalam perancangannya. Penentuan kelas jaringan yang akan digunakan hingga penggunaan *software* dalam pembuatan HMI menjadi bagian utama.

Untuk mengatur alamat masing-masing komputer pada suatu jaringan, digunakanlah *IP Address*. *IP Address* atau alamat IP adalah suatu alamat yang diberikan ke peralatan jaringan komputer untuk dapat diidentifikasi oleh komputer yang lain. Dengan demikian masing-masing komputer dapat melakukan proses tukar-menukar data/informasi atau mengakses ke suatu jaringan komputer dengan menggunakan protokol *modbus TCP/IP*. Sedangkan untuk pembuatan tampilan HMI, menggunakan *software* Microsoft Visual Studio 2017.

Kemudian agar HMI yang telah dibuat dapat dijalankan secara *wireless*, maka diperlukannya penggunaan *router*. *Router* yang digunakan pada tugas akhir ini adalah TP-LINK. Berikut adalah perancangan komunikasi *wireless* dalam tugas akhir ini:

1. Menentukan alamat IP pada *wireless router* TP-LINK.
2. Mengakses ke *website* TP-LINK dan mengubah alamat IP.
3. Koneksikan PC dengan jaringan TP-LINK pada *Network & Internet Setting*
4. Sambungkan *Ethernet* dengan slot LAN *router* menggunakan kabel LAN.
5. Buka *command prompt* dan lakukan ping pada IP *router* dan IP *Ethernet* yang telah di-*setting*.

Untuk lebih mempermudah penjelasan maka langkah-langkah di atas dapat diterjemahkan ke dalam bentuk *flowchart* pada Gambar 3.7.



Gambar 3.7 Flowchart Komunikasi Wireless

3.3.2 Perancangan Pengiriman Data

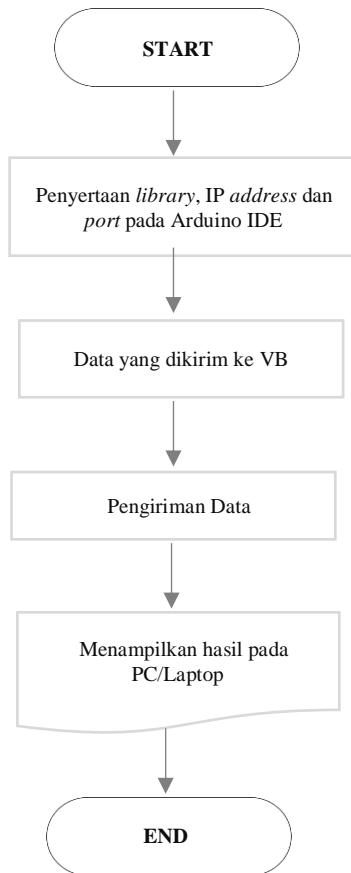
Software yang digunakan untuk melakukan pemrograman terhadap pembacaan data adalah menggunakan *Arduino IDE*.

Ada beberapa program yang akan dibuat untuk menjalankan peralatan pada Proyek Akhir ini. Untuk memperjelas proses kerja dari alat ini berikut adalah algoritma pemrogramannya :

1. Penyertaan *library Ethernet* pada *software Arduino IDE*.

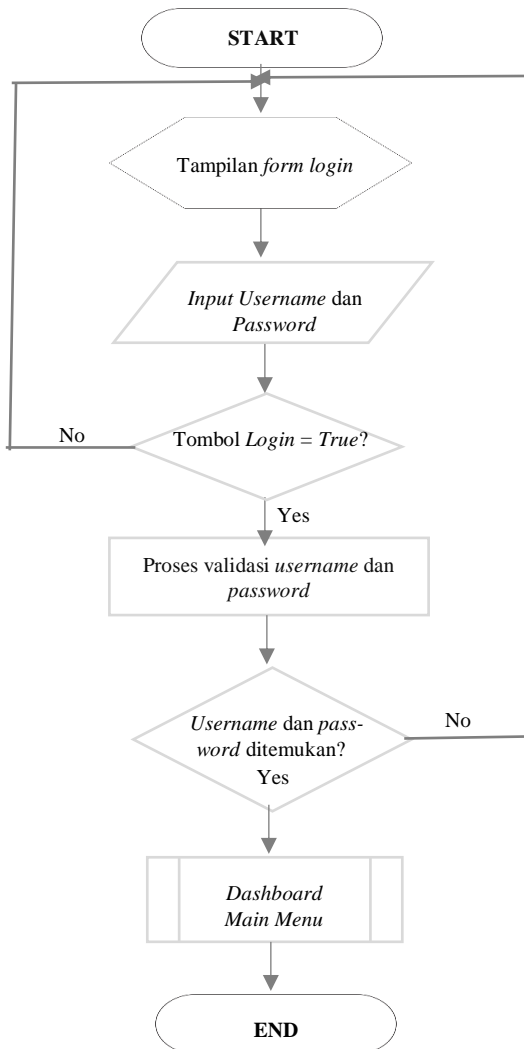
2. Penyettingan atau pengaturan alamat IP dan *port* yang digunakan.
3. Pembuatan program yang akan dikirimkan ke *Ethernet Shield* dan ditampilkan pada Microsoft Visual Studio 2017. Data yang telah dikirim akan diterima oleh laptop.

Untuk lebih mempermudah penjelasan maka algoritma, dapat diterjemahkan ke dalam bentuk *flowchart* pada Gambar 3.8.



Gambar 3.8 *Flowchart* Pengiriman Data

3.3.3 Perancangan *Human Machine Interface* (HMI)

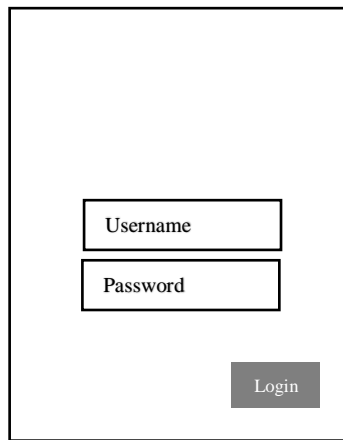


Gambar 3.9 *Flowchart* Perancangan HMI

Pada *flowchart* Gambar 3.9 menjelaskan mengenai rancangan atau tahap-tahap pengoperasian HMI. Langkah pertama adalah memasuki *form login* yang berisi *username* dan *password*. Jika *username* dan *password* benar, maka akan muncul notifikasi berhasil *login*, klik OK maka HMI akan menampilkan *main menu*. Namun, jika salah menginputkan *username* dan *password* pada *form login*, maka akan muncul notifikasi untuk menginputkan *username* dan *password* yang benar. Berikut merupakan rancangan tampilan HMI.

1. *Form login*

Form login berisi *username* dan *password* berfungsi untuk *security* dalam proses *monitoring*, agar tidak semua orang dapat mengaksesnya. Rancangan *form login* dapat dilihat pada Gambar 3.10.



The image shows a simple login form layout. It consists of a large rectangular frame containing three elements: a text input field labeled 'Username', a text input field labeled 'Password', and a button labeled 'Login' located in the bottom right corner of the frame.

Gambar 3.10 Rancangan *Form Login*



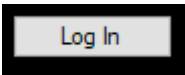
Adapun realisasi dari rancangan *Form Login* dapat dilihat pada Gambar 3.11



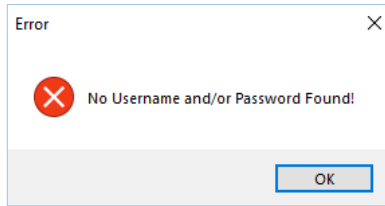
Gambar 3.11 Realisasi *Form Login*

Dapat dilihat pada Gambar 3.11 merupakan bentuk tampilan dari HMI *form login* yang digunakan sebagai proteksi agar tidak semua pengguna dapat mengakses aplikasi ini. Berikut penjelasan tampilan HMI pada *form login* yang dapat dilihat pada Tabel 3.2.

Tabel 3.2 Tampilan *Form Login*

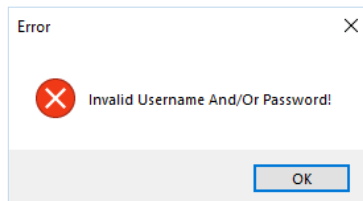
Ikon	Keterangan
	Masukkan <i>username</i> pada <i>textbox</i>
	Masukkan <i>password</i> pada <i>textbox</i> , <i>password</i> yang ditulis ditampilkan dalam bentuk **** untuk keamanan
	Tombol login untuk menuju ke <i>form</i> berikutnya

Apabila tombol *login* ditekan tanpa memasukkan *username* dan *password* kedalam *textbox*, maka akan muncul notifikasi seperti pada Gambar 3.12.



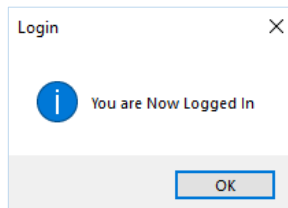
Gambar 3. 12 Tampilan Notifikasi Saat *Login Tanpa Username dan Password*

Apabila terdapat kesalahan pada *username* atau *password* kedalam *textbox*, maka akan muncul notifikasi seperti pada Gambar 3.13.



Gambar 3. 13 Tampilan Notifikasi Saat *Username/Password Salah*

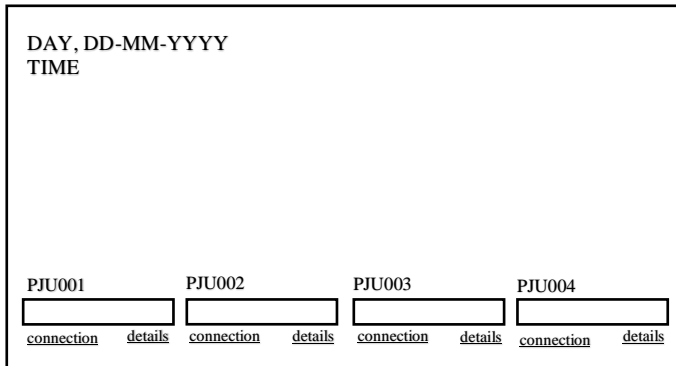
Apabila *username* dan *password* benar, maka akan muncul tampilan seperti pada Gambar 3.14.



Gambar 3.14 Tampilan Saat Berhasil *Login*

2. Form Main Menu

Form main menu berisi status *Normal/Error* masing-masing PJU. Terdiri dari fitur *details* untuk melihat secara rinci kondisi di masing-masing PJU. *Fitur connection* untuk memeriksa koneksi jaringan. Rancangan *form main menu* dapat dilihat pada Gambar 3.15.



DAY, DD-MM-YYYY
TIME

PJU001	PJU002	PJU003	PJU004
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
connection details	connection details	connection details	connection details

Gambar 3.15 Rancangan *Main Menu*

Adapun realisasi dari *Form Main Menu* dapat dilihat pada Gambar 3.16

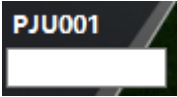

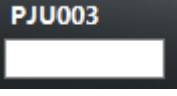

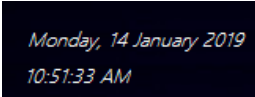



Gambar 3.16 Realisasi *Form Main Menu*

Pada Gambar 3.16 akan menampilkan data masing-masing PJU. Apabila terdapat *error* pada salah satu bagian PJU (PV, led, baterai), maka pada menu utama akan menampilkan status “ERROR”. Namun apabila tidak terdapat kesalahan pada bagian PJU, maka status pada menu utama adalah “NORMAL”.

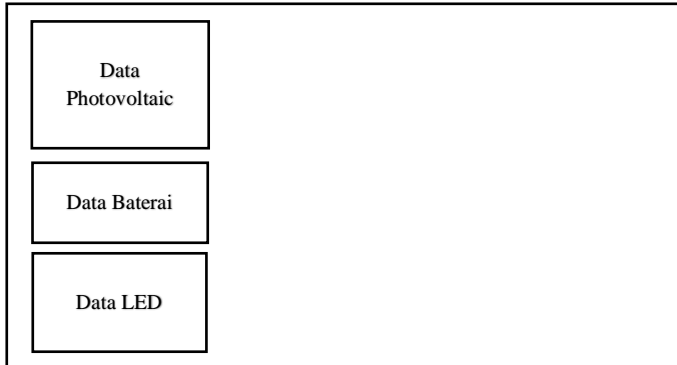
Berikut merupakan penjelasan dari *form* menu utama yang dapat dilihat pada Tabel 3.3.

Tabel 3.3 Tampilan *Main Menu*

Ikon	Keterangan
	Menampilkan status PJU001
	Menampilkan status PJU002
	Menampilkan status PJU003
	<i>Link details</i> untuk membuka form masing-masing PJU yang berisi rincian lengkap dari PJU
	Memuat hari, tanggal, tahun dan waktu akses aplikasi
	<i>Connection</i> berfungsi untuk mengetahui koneksi jaringan tiap PJU

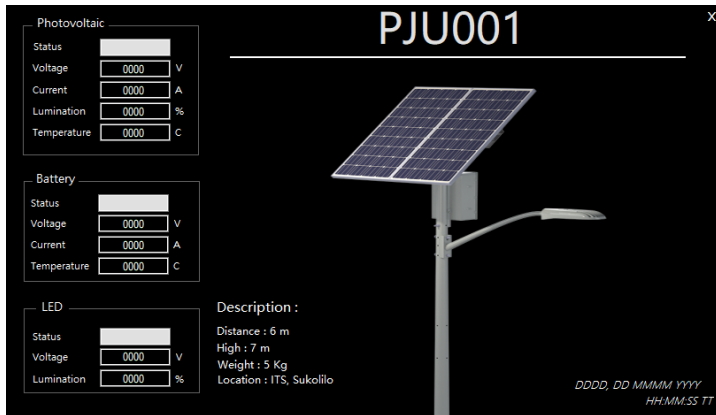
3. Form details

Form main menu berisi status *Normal/Error* masing-masing PJU. Terdiri dari fitur *details* untuk melihat secara rinci kondisi di masing-masing PJU. *Fitur connection* untuk memeriksa koneksi jaringan. Rancangan *form main menu* dapat dilihat pada Gambar 3.17.



Gambar 3.17 Rancangan *Form Details*






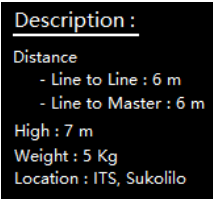
Adapun realisasi dari *Form Details* dapat dilihat pada Gambar 3.18.



Gambar 3.18 Realisasi *Form Details*

Berikut merupakan penjelasan dari *form details* yang dapat dilihat pada Tabel 3.4.

Tabel 3.4 Tampilan *Form Details*

Ikon	Keterangan
	Menampilkan kondisi “ERROR” atau “NORMAL”
	Menampilkan data tegangan dengan satuan Volt (V)
	Menampilkan data arus dengan satuan Ampere (A)
	Menampilkan data luminasi dengan satuan %
	Menampilkan data temperatur dengan satuan °C
	Berupa deskripsi PJU yang berisi jarak antar PJU maupun master, tinggi PJU, berat PJU, serta lokasi PJU

-----Halaman ini sengaja dikosongkan-----

BAB IV

HASIL DAN PEMBAHASAN

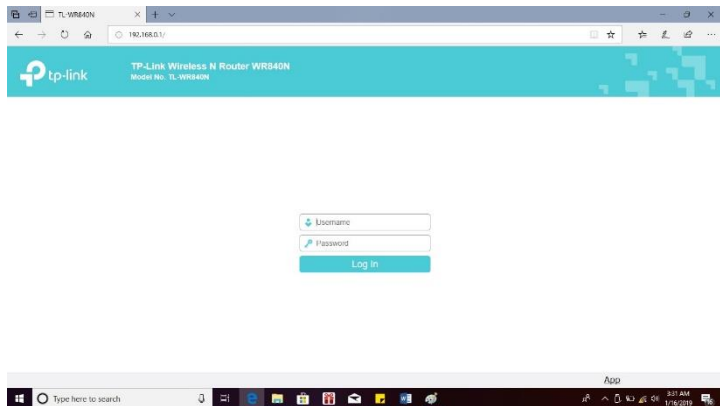
Dalam perencanaan dan pembuatan suatu sistem, pengujian dan analisa sangatlah diperlukan, karena dengan pengujian dan analisa dimaksudkan untuk mengetahui kinerja dari sistem dan komponen pendukung yang dibuat sudah sesuai dengan yang direncanakan atau belum.

Pada bab ini akan dibahas tentang pengujian *telemonitoring* menggunakan HMI dengan komunikasi Ethernet.

4.1 Pengujian Koneksi *Router* dan PC

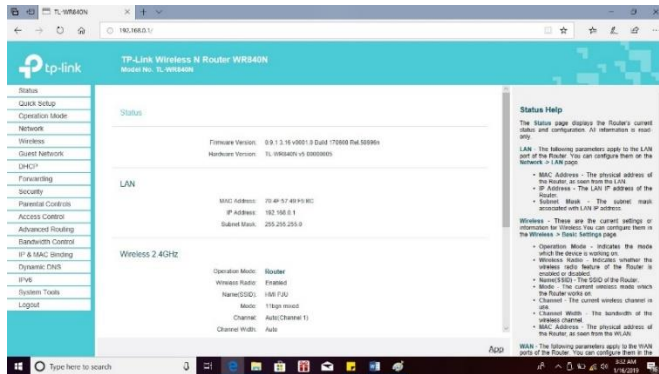
Tahapan awal untuk memulai komunikasi antara ethernet dan PC adalah dengan mengkoneksikan *router* dengan PC. *Router* dengan PC diberi *IP address* yang berbeda. Berikut merupakan pengujian koneksi *router* dengan PC.

Buka *website* TP-LINK untuk mengatur *IP address router* dengan mengisikan *username* “admin” dan *password* “admin”. Tampilan *website* TP-LINK dapat dilihat seperti Gambar 4.1.



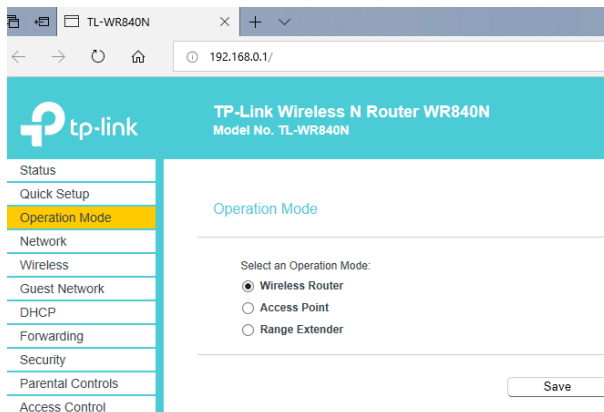
Gambar 4.1 Tampilan *Login Website* TP-LINK

Setelah *login*, maka akan muncul tampilan seperti pada Gambar 4.2. Tampilan yang muncul berupa informasi dan status dari *router* yang digunakan.



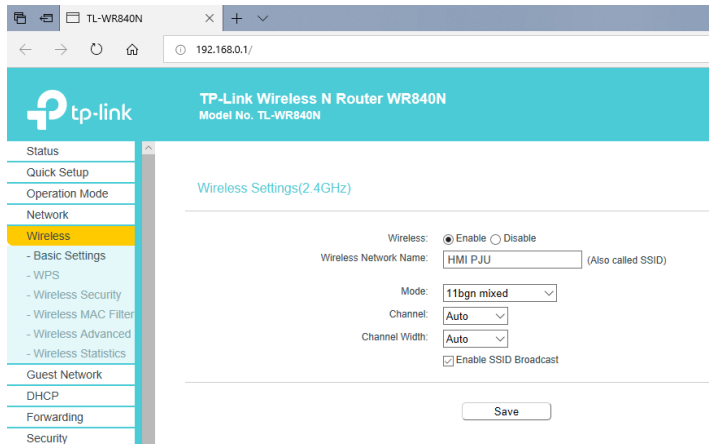
Gambar 4.2 Tampilan Menu Informasi dan Status *Router*

Untuk mengatur mode operasi *router*, pilih menu *operation mode*. Pada menu *operation mode*, terdapat tiga pilihan untuk mode pengoperasian *router* seperti yang ditampilkan pada Gambar 4.3. Pilih *Wireless Router* dan *Save*.



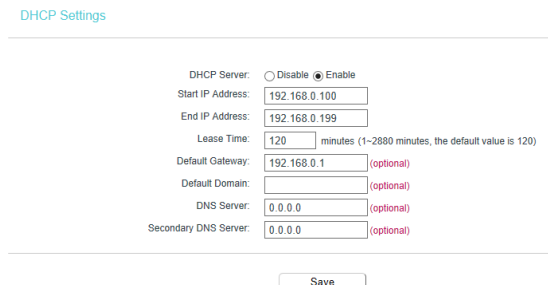
Gambar 4.3 Tampilan Menu *Operation Mode Router*

Untuk memudahkan mengganti nama jaringan agar lebih mudah dikenali, pilih menu **Wireless – Basic Settings**. Tuliskan “HMI PJU” sebagai nama jaringan pada kolom **Wireless Network Name** yang dapat dilihat seperti pada Gambar 4.4



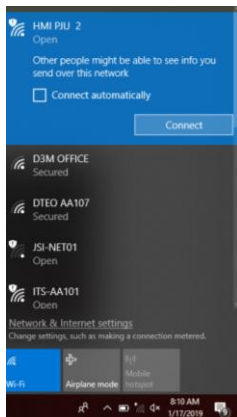
Gambar 4.4 Tampilan Menu *Wireless Setting*

Untuk mengatur IP *address* agar sesuai dengan yang diinginkan, pilih menu **DHCP** dan **DHCP settings**. Tuliskan IP *address* yang **192.168.0.1** pada kolom **Default Gateway**. IP *address* tersebut merupakan IP *address* dari *router*. Untuk lebih jelasnya, dapat dilihat pada Gambar 4.5.



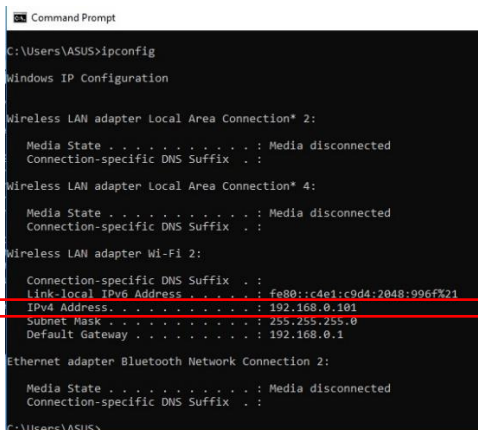
Gambar 4.5 Tampilan Menu *DHCP Settings*

Setelah dilakukan *setting* pada *router*, koneksikan PC dengan jaringan HMI PJU 2 seperti pada Gambar 4.6 .



Gambar 4.6 Koneksi dengan *Router*

Untuk mengetahui apakah PC dapat terhubung dengan *router*, lakukan ping IP PC terlebih dahulu untuk mengetahui IP *address* PC yang digunakan seperti pada Gambar 4.7 dan ping IP *router* untuk mengetahui apakah PC sudah terhubung dengan *router* seperti pada Gambar 4.8.



Gambar 4.7 Ping IP Address PC

```
Command Prompt
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time=2ms TTL=64
Reply from 192.168.0.1: bytes=32 time=15ms TTL=64
Reply from 192.168.0.1: bytes=32 time=2ms TTL=64
Reply from 192.168.0.1: bytes=32 time=5ms TTL=64

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 15ms, Average = 6ms

C:\Users\ASUS>
```

Gambar 4.8 Ping IP Address Router

Dari hasil pengujian koneksi *router* dengan PC, dapat disimpulkan bahwa saat PC dikoneksikan dengan *router* maka harus dilakukan pengecekan *ping* IP *router* melalui *command prompt*. Hal tersebut dilakukan karena saat PC dikoneksikan dengan *router*, terkadang mengalami *timed out* yang berarti koneksi PC dengan *router* terputus. Terputusnya koneksi dengan *router* dapat diakibatkan oleh kondisi jaringan yang tidak stabil.

4.2 Pengujian Tes Ping Network

Untuk mengetahui apakah PJU terkoneksi dengan jaringan atau tidak, maka diperlukan tes *ping*. Untuk melakukan pengujian tes ping, cukup klik *link* label **connection** pada HMI maka akan muncul *command prompt* yang otomatis meng-*ping* IP yang dituju pada masing-masing PJU. Berikut merupakan pengujian *ping network* yang dilakukan pada masing-masing PJU.

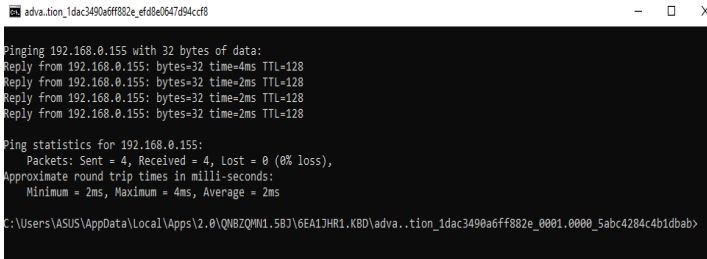
4.2.1 Pengujian Tes Ping PJU001

Pengujian tes *ping* PJU001 ditujukan kepada IP *address* ethernet 192.168.0.155. Adapun program *ping* pada VB dapat dilihat pada Gambar 4.9.

```
References
Private Sub LinkLabel5_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles LinkLabel5.LinkClicked
    Shell("cmd.exe /k" + "ping " + "192.168.0.155" + "-t", AppWinStyle.NormalFocus)
End Sub
```

Gambar 4.9 Program tes ping PJU001 pada VB

Saat PJU001 di *ping*, apabila berhasil terkoneksi maka akan menampilkan *command prompt* seperti pada Gambar 4.10 dan apabila tidak terkoneksi maka *command prompt* terlihat seperti pada Gambar 4.11.

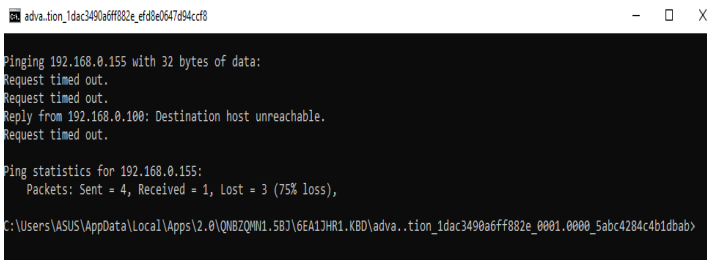


```
adva.tion_idac3490a6ff882e_efd8e0647d94cfc8
Pinging 192.168.0.155 with 32 bytes of data:
Reply from 192.168.0.155: bytes=32 time=4ms TTL=128
Reply from 192.168.0.155: bytes=32 time=2ms TTL=128
Reply from 192.168.0.155: bytes=32 time=2ms TTL=128
Reply from 192.168.0.155: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.0.155:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 4ms, Average = 2ms

C:\Users\ASUS\AppData\Local\Apps\2.0\QNBZQ\W1.5B3\6EA1JHR1.KBD\adva..tion_idac3490a6ff882e_0001.0000_5abc4284c4b1dbab>
```

Gambar 4.10 Program Tes *Ping* PJU001 Saat Berhasil Terkoneksi



```
adva.tion_idac3490a6ff882e_efd8e0647d94cfc8
Pinging 192.168.0.155 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 192.168.0.100: Destination host unreachable.
Request timed out.

Ping statistics for 192.168.0.155:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),

C:\Users\ASUS\AppData\Local\Apps\2.0\QNBZQ\W1.5B3\6EA1JHR1.KBD\adva..tion_idac3490a6ff882e_0001.0000_5abc4284c4b1dbab>
```

Gambar 4.11 Program Tes *Ping* PJU001 Saat Gagal Terkoneksi

Dari hasil pengujian tes ping PJU001, dapat disimpulkan bahwa saat PJU001 gagal terkoneksi, maka pada *command prompt* akan muncul “*request timed out*” yang berarti PJU001 terputus dengan jaringan, serta “*destination host unreachable*” yang berarti jaringan tidak dapat dijangkau, sehingga PJU001 tidak dapat mengirimkan data menuju *server*. Saat PJU001 gagal terkoneksi, didapatkan *loss* sebesar 75% karena saat *ping network*, PJU001 mengalami 3 kali *timed out*. Sedangkan saat PJU001 berhasil terkoneksi, maka persentase *loss* sebesar 0%, karena saat *ping network*, PJU001 tidak mengalami *timed out*.

4.2.2 Pengujian Tes *Ping* PJU002

Pengujian tes *ping* PJU002 ditujukan kepada IP *address* ethernet 192.168.0.156. Adapun program *ping* pada VB dapat dilihat pada Gambar 4.12.

```
References
Private Sub LinkLabel16_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles LinkLabel16.LinkClicked
    Shell("cmd.exe /k" + "ping " + "192.168.0.156" + "-t", AppWinStyle.NormalFocus)
End Sub
```

Gambar 4.12 Program tes ping PJU002 pada VB

Saat PJU002 di *ping*, apabila berhasil terkoneksi maka akan menampilkan *command prompt* seperti pada Gambar 4.13 dan apabila tidak terkoneksi maka *command prompt* terlihat seperti pada Gambar 4.14.

```
E:\HMI DEVI\AdvancedHMI\bin\Debug\AdvancedHMI.exe

Pinging 192.168.0.156 with 32 bytes of data:
Reply from 192.168.0.156: bytes=32 time=6ms TTL=128
Reply from 192.168.0.156: bytes=32 time=2ms TTL=128
Reply from 192.168.0.156: bytes=32 time=2ms TTL=128
Reply from 192.168.0.156: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.0.156:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 6ms, Average = 3ms

E:\HMI DEVI\AdvancedHMI\bin\Debug>
```

Gambar 4.13 Program Tes *Ping* PJU002 Saat Berhasil Terkoneksi

```
E:\HMI DEVI\AdvancedHMI\bin\Debug\AdvancedHMI.exe

Pinging 192.168.0.156 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Reply from 192.168.0.100: Destination host unreachable.

Ping statistics for 192.168.0.156:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),

E:\HMI DEVI\AdvancedHMI\bin\Debug>
```

Gambar 4. 14 Program Tes *Ping* PJU002 Saat Gagal Terkoneksi

Dari hasil pengujian tes ping PJU002, dapat disimpulkan bahwa saat PJU002 gagal terkoneksi, maka pada *command prompt* akan muncul “*request timed out*” yang berarti PJU002 terputus dengan jaringan, serta “*destination host unreachable*” yang berarti jaringan tidak dapat dijangkau, sehingga PJU002 tidak dapat mengirimkan data menuju *server*. Saat PJU002 gagal terkoneksi, didapatkan *loss* sebesar 75% karena saat *ping network*, PJU002 mengalami 3 kali *timed out*. Sedangkan saat PJU002 berhasil terkoneksi, maka persentase *loss* sebesar 0%, karena saat *ping network*, PJU002 tidak mengalami *timed out*.

4.2.3 Pengujian Tes Ping PJU003

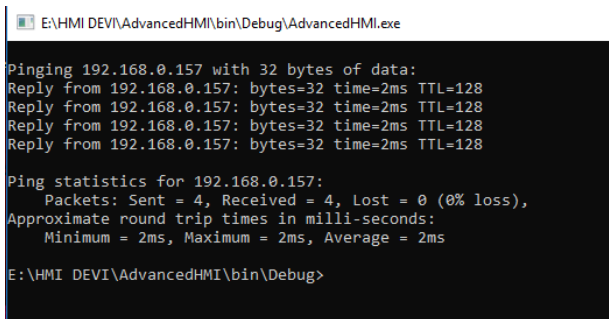
Pengujian tes *ping* PJU003 ditujukan kepada IP address ethernet 192.168.0.157. Adapun program *ping* pada VB dapat dilihat pada Gambar 4.15.



```
0 references
Private Sub LinkLabel7_LinkClicked(sender As Object, e As LinkLabel7_LinkClickedEventArgs) Handles LinkLabel7.LinkClicked
    Shell("cmd.exe /K" + "ping " + "192.168.0.157" + " -t", AppInStyle.NormalFocus)
End Sub
```

Gambar 4.15 Program Tes Ping PJU003 pada VB

Saat PJU003 di *ping*, apabila berhasil terkoneksi maka akan menampilkan *command prompt* seperti pada Gambar 4.16 dan apabila tidak terkoneksi maka *command prompt* terlihat seperti pada Gambar 4.17.



```
E:\HMI DEVI\AdvancedHMI\bin\Debug\AdvancedHMI.exe
Pinging 192.168.0.157 with 32 bytes of data:
Reply from 192.168.0.157: bytes=32 time=2ms TTL=128
Reply from 192.168.0.157: bytes=32 time=2ms TTL=128
Reply from 192.168.0.157: bytes=32 time=2ms TTL=128
Reply from 192.168.0.157: bytes=32 time=2ms TTL=128

Ping statistics for 192.168.0.157:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 2ms, Average = 2ms

E:\HMI DEVI\AdvancedHMI\bin\Debug>
```

Gambar 4.16 Program Tes Ping PJU003 Saat Berhasil Terkoneksi


```
E:\HMI DEVI\AdvancedHMI\bin\Debug\AdvancedHMI.exe

Pinging 192.168.0.157 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 192.168.0.100: Destination host unreachable.
Request timed out.

Ping statistics for 192.168.0.157:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),

E:\HMI DEVI\AdvancedHMI\bin\Debug>
```

Gambar 4.17 Program Tes *Ping* PJU003 Saat Gagal Terkoneksi

Dari hasil pengujian tes ping PJU003, dapat disimpulkan bahwa saat PJU003 gagal terkoneksi, maka pada *command prompt* akan muncul “*request timed out*” yang berarti PJU003 terputus dengan jaringan, serta “*destination host unreachable*” yang berarti jaringan tidak dapat dijangkau, sehingga PJU003 tidak dapat mengirimkan data menuju *server*. Saat PJU003 gagal terkoneksi, didapatkan *loss* sebesar 75% karena saat *ping network*, PJU003 mengalami 3 kali *timed out*. Sedangkan saat PJU003 berhasil terkoneksi, maka persentase *loss* sebesar 0%, karena saat *ping network*, PJU003 tidak mengalami *timed out*.

4.2.4 Pengujian Tes *Ping* PJU004

Pengujian tes *ping* PJU004 ditujukan kepada IP *address* ethernet 192.168.0.158. Adapun program *ping* pada VB dapat dilihat pada Gambar 4.18.

```
References
Private Sub LinkLabel8_LinkClicked(sender As Object, e As LinkLabelLinkClickedEventArgs) Handles LinkLabel8.LinkClicked
    Shell("cmd.exe /k " & "ping " & "192.168.0.158" & " -t", AppInStyle.NormalFocus)
End Sub
```

Gambar 4.18 Program Tes *Ping* PJU004 pada VB

Saat PJU004 di *ping*, apabila berhasil terkoneksi maka akan menampilkan *command prompt* seperti pada Gambar 4.19 dan apabila tidak terkoneksi maka *command prompt* terlihat seperti pada Gambar 4.20.

```
E:\HMI DEVI\AdvancedHMI\bin\Debug\AdvancedHMI.exe

Pinging 192.168.0.158 with 32 bytes of data:
Request timed out.
Reply from 192.168.0.158: bytes=32 time=2ms TTL=128
Request timed out.
Request timed out.

Ping statistics for 192.168.0.158:
    Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 2ms, Average = 2ms

E:\HMI DEVI\AdvancedHMI\bin\Debug>
```

Gambar 4.19 Program Tes *Ping* PJU004 Saat Berhasil Terkoneksi

```
E:\HMI DEVI\AdvancedHMI\bin\Debug\AdvancedHMI.exe

Pinging 192.168.0.158 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.158:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

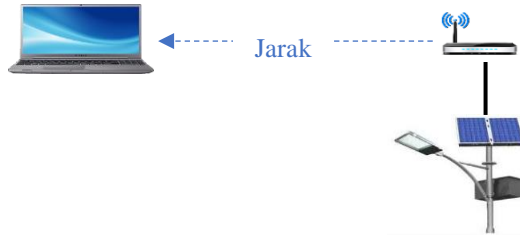
E:\HMI DEVI\AdvancedHMI\bin\Debug>
```

Gambar 4.20 Program Tes *Ping* PJU004 Saat Gagal Terkoneksi

Dari hasil pengujian tes ping PJU004, dapat disimpulkan bahwa saat PJU004 gagal terkoneksi, maka pada *command prompt* akan muncul “*request timed out*” yang berarti PJU004 terputus dengan jaringan, sehingga PJU004 tidak dapat mengirimkan data menuju *server*. Saat PJU004 gagal terkoneksi, didapatkan *loss* sebesar 100% karena saat *ping network*, PJU004 mengalami 4 kali *timed out*. Sedangkan saat PJU004 berhasil terkoneksi, maka persentase *loss* sebesar 75%, karena saat *ping network*, PJU004 mengalami *timed out* sebanyak 3 kali.

4.3 Pengujian Jarak Koneksi *Ethernet*

Pengujian jarak koneksi *Ethernet* bertujuan untuk mengetahui jangkauan komunikasi dari modul *Ethernet*, sehingga dapat memaksimalkan penggunaannya dalam komunikasi. Pengujian jarak koneksi *Ethernet* seperti pada Gambar 4.21.



Gambar 4.21 Pengujian Jarak Koneksi *Ethernet*

Untuk mencari persentase nilai *error* jarak koneksi *Ethernet*, berikut merupakan rumus perhitungan yang digunakan.

$$Error = \frac{\text{Data Terima}}{\text{Data Terkirim}} \times 100\%$$

Untuk mencari persentase nilai rata-rata *error* jarak koneksi *Ethernet* dalam satu jarak pengujian adalah sebagai berikut.

$$Error \text{ rata - rata} = \frac{\text{Data1} + \dots + \text{Data ke } n}{\text{Jumlah Data } n}$$

4.3.1 Pengujian Jarak 10 Meter

Pengujian jarak koneksi *Ethernet* sejauh 10 meter didapatkan hasil seperti pada Tabel 4.1. Dari hasil pengujian, *error* yang terjadi diperoleh dari hasil perbandingan data yang terkirim dan data terima. Adapun data yang dikirim dan diterima dalam pengujian jarak *Ethernet* ini merupakan data analog. Persentase kesalahan pembacaan pada jarak 10 meter sebanyak 0,0403226%.

Tabel 4.1 Pengukuran Jarak 10 Meter

No	Data terkirim	Data diterima	<i>Error</i> (%)
1	1	1	0
2	144	144	0
3	248	247	0,403226
4	300	300	0

No	Data terkirim	Data diterima	Error (%)
5	484	484	0
6	649	649	0
7	885	885	0
8	930	930	0
9	977	977	0
10	1019	1019	0
Persentase kesalahan jarak 10 meter (%)			0,0403226

Dari Tabel 4.1 dapat disimpulkan, bahwa pada jarak 10 meter dari 10 kali pengujian didapatkan *error* sebesar 0,0403226%. *Error* tersebut disebabkan oleh jaringan yang tidak stabil dan kualitas kabel LAN yang kurang baik, sehingga terdapat *error* saat pengiriman data.

4.3.2 Pengujian Jarak 20 Meter

Pengujian jarak koneksi *Ethernet* sejauh 20 meter didapatkan hasil seperti pada Tabel 4.2. Dari hasil pengujian, *error* yang terjadi diperoleh dari hasil perbandingan data yang terkirim dan data terima. Adapun data yang dikirim dan diterima dalam pengujian jarak *Ethernet* ini merupakan data analog. Persentase kesalahan pembacaan pada jarak 20 meter sebanyak 0,1232362%.

Tabel 4.2 Pengukuran Jarak 20 Meter

No	Data terkirim	Data diterima	Error (%)
1	11	11	0
2	61	61	0
3	137	136	0,729927
4	192	192	0
5	281	281	0
6	311	310	0,3215434
7	646	646	0
8	777	776	0,1287001

No	Data terkirim	Data diterima	Error (%)
9	919	919	0
10	1010	101	0
Persentase kesalahan jarak 20 meter (%)			0,1232362

Dari Tabel 4.2 dapat disimpulkan, bahwa pada jarak 20 meter dari 10 kali pengujian didapatkan *error* sebesar 0,1232362. *Error* tersebut disebabkan oleh jaringan yang tidak stabil dan kualitas kabel LAN yang kurang baik, sehingga terdapat *error* saat pengiriman data.

4.3.3 Pengujian Jarak 30 Meter

Pengujian jarak koneksi *Ethernet* sejauh 30 meter didapatkan hasil seperti pada Tabel 4.3. Dari hasil pengujian, *error* yang terjadi diperoleh dari hasil perbandingan data yang terkirim dan data terima. Adapun data yang dikirim dan diterima dalam pengujian jarak *Ethernet* ini merupakan data analog. Persentase kesalahan pembacaan pada jarak 30 meter sebanyak 0,01055966%.

Tabel 4.3 Pengukuran Jarak 30 Meter

No	Data terkirim	Data diterima	Error (%)
1	34	34	0
2	138	138	0
3	321	321	0
4	420	420	0
5	505	505	0
6	624	624	0
7	718	718	0
8	867	867	0
9	947	946	0,1055966
10	988	988	0
Persentase kesalahan jarak 30 meter (%)			0,01055966

Dari Tabel 4.3 dapat disimpulkan, bahwa pada jarak 30 meter dari 10 kali pengujian didapatkan *error* sebesar 0,01055966%. *Error* tersebut disebabkan oleh jaringan yang tidak stabil dan kualitas kabel LAN yang kurang baik, sehingga terdapat *error* saat pengiriman data.

4.3.4 Pengujian Jarak 40 Meter

Pengujian jarak koneksi *Ethernet* sejauh 40 meter didapatkan hasil seperti pada Tabel 4.4. Dari hasil pengujian, *error* yang terjadi diperoleh dari hasil perbandingan data yang terkirim dan data terima. Adapun data yang dikirim dan diterima dalam pengujian jarak *Ethernet* ini merupakan data analog. Persentase kesalahan pembacaan pada jarak 40 meter sebanyak 0,0282485%.

Tabel 4.4 Pengukuran Jarak 40 Meter

No	Data terkirim	Data diterima	<i>Error</i> (%)
1	44	44	0
2	94	94	0
3	133	133	0
4	222	222	0
5	267	267	0
6	289	289	0
7	3311	33	0
8	354	353	0,282485
9	372	372	0
10	401	401	0
Persentase kesalahan jarak 40 meter (%)			0,0282485

Dari Tabel 4.4 dapat disimpulkan, bahwa pada jarak 40 meter dari 10 kali pengujian didapatkan *error* sebesar 0,0282485%. *Error* tersebut disebabkan oleh jaringan yang tidak stabil dan kualitas kabel LAN yang kurang baik, sehingga terdapat *error* saat pengiriman data.

Dari hasil pengujian jarak 10 hingga 40 meter, untuk mengetahui persentase *error* selama pengujian, maka dapat dihitung dengan rumus sebagai berikut.

$$\begin{aligned}
 \text{Error} &= \frac{\text{Data 10 m} + \text{Data 20 m} + \text{Data 30 m} + \text{Data 40 m}}{4} \\
 &= \frac{0,0403226 + 0,1232362 + 0,01055966 + 0,0282485}{4} \\
 &= \frac{0,20236696}{4} \\
 &= 0,05059174 \%
 \end{aligned}$$

Persentase *error* dalam jarak pengujian 10-40 meter adalah sebesar 0,05%. Dan persentase keberhasilan pengiriman datanya adalah $100\% - 0,05\% = 99,95\%$.

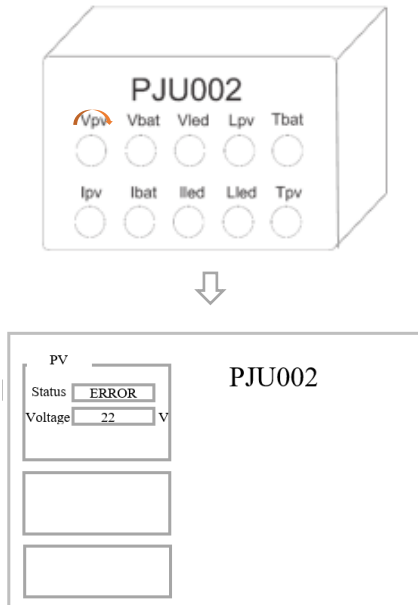
4.4 Pengujian Tampilan Status Normal/Error

Pengujian tampilan status normal/*error* dilakukan untuk mengetahui apakah fungsi dari diagnosa status normal/*error* PJU sudah berfungsi dengan baik atau belum. Berikut merupakan hasil pengujian status normal/*error* seperti pada Gambar 4.22.



Gambar 4.22 Tampilan Status Normal/Error pada Main Menu

Pengujian status normal/*error* bertujuan untuk mengetahui apakah program HMI sudah sesuai dengan fungsinya. Pengujian status normal/*error* seperti pada Gambar 4.23.



Gambar 4.23 Pengujian Status Normal/*Error*

Pengujian yang dilakukan menggunakan modul *dummy* adalah PJU002, PJU003, dan PJU003. Pengujian dilakukan dengan beberapa kondisi, yaitu saat kondisi normal dan *error*. Kondisi saat *error* terbagi menjadi beberapa penyebab, yaitu saat PV rusak, saat baterai rusak, dan saat LED rusak.

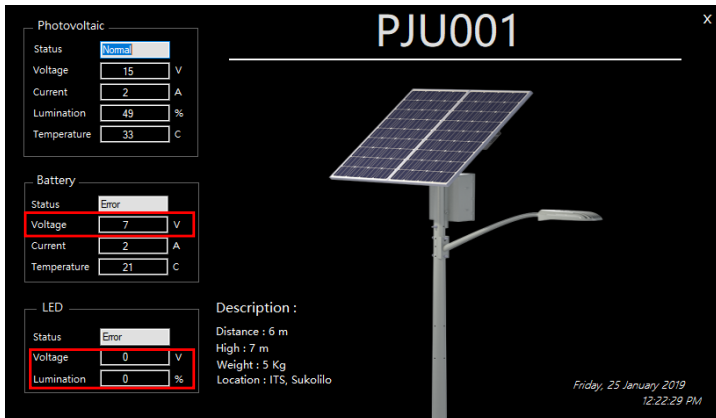
4.4.1 Pengujian Status Normal/*Error* Pada Komponen PJU001

Pengujian status normal/*error* pada PJU001 pada beberapa kondisi, adapun hasil pengujian pada PJU001 dapat dilihat pada Gambar 4.24 dan Gambar 4.25.

Pada Gambar 4.24 dapat dilihat bahwa PJU001 mengalami *error* atau kerusakan. Untuk mengetahui komponen yang rusak, klik details, maka akan muncul tampilan seperti pada Gambar 4.25.



Gambar 4.24 Pengujian Status PJU001 pada *Main Menu*



Gambar 4.25 Status Normal PJU001 pada *Form Details*

Pada Gambar 4.25, PJU001 mengalami *error* disebabkan baterai dan LED yang memiliki nilai di bawah ketentuan komponen normal. Adapun kerusakan pada PJU001 disebabkan oleh hal sebagai berikut.

Kondisi PV (Normal)

Kondisi PV normal, karena nilai tegangan yang terbaca pada HMI sebesar 15 V, dan arus sebesar 2 A. Nilai tersebut memenuhi kondisi PV aman, karena tegangan berada di bawah nilai 21,6 V, dan nilai arus berada di bawah 2,8 A.

Kondisi Baterai (Error)

Kondisi baterai pada PJU001 dikatakan rusak karena tegangan baterai yang terukur sebesar 7V, dimana tegangan tersebut memenuhi kondisi baterai *deep charging*.

Kondisi LED (Error)

Kondisi LED pada PJU001 dikatakan rusak karena nilai tegangan yang terukur sebesar 0 V dan luminasi sebesar 0% yang berarti LED tidak dapat digunakan atau rusak.

Dari hasil pengujian pada komponen PJU001, dapat disimpulkan bahwa penampilan status normal/error sudah dapat berjalan dengan benar sesuai dengan nilai yang diterima karena selama pengujian tidak didapatkan kesalahan pembacaan nilai pada program.

4.4.2 Pengujian Status Normal/Error Pada Komponen PJU002

Pengujian status normal/error pada PJU002 pada beberapa kondisi, adapun hasil pengujian pada PJU002 adalah sebagai berikut.

Kondisi Normal

Apabila PV, baterai, dan LED dalam kondisi baik, maka menu utama HMI PJU002 akan menampilkan “Normal” sebagai indikator bahwa PJU002 dalam keadaan baik, sehingga tidak diperlukan pemeriksaan dan penggantian komponen. Tampilan indikator PJU002 dalam keadaan “Normal” dapat dilihat pada Gambar 4.26.



Gambar 4.26 Status Normal PJU002 pada *Main Menu*

Untuk mengetahui lebih detail masing-masing performa komponen PJU002, maka dapat dilihat pada Gambar 4.27. Kondisi PJU002 dapat diindikasikan “Normal” dikarenakan hal berikut:

Kondisi PV (Normal)

- Tegangan : Tegangan PV yang terbaca pada HMI sebesar 20 V, dimana tegangan tersebut masih berada di bawah 21,6 V yang merupakan batas maksimum PV dalam keadaan *short circuit*.
- Arus : Arus PV yang terbaca pada HMI sebesar 1 A, dimana arus tersebut masih berada di bawah 2,8 A yang merupakan batas maksimum PV dalam keadaan *open circuit*.

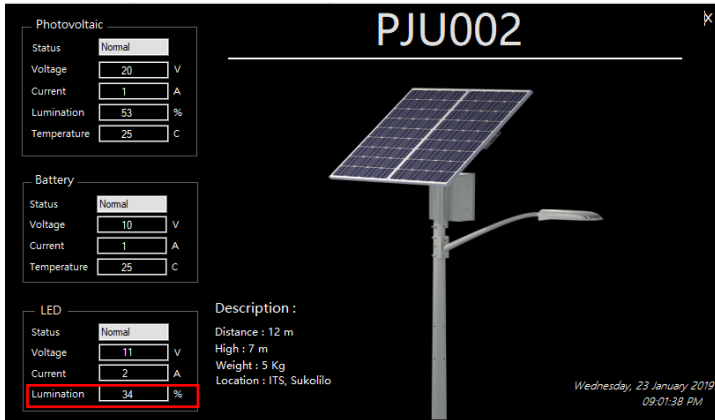
Kondisi Baterai (Normal)

- Tegangan : Tegangan baterai yang terbaca pada HMI sebesar 10 V, dimana tegangan tersebut masih berada di bawah 15 V yang merupakan batas kondisi baterai dalam keadaan *over charging* dan di atas 8,5 V yang merupakan batas kondisi baterai dalam keadaan *deep charging*.
- Temperatur : Temperatur baterai yang terbaca pada HMI sebesar 25°C, dimana temperatur tersebut masih berada di

bawah 28°C yang merupakan batas kondisi baterai *over-heating*.

Kondisi LED (Normal)

- Luminasi : Luminasi LED yang terbaca pada HMI sebesar 34%, dimana nilai tersebut masih berada di bawah 60% yang merupakan batas kondisi luminasi LED normal.



Gambar 4.27 Status Normal PJU002 pada *Form Details*

Kondisi *Error*

Apabila PV, baterai, dan LED dalam kondisi rusak, maka menu utama HMI PJU002 akan menampilkan “*Error*” sebagai indikator bahwa PJU002 dalam keadaan rusak, sehingga diperlukan pemeriksaan dan penggantian komponen. Tampilan indikator PJU002 dalam keadaan “*Error*” dapat dilihat pada Gambar 4.28.

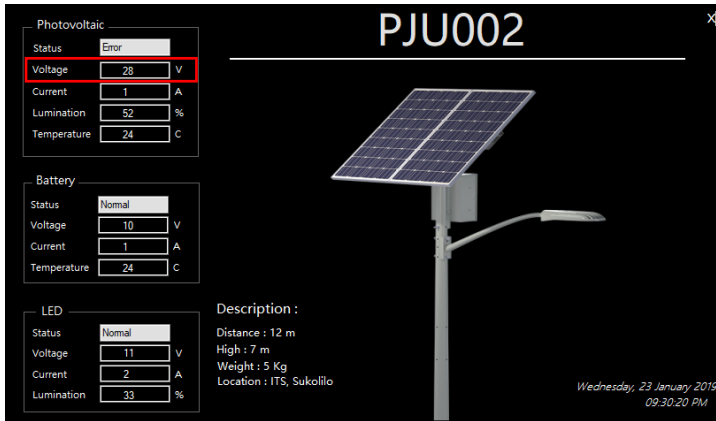


Gambar 4.28 Status *Error* PJU002 pada *Main Menu*

Untuk mengetahui lebih detail komponen mana yang mengalami kerusakan pada PJU002, berikut merupakan pengujian PJU002 yang mengalami kerusakan karena beberapa faktor.

Kondisi PV (*Error*)

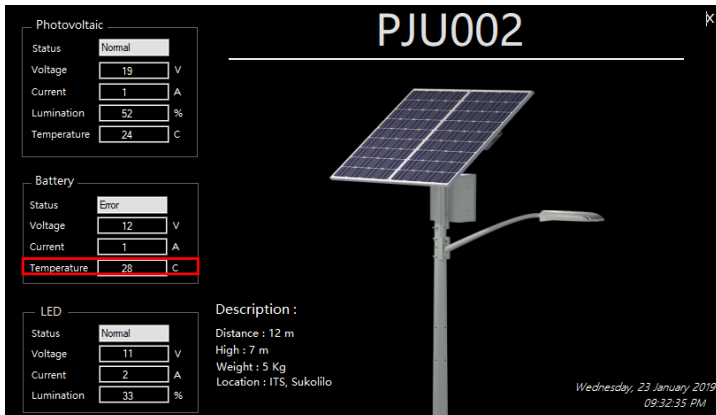
Kondisi PV dikatakan rusak apabila salah satu nilai indikator tidak terpenuhi, seperti yang terlihat pada Gambar 4.29. Adapun kerusakan pada PV disebabkan oleh tegangan yang terbaca pada HMI sebesar 28 V, dimana tegangan tersebut berada di atas 21,6 V yang merupakan batas maksimum PV dalam keadaan *short circuit*.



Gambar 4.29 Status PV PJU002 Saat *Error*

Kondisi Baterai (*Error*)

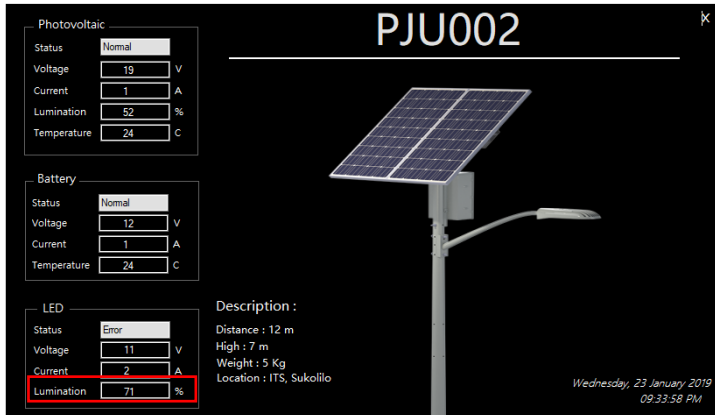
Kondisi baterai dikatakan rusak apabila salah satu nilai indikator tidak terpenuhi, seperti yang terlihat pada Gambar 4.30. Adapun kerusakan pada baterai disebabkan oleh temperatur yang terbaca pada HMI sebesar 28°C, dimana temperatur tersebut merupakan kondisi *overheating* pada baterai.



Gambar 4.30 Status Baterai PJU002 Saat *Error*

Kondisi LED (*Error*)

Kondisi LED dikatakan rusak apabila salah satu nilai indikator tidak terpenuhi, seperti yang terlihat pada Gambar 4.31. Adapun kerusakan pada LED disebabkan oleh luminasi yang terbaca pada HMI sebesar 71%, dimana nilai tersebut melampaui batas normal LED.



Gambar 4.31 Status LED PJU002 Saat *Error*

Dari hasil pengujian pada komponen PJU002, dapat disimpulkan bahwa penampilan status normal/*error* sudah dapat berjalan dengan benar sesuai dengan nilai yang diterima karena selama pengujian tidak didapatkan kesalahan pembacaan nilai pada program.

4.4.3 Pengujian Status Normal/*Error* Pada Komponen PJU003

Pengujian status normal/*error* pada PJU003 pada beberapa kondisi, adapun hasil pengujian pada PJU003 adalah sebagai berikut.

Kondisi Normal

Apabila PV, baterai, dan LED dalam kondisi baik, maka menu utama HMI PJU003 akan menampilkan “Normal” sebagai indikator bahwa PJU003 dalam keadaan baik, sehingga tidak diperlukan pemeriksaan dan penggantian komponen. Tampilan

indikator PJU003 dalam keadaan “Normal” dapat dilihat pada Gambar 4.32.



Gambar 4.32 Status Normal PJU003 pada *Main Menu*

Untuk mengetahui lebih detail masing-masing performa komponen PJU003, maka dapat dilihat pada Gambar 4.33. Kondisi PJU002 dapat diindikasikan “Normal” dikarenakan hal berikut:

Kondisi PV (Normal)

- Tegangan : Tegangan PV yang terbaca pada HMI sebesar 17 V, dimana tegangan tersebut masih berada di bawah 21,6 V yang merupakan batas maksimum PV dalam keadaan *short circuit*.
- Arus : Arus PV yang terbaca pada HMI sebesar 2 A, dimana arus tersebut masih berada di bawah 2,8 A yang merupakan batas maksimum PV dalam keadaan *open circuit*.

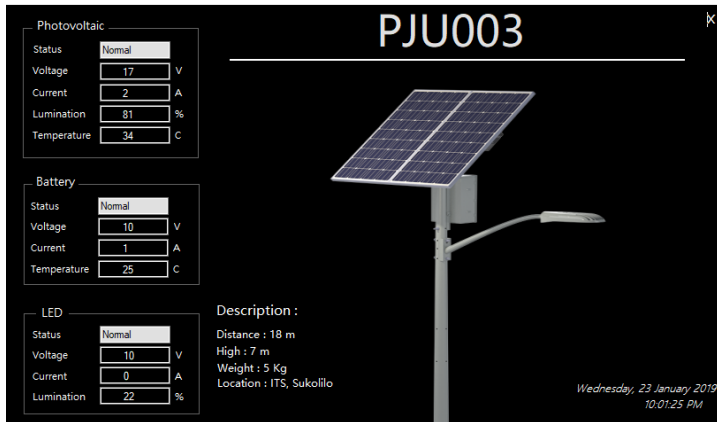
Kondisi Baterai (Normal)

- Tegangan : Tegangan baterai yang terbaca pada HMI sebesar 10 V, dimana tegangan tersebut masih berada di bawah 15 V yang merupakan batas kondisi baterai dalam keadaan *over charging* dan di atas 8,5 V yang merupakan batas kondisi baterai dalam keadaan *deep charging*.

- Temperatur : Temperatur baterai yang terbaca pada HMI sebesar 25°C, dimana temperatur tersebut masih berada di bawah 28°C yang merupakan batas kondisi baterai *overheating*.

Kondisi LED (Normal)

- Luminasi : Luminasi LED yang terbaca pada HMI sebesar 22%, dimana nilai tersebut masih berada di bawah 60% yang merupakan batas kondisi luminasi LED normal.



Gambar 4.33 Status Normal PJU003 pada *Form Details*

Kondisi Error

Apabila PV, baterai, dan LED dalam kondisi rusak, maka menu utama HMI PJU003 akan menampilkan “Error” sebagai indikator bahwa PJU003 dalam keadaan rusak, sehingga diperlukan pemeriksaan dan penggantian komponen. Tampilan indikator PJU003 dalam keadaan “Error” dapat dilihat pada Gambar 4.34.

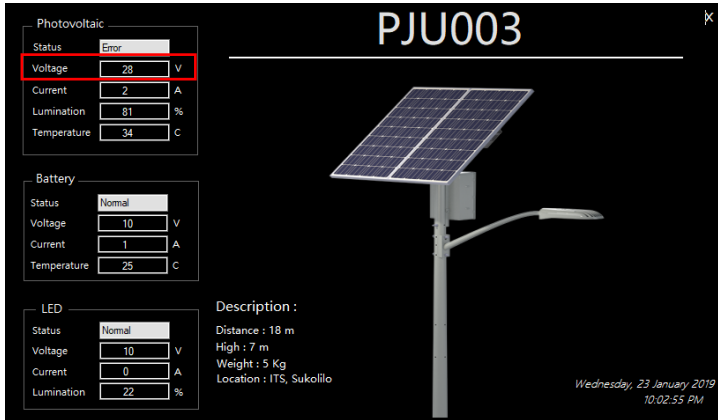


Gambar 4.34 Status *Error* PJU003 pada *Main Menu*

Untuk mengetahui lebih detail komponen mana yang mengalami kerusakan pada PJU003, berikut merupakan pengujian PJU003 yang mengalami kerusakan karena beberapa faktor.

Kondisi PV (*Error*)

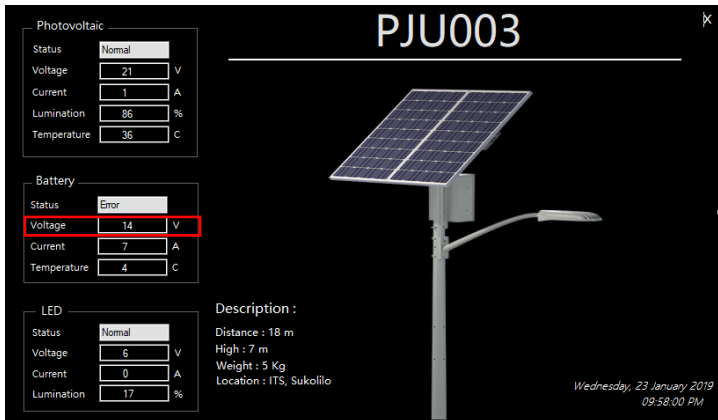
Kondisi PV dikatakan rusak apabila salah satu nilai indikator tidak terpenuhi, seperti yang terlihat pada Gambar 4.35. Adapun kerusakan pada PV disebabkan oleh tegangan yang terbaca pada HMI sebesar 28 V, dimana tegangan tersebut berada di atas 21,6 V yang merupakan batas maksimum PV dalam keadaan *short circuit*.



Gambar 4.35 Status PV PJU003 Saat *Error*

Kondisi Baterai (*Error*)

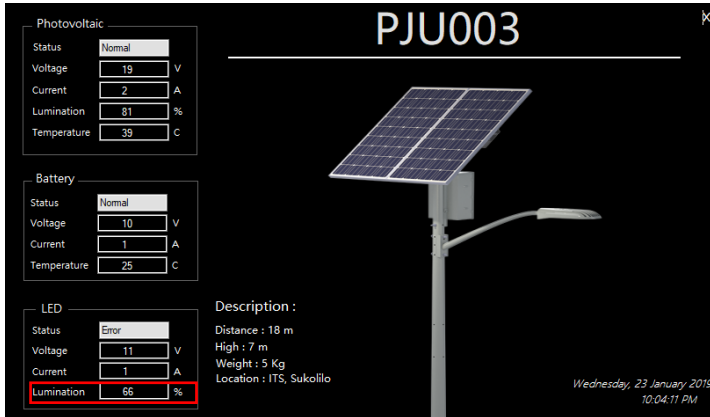
Kondisi baterai dikatakan rusak apabila salah satu nilai indikator tidak terpenuhi, seperti yang terlihat pada Gambar 4.36. Adapun kerusakan pada baterai disebabkan oleh tegangan yang terbaca pada HMI sebesar 14 V, dimana tegangan tersebut merupakan batas kondisi baterai dalam keadaan *over charging*.



Gambar 4.36 Status Baterai PJU003 Saat *Error*

Kondisi LED (*Error*)

Kondisi LED dikatakan rusak apabila salah satu nilai indikator tidak terpenuhi, seperti yang terlihat pada Gambar 4.37. Adapun kerusakan pada LED disebabkan oleh luminasi yang terbaca pada HMI sebesar 66%, dimana nilai tersebut melampaui batas normal LED.



Gambar 4.37 Status LED PJU003 Saat *Error*

Dari hasil pengujian pada komponen PJU003, dapat disimpulkan bahwa penampilan status normal/*error* sudah dapat berjalan dengan benar sesuai dengan nilai yang diterima karena selama pengujian tidak didapatkan kesalahan pembacaan nilai pada program.

4.4.4 Pengujian Status Normal/*Error* Pada Komponen PJU004

Pengujian status normal/*error* pada PJU004 pada beberapa kondisi, adapun hasil pengujian pada PJU004 adalah sebagai berikut.

Kondisi Normal

Apabila PV, baterai, dan LED dalam kondisi baik, maka menu utama HMI PJU004 akan menampilkan “Normal” sebagai indikator bahwa PJU004 dalam keadaan baik, sehingga tidak diperlukan pemeriksaan dan penggantian komponen. Tampilan

indikator PJU004 dalam keadaan “Normal” dapat dilihat pada Gambar 4.38.



Gambar 4.38 Status Normal PJU004 pada *Main Menu*

Untuk mengetahui lebih detail masing-masing performa komponen PJU004, maka dapat dilihat pada Gambar 4.39. Kondisi PJU004 dapat diindikasikan “Normal” dikarenakan hal berikut:

Kondisi PV (Normal)

- Tegangan : Tegangan PV yang terbaca pada HMI sebesar 20 V, dimana tegangan tersebut masih berada di bawah 21,6 V yang merupakan batas maksimum PV dalam keadaan *short circuit*.
- Arus : Arus PV yang terbaca pada HMI sebesar 1 A, dimana arus tersebut masih berada di bawah 2,8 A yang merupakan batas maksimum PV dalam keadaan *open circuit*.

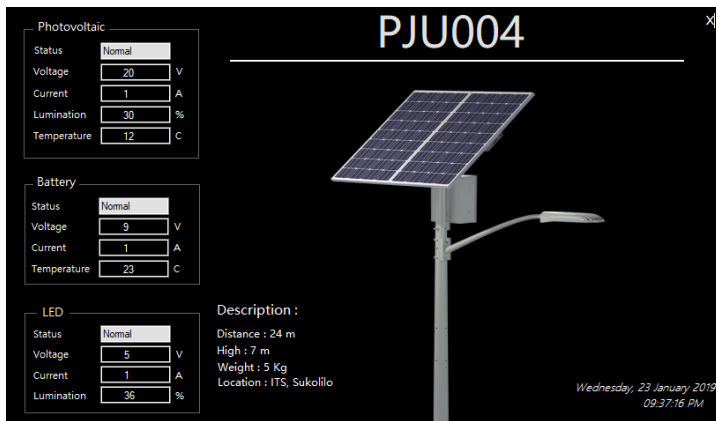
Kondisi Baterai (Normal)

- Tegangan : Tegangan baterai yang terbaca pada HMI sebesar 9 V, dimana tegangan tersebut masih berada di bawah 15 V yang merupakan batas kondisi baterai dalam keadaan *over charging* dan di atas 8,5 V yang merupakan batas kondisi baterai dalam keadaan *deep charging*.

- Temperatur : Temperatur baterai yang terbaca pada HMI sebesar 23°C, dimana temperatur tersebut masih berada di bawah 28°C yang merupakan batas kondisi baterai *overheating*.

Kondisi LED (Normal)

- Luminasi : Luminasi LED yang terbaca pada HMI sebesar 36%, dimana nilai tersebut masih berada di bawah 60% yang merupakan batas kondisi luminasi LED normal.



Gambar 4.39 Status Normal PJU004 pada *Form Details*

Kondisi Error

Apabila PV, baterai, dan LED dalam kondisi rusak, maka menu utama HMI PJU004 akan menampilkan “Error” sebagai indikator bahwa PJU004 dalam keadaan rusak, sehingga diperlukan pemeriksaan dan penggantian komponen. Tampilan indikator PJU004 dalam keadaan “Error” dapat dilihat pada Gambar 4.40.

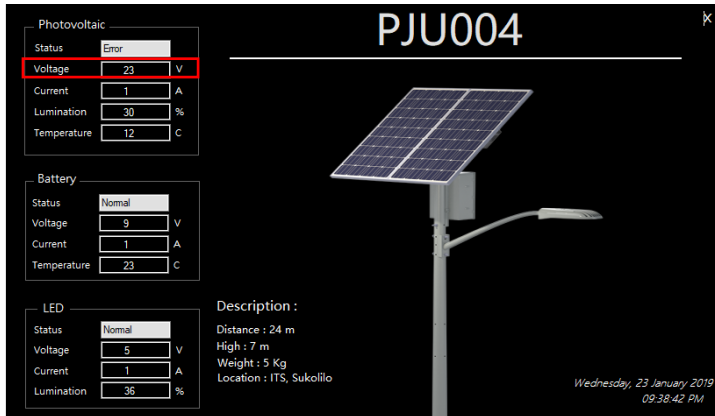


Gambar 4.40 Status *Error* PJU004 pada *Main Menu*

Untuk mengetahui lebih detail komponen mana yang mengalami kerusakan pada PJU004, berikut merupakan pengujian PJU004 yang mengalami kerusakan karena beberapa faktor.

Kondisi PV (*Error*)

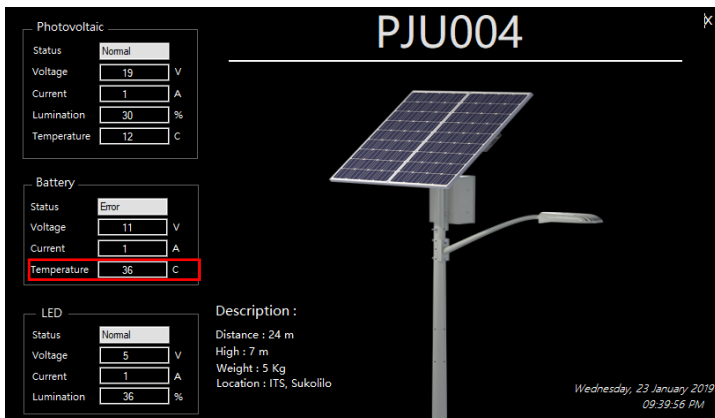
Kondisi PV dikatakan rusak apabila salah satu nilai indikator tidak terpenuhi, seperti yang terlihat pada Gambar 4.41. Adapun kerusakan pada PV disebabkan oleh tegangan yang terbaca pada HMI sebesar 23 V, dimana tegangan tersebut berada di atas 21,6 V yang merupakan batas maksimum PV dalam keadaan *short circuit*.



Gambar 4.41 Status PV PJU004 Saat *Error*

Kondisi Baterai (*Error*)

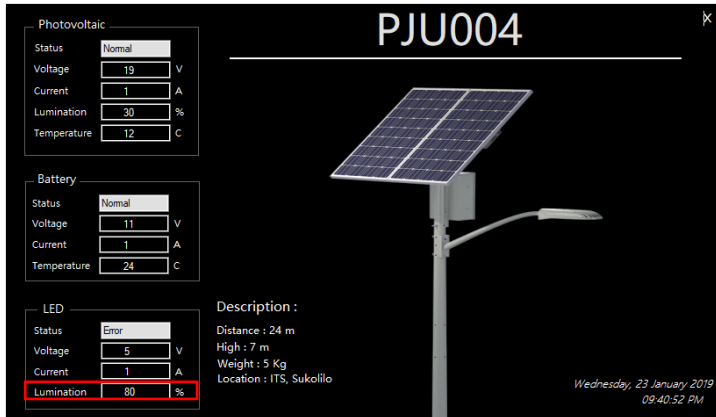
- Kondisi baterai dikatakan rusak apabila salah satu nilai indikator tidak terpenuhi, seperti yang terlihat pada Gambar 4.42. Adapun kerusakan pada baterai disebabkan oleh temperatur yang terbaca pada HMI sebesar 36°C, dimana tegangan tersebut merupakan batas kondisi baterai dalam keadaan *overheating*.



Gambar 4.42 Status Baterai PJU004 Saat *Error*

Kondisi LED (*Error*)

Kondisi LED dikatakan rusak apabila salah satu nilai indikator tidak terpenuhi, seperti yang terlihat pada Gambar 4.43. Adapun kerusakan pada LED disebabkan oleh luminasi yang terbaca pada HMI sebesar 80%, dimana nilai tersebut melampaui batas normal LED.



Gambar 4.43 Status LED PJU004 Saat *Error*

Dari hasil pengujian pada komponen PJU004, dapat disimpulkan bahwa penampilan status normal/*error* sudah dapat berjalan dengan benar sesuai dengan nilai yang diterima karena selama pengujian tidak didapatkan kesalahan pembacaan nilai pada program.

Dari keseluruhan hasil pengujian PJU001-PJU004 dapat disimpulkan bahwa program status normal/*error* PJU berfungsi dengan benar, karena saat pengujian tidak didapatkan kesalahan dalam mengidentifikasi kondisi komponen normal atau *error*. Adapun PV bernilai *error* saat tegangannya bernilai di atas 21,6 V dan arusnya di atas 2,8 A. Baterai bernilai *error* saat tegangannya berada di bawah 8,5 V dan di atas 15 V, dan temperaturnya di atas 28°C. LED bernilai *error* saat luminasinya diatas 60%.

4.5 Keberhasilan dan Efektifitas HMI

Setelah dilakukan pembuatan alat dan pengujian, diperlukan penilaian berupa persentase keberhasilan HMI dan efektifitas HMI. Penilaian tersebut bertujuan untuk mengetahui kelemahan HMI yang dibuat dan mengetahui apakah HMI yang dibuat sudah sesuai dengan fungsi awalnya. Berikut merupakan penjabaran dari keberhasilan dan efektifitas HMI yang sudah dibuat.

4.5.1 Keberhasilan HMI

Setelah dilakukan pengujian dan analisa data, penting untuk diketahui tingkat keberhasilan dan kesalahan dari alat yang dibuat, agar kedepannya dapat ditingkatkan lagi kinerjanya.

Adapun tingkat keberhasilan dan kesalahan pada HMI yang telah dibuat dijabarkan pada Tabel 4.5.

Tabel 4.5 Keberhasilan HMI

Keterangan	Jumlah Pengujian	Pengujian		Persentase	
		Keberhasilan	Kesalahan	Keberhasilan	Kesalahan
Pengiriman data jarak 10 M	10	9	1	90%	10%
Pengiriman data jarak 20 M	10	7	3	70%	30%
Pengiriman data jarak 30 M	10	9	1	90%	10%
Pengiriman data jarak 40 M	10	9	1	90%	10%
Program identifikasi status Normal/Error	16	16	0	100%	0%
Total				440%	60%
Rata-rata				88%	12%

Dari Tabel 4.5 dapat diketahui bahwa persentase kesalahan pengiriman data modul Ethernet shield sebesar 88%. Kesalahan ter-

sebut disebabkan oleh kualitas jaringan yang tidak stabil dan kualitas kabel LAN yang digunakan untuk menghubungkan Ethernet dengan *router*. Sedangkan untuk program identifikasi status normal/error pada Microsoft Visual Studio 2017 tidak terdapat kesalahan.

4.5.2 Efektifitas HMI

Untuk mengetahui apakah HMI yang dibuat sudah sesuai dengan tujuan, maka diperlukan adanya penilaian dari beberapa orang sebagai *sample*. Dengan adanya *survey* tersebut, dapat diperoleh saran-saran membangun yang dapat dijadikan sebagai acuan untuk memperbaiki HMI.

Survey yang dilakukan melibatkan 10 orang dengan 3 penilaian, antara lain penilaian desain HMI, aksestabilitas atau kemudahan pengoperasian, dan informatif. Penilaian yang dilakukan menggunakan penilaian dengan skala 0-10. Adapun hasil *survey* HMI dapat dilihat pada Tabel 4.6.

Tabel 4.6 Efektifitas HMI

No	Nama	Penilaian			Saran
		Desain	Aksestabilitas	Informatif	
1	Fajar	8,99	7,90	7,5	Utamakan kinerja HMI, kecepatan akses dan informasi yang lebih lengkap
2	Alifah	8,8	7,5	9,1	Kembangkan sistem komunikasi agar tidak putus nyambung
3	Dwi	9	9	9	<i>Font</i> lebih diperbesar lagi
4	Fatih	7	9	9	Perbaiki desain
5	Wisnu	9	8,7	8	Tambah fitur lagi

No	Nama	Penilaian			Saran
		Desain	Aksestabilitas	Informatif	
6	Amel	9	8	8	Perbaiki sistem komunikasi agar tidak putus nyambung
7	Sofyan	9	8,5	8	Lebih informatif
8	Sadudin	8	8	7,5	Tambahkan fitur yang lebih banyak
9	Abiy	8	9	9	Perbarui desain agar lebih menarik
10	Qiqin	8	8	9	Perbaiki lagi desain
Nilai rata-rata		8,479	8,36	8,41	

Dari hasil *survey*, didapatkan penilaian untuk desain sebesar 8,479, aksestabilitas sebesar 8,36, dan informatif sebesar 8,41. Saran yang didapatkan antara lain desain yang lebih diperbaiki, menambah informasi yang ditampilkan, serta perbaikan terhadap sistem komunikasi.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pengujian dan analisa perancangan dan implementasi *Human Machine Interface* (HMI) pada Sistem *Telemonitoring* Jaringan Penerangan Jalan Umum (PJU). Diperoleh beberapa kesimpulan, diantaranya:

1. Sering terjadi kegagalan koneksi dengan jaringan sehingga data tidak terbaca karena konektivitas jaringan dan *Ethernet* yang tidak stabil.
2. Terganggunya koneksi *Ethernet* dengan PC sering terganggu karena faktor jaringan yang tidak stabil dan kualitas kabel LAN yang kurang baik.
3. Persentase *error* dalam jarak pengujian 10-40 meter adalah sebesar 0,05%. Dan persentase keberhasilan pengiriman datanya adalah 99,95%.
4. Jangkauan koneksi *Ethernet* sejauh 40 meter dipengaruhi oleh kemampuan daya pancar koneksi *router* yang digunakan.
5. *Range* PV bernilai *error* saat tegangannya bernilai diatas 21,6 V dan arusnya diatas 2,8 A.
6. *Range* Baterai bernilai *error* saat tegangannya berada dibawah 8,5 V dan diatas 15 V, dan temperaturnya diatas 28°C.
7. *Range* LED bernilai *error* saat luminasinya diatas 60%.
8. Dari hasil *survey* efektifitas HMI, didapatkan penilaian desain HMI sebesar 8,479 , aksestabilitas sebesar 8,36 dan informatif sebesar 8,41.

5.2 Saran

Berikut merupakan saran-saran yang dapat diberikan untuk implementasi dan pengembangan lebih lanjut dari sistem ini.

1. Desain HMI diperbarui agar lebih baik lagi.
2. Perlu adanya pembaruan sistem komunikasi yang lebih efektif dan efisien dari segi harga dan kualitas pengiriman datanya.

3. Perlu adanya penambahan notifikasi personal melalui *Handphone*.

DAFTAR PUSTAKA

- [1] Umacina, Dicky, *Lampu PJU Banyak Rusak, Jalan Tanjung Selor Gelap*, URL: korankaltara.com/lampu-pju-banyak-rusak-jalan-tanjung-selor-gelap, Diakses pada tanggal 17 Januari 2019
- [2], *Penerangan Jalan Umum*. URL:<http://eprints.pol-sri.ac.id> , Diakses pada tanggal 19 Desember 2018
- [3] Shah, Poonam, *Introduction to Human Machine Interface (HMI)*, URL:[http://www.doyouknow.in/Articles/Technology/Introduction -To-Human-Machine-Interface-HMI-Human-Machine-Interface-Design-Human-Machine-Interface-Software.aspx](http://www.doyouknow.in/Articles/Technology/Introduction-To-Human-Machine-Interface-HMI-Human-Machine-Interface-Design-Human-Machine-Interface-Software.aspx), Diakses pada tanggal 1 Desember 2018.
- [4] Wicaksono, Christian Agung, *Telemetering Jaringan Penerangan Jalan Umum Menggunakan Modul Komunikasi Radio Frequency dengan Metode Autoroute*, **Tugas Akhir**, D3 Teknik Elektro, Institut Teknologi Sepuluh Nopember Surabaya, 2018.
- [5], *Arduino Mega 2560*, URL: www.it-jurnal.com, Diakses pada tanggal 19 Desember 2018
- [6], *Ethernet W5100*, URL: <https://eprintas.umm.ac.id>, Diakses pada tanggal 19 Desember 2018.
- [7], *Software Arduino*, URL: <http://eprints.polsri.ac.id/>, Diakses pada tanggal 19 Desember 2018.
- [8], *Modbus*, URL:<https://www.prosoft-technology.com>, Diakses pada tanggal 19 Desember 2018
- [9] Sianipar, R. H., *Pemrograman Visual Basic. Net*, Bandung, Informatika, 2014.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A HASIL PENGUJIAN HMI

A.1 Pengujian PJU002

No	Komponen	Indikator	Nilai	Status Komponen	Status pada Main Menu
1	PV	<i>Voltage</i>	20 V	Normal	NORMAL
		<i>Current</i>	1 A		
		<i>Lumination</i>	53%		
		<i>Temperature</i>	25°C		
	Baterai	<i>Voltage</i>	10 V	Normal	
		<i>Current</i>	1 A		
		<i>Temperature</i>	25°C		
	LED	<i>Voltage</i>	11 V	Normal	
		<i>Current</i>	2 A		
		<i>Lumination</i>	34%		
2	PV	<i>Voltage</i>	28 V	<i>Error</i>	<i>ERROR</i>
		<i>Current</i>	1 A		
		<i>Lumination</i>	52%		
		<i>Temperature</i>	24°C		
	Baterai	<i>Voltage</i>	10 V	Normal	
		<i>Current</i>	1 A		
		<i>Temperature</i>	24°C		
	LED	<i>Voltage</i>	11 V	Normal	
		<i>Current</i>	2 A		
		<i>Lumination</i>	33%		
3	PV	<i>Voltage</i>	19 V	Normal	<i>ERROR</i>
		<i>Current</i>	1 A		
		<i>Lumination</i>	52%		
		<i>Temperature</i>	24°C		
	Baterai	<i>Voltage</i>	12 V	<i>Error</i>	
		<i>Current</i>	1 A		
		<i>Temperature</i>	28°C		
	LED	<i>Voltage</i>	11 V	Normal	
		<i>Current</i>	2 A		
<i>Lumination</i>		33%			

No	Komponen	Indikator	Nilai	Status Komponen	Status pada Main Menu
4	PV	<i>Voltage</i>	19 V	Normal	<i>ERROR</i>
		<i>Current</i>	1 A		
		<i>Lumination</i>	52%		
		<i>Temperature</i>	24°C		
	Baterai	<i>Voltage</i>	12 V	Normal	
		<i>Current</i>	1 A		
		<i>Temperature</i>	24°C		
	LED	<i>Voltage</i>	11 V	<i>Error</i>	
		<i>Current</i>	2 A		
<i>Lumination</i>		71%			

A.2 Pengujian PJU003

No	Komponen	Indikator	Nilai	Status Komponen	Status pada Main Menu
1	PV	<i>Voltage</i>	17 V	Normal	NORMAL
		<i>Current</i>	2 A		
		<i>Lumination</i>	81%		
		<i>Temperature</i>	34 °C		
	Baterai	<i>Voltage</i>	10 V	Normal	
		<i>Current</i>	1 A		
		<i>Temperature</i>	25°C		
	LED	<i>Voltage</i>	10 V	Normal	
		<i>Current</i>	0 A		
<i>Lumination</i>		22%			
2	PV	<i>Voltage</i>	28 V	<i>Error</i>	<i>ERROR</i>
		<i>Current</i>	2 A		
		<i>Lumination</i>	81%		
		<i>Temperature</i>	34 °C		
	Baterai	<i>Voltage</i>	10 V	Normal	
		<i>Current</i>	1 A		
		<i>Temperature</i>	25°C		
	LED	<i>Voltage</i>	10 V	Normal	
		<i>Current</i>	0 A		
<i>Lumination</i>		22%			

No	Komponen	Indikator	Nilai	Status Komponen	Status pada Main Menu
3	PV	<i>Voltage</i>	21 V	Normal	<i>ERROR</i>
		<i>Current</i>	1 A		
		<i>Lumination</i>	86%		
		<i>Temperature</i>	36 °C		
	Baterai	<i>Voltage</i>	14 V	<i>Error</i>	
		<i>Current</i>	1 A		
		<i>Temperature</i>	4°C		
	LED	<i>Voltage</i>	10 V	Normal	
		<i>Current</i>	0 A		
<i>Lumination</i>		22%			
4	PV	<i>Voltage</i>	19 V	Normal	<i>ERROR</i>
		<i>Current</i>	2 A		
		<i>Lumination</i>	81%		
		<i>Temperature</i>	39 °C		
	Baterai	<i>Voltage</i>	10 V	Normal	
		<i>Current</i>	1 A		
		<i>Temperature</i>	25°C		
	LED	<i>Voltage</i>	11 V	<i>Error</i>	
		<i>Current</i>	1 A		
<i>Lumination</i>		66%			

A.3 Pengujian PJU004

No	Komponen	Indikator	Nilai	Status Komponen	Status pada Main Menu
1	PV	<i>Voltage</i>	20 V	Normal	NORMAL
		<i>Current</i>	1 A		
		<i>Lumination</i>	30%		
		<i>Temperature</i>	12°C		
	Baterai	<i>Voltage</i>	9 V	Normal	
		<i>Current</i>	1 A		
		<i>Temperature</i>	23°C		
	LED	<i>Voltage</i>	5 V	Normal	
		<i>Current</i>	1 A		
<i>Lumination</i>		36%			

No	Komponen	Indikator	Nilai	Status Komponen	Status pada Main Menu
2	PV	<i>Voltage</i>	23 V	<i>Error</i>	<i>ERROR</i>
		<i>Current</i>	1 A		
		<i>Lumination</i>	30%		
		<i>Temperature</i>	12°C		
	Baterai	<i>Voltage</i>	9 V	Normal	
		<i>Current</i>	1 A		
		<i>Temperature</i>	23°C		
	LED	<i>Voltage</i>	5 V	Normal	
		<i>Current</i>	1 A		
<i>Lumination</i>		35%			
3	PV	<i>Voltage</i>	19 V	Normal	<i>ERROR</i>
		<i>Current</i>	1 A		
		<i>Lumination</i>	30%		
		<i>Temperature</i>	12°C		
	Baterai	<i>Voltage</i>	11 V	<i>Error</i>	
		<i>Current</i>	1 A		
		<i>Temperature</i>	36°C		
	LED	<i>Voltage</i>	5 V	Normal	
		<i>Current</i>	1 A		
<i>Lumination</i>		36%			
4	PV	<i>Voltage</i>	19 V	Normal	<i>ERROR</i>
		<i>Current</i>	1 A		
		<i>Lumination</i>	30%		
		<i>Temperature</i>	12°C		
	Baterai	<i>Voltage</i>	11 V	Normal	
		<i>Current</i>	1 A		
		<i>Temperature</i>	24°C		
	LED	<i>Voltage</i>	5 V	<i>Error</i>	
		<i>Current</i>	1 A		
<i>Lumination</i>		80%			

LAMPIRAN B

SURVEY EFEKTIFITAS HMI

KUISIONER EFEKTIFITAS HMI

Nama : Fajar
 Jenis Kelamin : LAKI-LAKI

Keterangan	Penilaian			Nilai
	Kurang	Baik	Sangat Baik	
Desain Desain HMI pada Microsoft Visual Studio 2017			✓	8,99
Aksestabilitas : Kemudahan pengoperasian HMI		✓		7,90
Informatif : Penyampaian informasi pada HMI		✓		7,5

Saran :
 UTAMAKAN KINERJA HMI, KECEPATAN AKSES DAN INFORMASI YANG LEBIH LENGKAP

KUISIONER EFEKTIFITAS HMI

Nama : Saadudin
 Jenis Kelamin : LAKI-LAKI

Keterangan	Penilaian			Nilai
	Kurang	Baik	Sangat Baik	
Desain Desain HMI pada Microsoft Visual Studio 2017		✓		8
Aksestabilitas : Kemudahan pengoperasian HMI		✓		8
Informatif : Penyampaian informasi pada HMI		✓		7,5

Saran :
 Tambahkan fitur yang lebih banyak

KUISIONER EFEKTIFITAS HMI

Nama : Anel
 Jenis Kelamin : Perempuan

Keterangan	Penilaian			Nilai
	Kurang	Baik	Sangat Baik	
Desain : Desain HMI pada Microsoft Visual Studio 2017			✓	9
Aksestabilitas : Kemudahan pengoperasian HMI		✓		8
Informatif : Penyampaian informasi pada HMI		✓		8

Saran :
 Perbaiki sistem komunikasi agar tidak
 terus nyambung

KUISIONER EFEKTIFITAS HMI

Nama : Ahy
 Jenis Kelamin : Laki-laki

Keterangan	Penilaian			Nilai
	Kurang	Baik	Sangat Baik	
Desain : Desain HMI pada Microsoft Visual Studio 2017		✓		8
Aksestabilitas : Kemudahan pengoperasian HMI			✓	9
Informatif : Penyampaian informasi pada HMI			✓	9

Saran :
 Perbaiki desain agar lebih menarik

KUISIONER EFEKTIVITAS HMI

Nama : Alifah
 Jenis Kelamin : perempuan

Keterangan	Penilaian			Nilai
	Kurang	Baik	Sangat Baik	
Desain Desain HMI pada Microsoft Visual Studio 2017			✓	8,8
Aksesibilitas : Kemudahan pengoperasian HMI		✓	.	7,5
Informatif Penyampaian informasi pada HMI			✓	9,1

Saran : kembangkan sistem komunikasi agar tidak putus nyambung

KUISIONER EFEKTIVITAS HMI

Nama : Sofyan
 Jenis Kelamin : Laki-laki

Keterangan	Penilaian			Nilai
	Kurang	Baik	Sangat Baik	
Desain : Desain HMI pada Microsoft Visual Studio 2017			✓	9
Aksesibilitas : Kemudahan pengoperasian HMI			✓	8,5
Informatif Penyampaian informasi pada HMI		✓		8

Saran : Lebih Informatif

LAMPIRAN C DATASHEET

C.1 Ethernet Shield



W5100 Ethernet Shield

-A high performance Ethernet shield for Arduino

Overview



W5100 Ethernet shield is a WIZnet W5100 breakout board with POE and Micro-SD designed for Arduino platform. 5V/3.3V compatible operation voltage level makes it compatible with Arduino boards, leafmaple, and other Arduino compatible board.

Features

- With Micro SD Interface
- 5V/3.3V double operational voltage level
- 10Mb/100Mb Ethernet socket with POE
- All electronic brick interface are broken out
- Operation temperature: -40°C ~ +85°C

Specifications

PCB size	55.88mm X 68.58mm X 1.6mm
Indicators	TX, RX, CDL, FEX, SPD, LNK
Power supply	5V
Communication Protocol	SPI
RoHS	Yes

Electrical Characteristics

Specification	Min	Type	Max	Unit
Power Voltage	3V	-	5.5	VDC
Input Voltage VH:	3	-	5.5	V
Input Voltage VL:	-0.3	0	0.5	V
Current	-	-	100	mA

Hardware

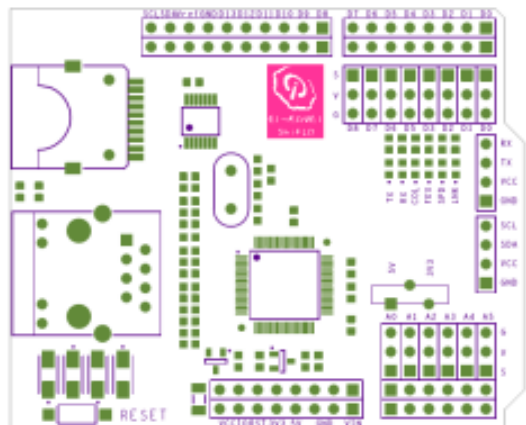


Figure 1 Top Map

Arduino PIN	Description
D0	Rx/Breakout
D1	Tx/Breakout
D2	Breakout
D3	Breakout
D4	SD_CS
D5	Breakout
D6	Breakout
D7	Breakout
D8	Breakout
D9	W5100_Reset
D10	W5100_CS
D11	MOSI
D12	MISO
D13	SCK

A0	Breakout
A1	Breakout
A2	Breakout
A3	Breakout
A4	IIC_SDA/Breakout
A5	IIC_SCL/Breakout

Installation

When install W5100 Ethernet shield to ItEadduino, please check the operation voltage level of development board. If the voltage is 3.3V (IFLAT32,Leafmaple), set the Operation Level Setting switch to 3.3V. If the voltage is 5V(Arduino), set the Operation Level Setting switch to 5V.

ItEadduino communicates with both the W5100 and SD card using the SPI bus. This is on digital pins 11, 12, and 13 on the UNO/Duemilanove and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general I/O. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

Inductor LED

The shield contains a number of informational LEDs:

- LNK: Indicates the presence of a network link and flashes when the shield transmits or receives data
- FEX: Indicates that the network connection is full duplex

- SPD: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COL: flashes when network collisions are detected

Revision History

Rev.	Description	Release date
v1.0	Initial version	2012-09-14

C.2 Arduino Mega 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies
half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



Technical Specification

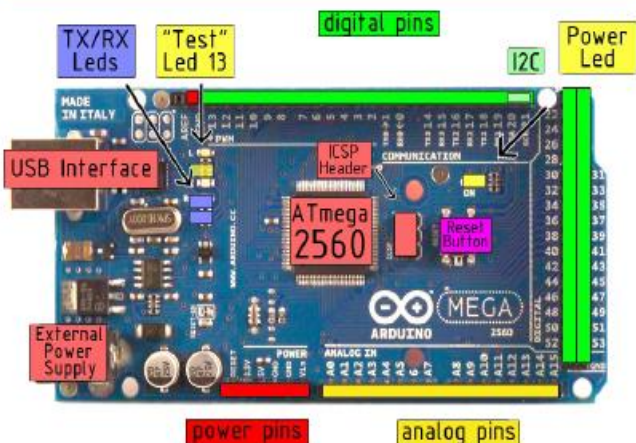


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 (Interrupt 0), 3 (Interrupt 1), 18 (Interrupt 5), 19 (Interrupt 4), 20 (Interrupt 3), and 21 (Interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C:** 20 (SDA) and 21 (SCL). Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares RADIONICS



Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



radiospares **RADIONICS**



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your skeeth you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // Initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power.

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000);                // wait for a second
}
```

Done compiling. Press Compile button (to check for errors)

Upload

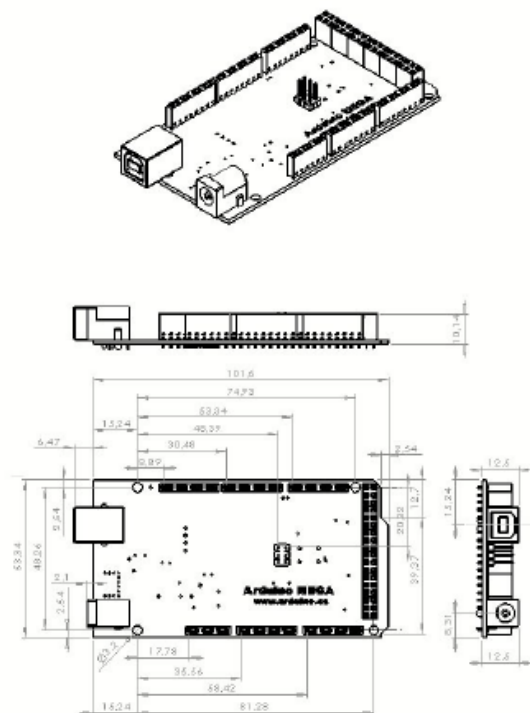
TX RX Flashing

Blinking Led!



radiospares **RADIONICS**





radiospares **RADIONICS**



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or misstatement by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, the producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any application-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under the terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.IT. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forests.



radiospares

RADIONICS



-----Halaman ini sengaja dikosongkan-----

RIWAYAT PENULIS



Nama : Devi Dwi Ariyani
TTL : Surabaya, 19 Desember
1996
Jenis Kelamin : Perempuan
Agama : Islam
Alamat : Gunung Anyar Lor
Gang Baru I/102C,
Surabaya
Telp/HP : 085607282220
E-mail : devidwiari
yani@gmail.com

RIWAYAT PENDIDIKAN

- 2003 – 2009 : SDN Rungkut Menanggal II/583 Surabaya
- 2009 – 2012 : SMP Al- Islah Surabaya
- 2012 – 2015 : SMA Dr. Soetomo Surabaya
- 2015 – 2019 : Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya

PENGALAMAN KERJA

- Kerja Praktek di PT. PLN Persero, Mojokerto (Juli-Agustus 2017)
- Kerja Praktek di PT. PAL Indonesia Persero, Surabaya (Desember 2017 - Februari 2018)

PENGALAMAN ORGANISASI

- Staff Departemen Big Event HIMAD3TEKTRO 2016-2017
- Staff Departemen PSDM BEM FTI-ITS 2016-2017
- Kabiro Departemen Ristek HIMAD3TEKTRO 2017-2018
- Divisi Materi Trainer Keilmiah ITS 2017-2018