TUGAS AKHIR – TE145561

MONITORING PEMAKAIAN DAYA PJU PADA PANEL METER

Sanynita Kiskindy
NRP 2212038006
Indra Insan Prasetyo
NRP 2212038022

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015

TUGAS AKHIR – TE145561

MONITORING PEMAKAIAN DAYA PJU PADA PANEL METER

Sanynita Kiskindy
NRP 2212038006
Indra Insan Prasetyo
NRP 2212038022

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015

FINAL PROJECT – **TE145561**

MONITORING POWER CONSUMPTION OF ROAD LIGHTING ON PANEL METERS

Sanynita Kiskindy
ID 2212038006
Indra Insan Prasetyo
ID 2212038022

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.

ELECTRICAL ENGINEERING D3 STUDY PROGRAM
Industrial Technology Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2015

FINAL PROJECT – **TE145561**

MONITORING POWER CONSUMPTION OF ROAD LIGHTING ON PANEL METERS

Sanynita Kiskindy
ID 2212038006
Indra Insan Prasetyo
ID 2212038022

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.

ELECTRICAL ENGINEERING D3 STUDY PROGRAM
Industrial Technology Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2015

MONITORING PEMAKAIAN DAYA PJU PADA PANEL METER

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Bidang Studi Teknik Listrik
Program Studi D3 Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Menyetujui :
Dosen Pembimbing,

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI INDUSTRI

Ir. Josaphat Pramudijanto, M.Eng.
NIP. 19621005 199003 1 003

SURABAYA
JUNI, 2015

MONITORING PEMAKAIAN DAYA PJU PADA PANEL METER

Nama Mahasiswa 1	**:**	**Sanynita Kiskindy**
NRP	**:**	**2212 038 006**
Nama Mahasiswa 2	**:**	**Indra Insan Prasetyo**
NRP	**:**	**2212 038 022**
Dosen Pembimbing	**:**	**Ir. Josaphat Pramudijanto, M.Eng.**
NIP	**:**	**19621005 199003 1 003**

ABSTRAK

Pencatatan pemakaian daya pada panel meter Penerangan Jalan Umum (PJU) oleh petugas catat meter (cater) masih dilakukan secara manual. Petugas cater setiap bulannya datang ke lokasi panel meter PJU untuk mencatat pemakaian daya yang terpakai. Pencatatan secara manual ini tidaklah efisien karena seringkali petugas cater tidak datang ke lokasi panel dan hanya memperkirakan data pemakaian daya dari pemakaian sebelumnya.

Melihat permasalahan tersebut, maka dibuatlah alat monitoring pemakaian daya PJU pada sebuah panel meter untuk meminimalisirnya. Sensor arus dan sensor tegangan digunakan untuk mengukur arus dan tegangan pada beban untuk mendapatkan nilai daya. Data hasil pengukuran sensor akan diolah Arduino dan juga disimpan pada *microSD*. Hasil olah data akan dikirim melalui media komunikasi *WiFi* yang kemudian ditampilkan pada PC melalui *software* LabVIEW. *Software* ini menampilkan data arus, tegangan, daya, pemakaian daya, dan indikator beban. Semua data yang dikirimkan ke PC akan disimpan secara otomatis pada *database* PC.

Alat ini mampu mengukur arus dan tegangan dengan hasil yang linier dengan persentase kesalahan rata-rata baca sensor tegangan sebesar 1,5% dan sensor arus sebesar 1,38%. Pembacaan sensor arus, tegangan, dan nilai daya ditampilkan pada LCD dan LabVIEW. Tampilan LabVIEW dilengkapi dengan kolom *login* khusus admin untuk memonitor *database* pemakaian daya pelanggan PJU.

Kata Kunci : Monitoring, Arus, Tegangan, Daya, PJU

Halaman ini sengaja dikosongkan

MONITORING POWER CONSUMPTION OF ROAD LIGHTING ON PANEL METERS

***Student Name* 1 : Sanynita Kiskindy**
***ID* : 2212 038 006**
***Student Name* 2 : Indra Insan Prasetyo**
***ID* : 2212 038 022**
***Supervisor* : Ir. Josaphat Pramudijanto, M.Eng.**
***ID* : 19621005 1990031003**

ABSTRACT

Nowadays, power consumption of road lighting (PJU) *is records manually. Every month, the "cater" officers come to the location of road lighting panel meter for recording power consumption. This method is not efficient because the officers did not come to panel meters location and they are predicting the power consumption data based on the previous data.*

Based on these problems there was an idea of monitoring power consumption of road lighting on panel meters to minimize the problems. Current sensor and voltage sensor are placed on for measure the value of current and voltage from the load to get the power value. The measurements of sensors will be processed by Arduino and also will be stored in microSD. The processed data will be transfer over Wi-Fi communication media then the data will be displayed on the PC through LabVIEW. *It is displaying current, voltage, power, power consumption, and indicator lamp. The results of reading data will be stored automatically in* PC *database.*

The prototype is capable to measure electric current and voltage linearly with error percentage 1.5% for voltage sensor, 1.38% for current sensor. The measurements data will be displayed on LCD *and* LabVIEW. LabVIEW interface are *added by login column for admin to monitoring the database of* PJU *consumers.*

***Keywords*: *Monitoring, Current, Voltage, Power,* PJU**

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah serta karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul :

”MONITORING PEMAKAIAN DAYA PJU PADA PANEL METER”

Tugas Akhir ini merupakan syarat untuk kelulusan program sudi D3 Teknik Elektro ITS.

Dengan selesainya Tugas Akhir ini penulis menyampaikan terima kasih sebesar-besarnya kepada:

1. Kedua orang tua atas limpahan doa, kasih sayang dan perhatian yang telah diberikan kepada penulis.
2. Bapak Eko Setijadi, ST., MT., Ph.D selaku Ketua Program D3 Teknik Elektro Bidang Studi Teknik Listrik, FTI-ITS Surabaya.
3. Bapak Ir. Josaphat Pramudijanto, M.Eng. selaku dosen pembimbing
4. Para sahabat yang senantiasa memberi semangat, motivasi, dan membantu khususnya Muhammad Alif, Ladysa Ashadita, teman-teman D3K PLN 2012, dan Armageddon.
5. Semua pihak yang telah banyak membantu untuk menyelesaikan Tugas Akhir ini yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari masih banyak kekurangan dalam Tugas Akhir ini. Kritik dan saran untuk perbaikan Tugas Akhir ini sangat diperlukan. Akhir kata semoga tugas ini dapat bermanfaat bagi kita semua.

Surabaya, Juni 2015

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL i
LEMBAR PENGESAHAN iii
ABSTRAK v
ABSTRACK vii
KATA PENGANTAR ix
DAFTAR ISI xi
DAFTAR GAMBAR xiii
DAFTAR TABEL xv

BAB I PENDAHULUAN
1.1 Latar Belakang 1
1.2 Permasalahan 1
1.3 Tujuan 2
1.4 Batasan Masalah 2
1.5 Sistematika Laporan 2
1.6 Relevansi 3

BAB II TEORI PENUNJANG
2.1 Penerangan Jalan Umum (PJU) 5
2.2 Alat Pengukur dan Pembatas PJU 6
2.3 Sensor Tegangan 9
2.4 Sensor Arus 10
2.5 Arduino UNO 11
2.6 *Ethernet Shield* 13
2.7 *Real Time Clock* (RTC) 14
2.8 LCD 15
2.9 Modul I2C *Backpack* 17
2.10 Modul Relai 18
2.11 *Wireless Outdoor* TP-Link TL-WA5210G 18
2.12 LabVIEW 19

BAB III PERANCANGAN DAN PEMBUATAN ALAT
3.1 Perancangan Sistem Keseluruhan 23
3.2 Perancangan Mekanik 24
3.3 Perancangan *Hardware* 25
3.3.1 Perancangan *Power Supply* 25
3.3.2 Perancangan Sensor Tegangan 26

3.3.3 Modul Sensor Arus 26
3.3.4 Perancangan LCD dengan I2C..... 27
3.3.5 Perancangan RTC 27
3.3.6 Perancangan *Driver* Relai 28
3.3.7 Penggunaan Modul Arduino UNO..... 29
3.3.8 *Setting* Sistem Komunikasi 30
3.4 Perancangan *Software*..... 31
3.4.1 Program pada Arduino IDE 31
3.4.2 LabVIEW 2014..... 33
3.4.3 Tampilan Monitoring 35

BAB IV PENGUJIAN DAN ANALISA ALAT
4.1 Pengujian *Power Supply* 37
4.2 Pengujian Sensor Tegangan..... 39
4.3 Pengujian Sensor Arus..... 41
4.4 Pengujian Relai 44
4.5 Pengujian *WiFi* TP-LINK 45
4.6 Pengujian HMI 47
4.6.1 Pengujian Kolom *Login* 47
4.6.2 Pengujian Tampilan Monitoring 48
4.6.3 Pengujian *Database* 49
4.7 Pengujian Sistem Secara Keseluruhan 50
4.8 Analisa Keseluruhan..... 57

BAB V PENUTUP
5.1 Kesimpulan 59
5.2 Saran..... 59

DAFTAR PUSTAKA..... 61

LAMPIRAN
Lampiran A Program A-1
Lampiran B Dokumentasi Alat dan Pengujian..... B-1
Lampiran C *Datasheet*..... C-1
Lampiran D Daftar Riwayat Penulis D-1

DAFTAR GAMBAR

Gambar 2.1 Panel Meter PJU 1 Fasa 5
Gambar 2.2 Diagram Satu Garis Letak Penempatan APP 6
Gambar 2.3 KWh Meter Digital 1 Fasa 7
Gambar 2.4 Skema KWh Meter Elektromagnet 1 Fasa 7
Gambar 2.5 Konstruksi KWh Meter 1 Fasa 8
Gambar 2.6 Trafo Nol 500 mA 10
Gambar 2.7 CT Ta12-100 11
Gambar 2.8 Modul Arduino UNO 12
Gambar 2.9 Modul *Ethernet Shield* 14
Gambar 2.10 Pin IC DS3231 14
Gambar 2.11 LCD 16x2 16
Gambar 2.12 Modul I2C LCD *Backpack* 17
Gambar 2.13 TP-Link TL-WA5210G 19
Gambar 2.14 Tampilan Utama *Software* LabVIEW 21
Gambar 3.1 Diagram Fungsional Sistem Alat 23
Gambar 3.2 Purwarupa Panel Monitoring Daya PJU 24
Gambar 3.3 Purwarupa PJU 25
Gambar 3.4 Rangkaian *Power Supply* 25
Gambar 3.5 Rangkaian Sensor Tegangan 26
Gambar 3.6 Cara Menghubungkan Sensor Arus ke Arduino 26
Gambar 3.7 Cara Menghubungkan I2C LCD ke Arduino 27
Gambar 3.8 Cara Menghubungkan Modul RTC pada Arduino 27
Gambar 3.9 Rangkaian *Driver* Relai 28
Gambar 3.10 Penggunaan *Port* pada Arduino UNO 29
Gambar 3.11 Tampilan Menu Utama AP 30
Gambar 3.12 Penyetelan Alamat IP AP 30
Gambar 3.13 Diagram Alir Arduino 32
Gambar 3.14 Diagram Alir LabVIEW 34
Gambar 3.15 Tampilan Monitoring Pemakaian Daya PJU 35
Gambar 4.1 Tampilan Fisik *Hardware* Keseluruhan 37
Gambar 4.2 Pengujian *Power Supply* 37
Gambar 4.3 Skematik Pengujian *Power Supply* 38
Gambar 4.4 Diagram Pengujian Sensor Tegangan 39
Gambar 4.5 Karakteristik Sensor Tegangan 40
Gambar 4.6 Kesalahan Pengukuran Sensor Tegangan 41
Gambar 4.7 Diagram Pengujian Sensor Arus 42
Gambar 4.8 Pengujian Sensor Arus 42

Gambar 4.9 Karakteristik Sensor Arus 43
Gambar 4.10 Kesalahan Pengukuran Sensor Arus 44
Gambar 4.11 Skema Pengujian *Driver* Relai 45
Gambar 4.12 Kondisi Lampu Indikator saat Pengujian *Driver* Relai. 45
Gambar 4.13 Koneksi *WiFi* TP-LINK 46
Gambar 4.14 Hasil *Ping Test* 46
Gambar 4.15 Tampilan Kolom *Login Admin* dan *Form User* 48
Gambar 4.16 Tampilan 1 *Form Admin* 48
Gambar 4.17 Tampilan 2 *Form Admin* 49
Gambar 4.18 Pengujian *Database* pada PC 49
Gambar 4.19 Pengujian Alat Keseluruhan 50
Gambar 4.20 Tampilan Awal LCD 50
Gambar 4.21 Tampilan Monitoring pada LCD 50
Gambar 4.22 Penyalaan Purwarupa PJU 51
Gambar 4.23 Tampilan LCD dan Indikator pada Kondisi Pertama ... 52
Gambar 4.24 Tampilan HMI pada Kondisi Pertama 52
Gambar 4.25 Tampilan LCD dan Indikator pada Kondisi Kedua 53
Gambar 4.26 Tampilan HMI pada Kondisi Kedua 53
Gambar 4.27 Tampilan LCD dan Indikator pada Kondisi Ketiga 54
Gambar 4.28 Tampilan HMI pada Kondisi Ketiga 54
Gambar 4.29 Tampilan *Database* yang Tersimpan pada *MicroSD* 55
Gambar 4.30 *Login* ke Data Pelanggan Jalan Gebang Kidul 55
Gambar 4.31 Data Pelanggan Jalan Gebang Kidul 56
Gambar 4.32 *Login* ke Data Pelanggan Jalan Keputih 56
Gambar 4.33 Data Pelanggan Jalan Keputih 57

DAFTAR TABEL

Tabel 2.1 Konfigurasi I2C LCD *Backpack* pada Arduino 17
Tabel 4.1 Hasil Pengujian *Power Supply* 38
Tabel 4.2 Hasil Pengujian Sensor Tegangan 39
Tabel 4.3 Persentase Kesalahan Pengukuran Sensor Tegangan 41
Tabel 4.4 Hasil Pengujian Sensor Arus 42
Tabel 4.5 Persentase Kesalahan Pengukuran Sensor Arus 43
Tabel 4.6 Hasil Pengujian *WiFi* 47

Halaman sengaja dikosongkan

BAB I
PENDAHULUAN

1.1 Latar Belakang

PJU (Penerangan Jalan Umum) merupakan sebuah sistem penerangan yang bersifat publik (untuk kepentingan bersama) dan biasanya terpasang pada ruas-ruas jalan, taman, dan tempat umum lainnya. Setiap jalur PJU memiliki kWh meter sendiri dan terpasang pada panel. Namun, selama ini pencatatan pemakaian daya PJU masih dilakukan secara manual yaitu petugas catat meter (cater) harus datang ke lokasi panel dan mencatat pemakaian daya PJU di atas *form*. Pencatatan secara manual ini tidaklah efisien, karena pihak administrasi PLN perlu merekap kembali data pemakaian daya pelanggan PJU dari setoran *form* petugas cater untuk pembuatan tagihan rekening listrik. Dalam beberapa kasus, petugas cater tidak datang ke lokasi panel meter PJU dan memperkirakan data pemakaian daya PJU dari pemakaian sebelumnya sehingga data yang didapatkan kurang akurat. Selain itu, resiko kehilangan data pemakaian daya PJU lebih tinggi dikarenakan *form* hilang atau basah. Hal ini menimbulkan masalah bagi PLN karena laporan bulanan dari pemakaian daya tersebut sangat dibutuhkan untuk membuat tagihan pembayaran rekening listrik dari jalur PJU tersebut.

Oleh karena itu, perlu adanya sebuah alat untuk memonitor pemakaian daya pada sebuah jalur PJU yang dapat memonitor pemakaian daya pada panel meter tersebut secara online melalui s*erver* sehingga pihak PLN dapat mengetahui kondisi kekinian pemakaian daya PJU dan petugas cater tidak perlu datang ke tempat lokasi panel PJU untuk melakukan pencatatan pemakaian daya PJU. Resiko kehilangan data pemakaian daya PJU juga lebih rendah karena data pemakaian daya PJU langsung disimpan di dalam *server* dan juga tersedia media penyimpanan lokal pada *microSD*.

1.2 Permasalahan

Dari latar belakang diatas, maka dapat dirumuskan permasalahan bahwa pencatatan pemakaian daya PJU secara manual tidak efisien karena petugas cater harus datang ke tempat lokasi panel PJU untuk melakukan pencatatan, metode pencatatan manual juga memiliki resiko kehilangan data pemakaian daya PJU sehingga beberapa petugas cater memperkirakan data pemakaian daya PJU sehingga data yang didapatkan tidak akurat.

1.3 Tujuan

Tujuan penulisan Tugas Akhir ini adalah:

1. Merancang dan membuat purwarupa dari Penerangan Jalan Umum (PJU), yang diukur dan dimonitor adalah daya pemakaian PJU.
2. Merancang dan membuat sistem metode pengiriman data dari pemakaian daya PJU menggunakan sensor arus dan sensor tegangan yang kemudian dikirimkan menggunakan Arduino melalui *WiFi*.
3. Membuat *Human Machine Interface* (HMI) berupa tampilan monitoring pemakaian daya PJU.

1.4 Batasan Masalah

Dari perumusan masalah di atas, maka batasan masalah dari Tugas Akhir ini adalah:

1. Parameter yang diukur adalah daya aktif.
2. Nilai Faktor Daya diasumsikan sama dengan 1.
3. PJU yang di monitoring adalah sebuah jalur PJU satu fasa pada komplek perumahan.
4. PC *server* dianggap terus menyala.
5. Jam RTC dan jam PC dianggap sama.
6. Penempatan sensor dianggap ideal tanpa ada kendala teknis.

1.5 Sistematika Laporan

Pembahasan pada Buku Tugas Akhir ini terdiri dari lima bab. Berikut diberikan penjelasan isi dari masing – masing bab, yaitu:

Bab I : PENDAHULUAN

Dalam bab ini membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi penulisan pada Tugas Akhir.

Bab II : TEORI PENUNJANG

Berisi teori penunjang yang mendukung dalam perencanaan dan pembuatan alat.

Bab III : PERANCANGAN DAN PEMBUATAN ALAT

Membahas tentang perencanaan dan pembuatan perangkat keras yang meliputi rangkaian-rangkaian, desain bangun, dan perangkat lunak yang meliputi program yang akan digunakan untuk mengaktifkan alat tersebut.

Bab IV : PENGUJIAN DAN ANALISA DATA

Dalam bab ini membahas tentang pengukuran, pengujian, dan analisa terhadap prinsip kerja dan proses dari alat yang telah dibuat.

Bab V : PENUTUP

Dalam bab ini berisi tentang penutup yang menjelaskan tentang kesimpulan dari Tugas Akhir dan saran – saran untuk pengembangan alat lebih lanjut.

1.6 Relevansi

Manfaat dari Pengerjaan Tugas Akhir (TA) ini untuk memenuhi kurikulum dari kuliah Diploma 3 (tiga) ini. Selain itu, juga dapat direalisasikan serta dimanfaatkan untuk proses pemantauan *database* pemakaian daya PJU secara otomatis guna memudahkan pihak PLN dalam mendapatkan data historis dari pemakaian daya PJU yang nantinya akan digunakan sebagai dasar pembuatan tagihan rekening listrik.

Halaman ini sengaja dikosongkan

BAB II
TEORI PENUNJANG

Pada bab ini menjelaskan mengenai teori – teori yang berkaitan dengan topik pengerjaan Tugas Akhir dan peralatan yang dipergunakan dalam pengerjaannya.

2.1 Penerangan Jalan Umum (PJU) [1]

PJU adalah singkatan dari Penerangan Jalan Umum yang dapat juga diartikan sebagai Lampu Penerangan yang dipasang oleh Pemerintah Daerah (PEMDA) setempat untuk sebesar-besarnya bagi kepentingan umum. PJU dipasang, dipelihara, dan dibayar rekeningnya oleh PEMDA sesuai kontrak yang telah disepakati dengan PLN. Pengelolaan PJU sepenuhnya wewenang dan tanggung jawab Pemerintah Daerah (PEMDA setempat / Pemerintah Kota) melalui Dinas Kebersihan & Pertamanan (DKP). Lampu penerangan jalan juga harus dipasang dengan menggunakan jaringan penerangan jalan tersendiri.

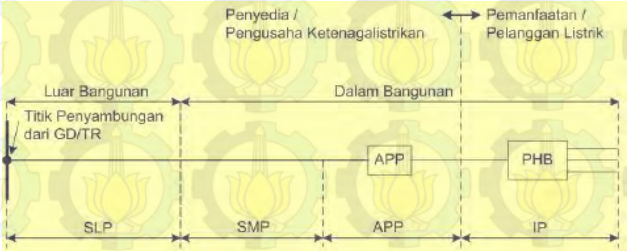
Gambar 2.1 Panel Meter PJU 1 Fasa

Setiap jalur PJU yang telah termeterisasi memiliki kWh meter sendiri dan terpasang pada panel. Panel kWh meter (Gambar 2.1) inilah yang dipasang serta dipelihara oleh pihak PLN. Kewajiban PLN dalam hal mengenai PJU adalah menyediakan pasokan aliran listrik sesuai kontrak yang telah dibayar dan disetujui PEMDA serta mengontrol penggunaan daya terpakai oleh PJU. Pencatatan pemakaian daya PJU saat ini masih dilakukan secara manual oleh petugas catat meter (cater). Petugas cater ditugaskan untuk datang ke lokasi panel meter untuk mencatat pemakaian daya PJU setiap bulannya. Data pemakaian daya

setiap pelangan PJU dicatat dalam sebuah *form*, yang kemudian diberikan kepada pihak administrasi PLN untuk direkap dan digunakan sebagai dasar pembuatan rekening listrik. Namun, dalam pelaksanaannya terkadang banyak petugas yang tidak datang ke lokasi panel meter sehingga mereka memperkirakan data pemakaian daya dari pemakaian bulan sebelumnya pada *form*. Selain itu, terkadang para petugas lupa tempat menempatkan *form* pencatatan sehingga kehilangan data pemakaian daya PJU sering sekali terjadi.

2.2 Alat Pengukur dan Pembatas PJU [1]

Untuk mengetahui besarnya tenaga listrik yang digunakan oleh pemakai/pelanggan listrik perlu dilakukan pengukuran dan pembatasan daya listrik. Alat pengukur dan pembatas atau biasa disebut APP merupakan alat yang digunakan PT. PLN (Persero) sebagai dasar dalam pembuatan rekening listrik. Letak penempatan APP dapat dilihat pada Gambar 2.2 berikut ini:



Gambar 2.2 Diagram Satu Garis Letak Penempatan APP

- Keterangan:

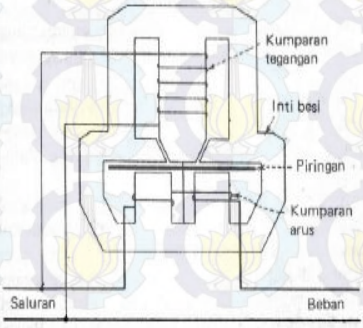
APP	: Alat Pengukur dan Pembatas
GD	: Gardu Distribusi
IP	: Instalasi Pelanggan
PHB	: Papan Hubung Bagi
SLP	: Sambungan Luar Pelayanan
SLTR	: Sambungan Tenaga Listrik Tegangan Rendah
SMP	: Sambungan Masuk Pelayanan
TR	: Jaringan tegangan Rendah

Pengukuran adalah untuk menentukan besarnya pemakaian daya dan energi listrik. Sedangkan yang dimaksud dengan pembatasan adalah pembatasan untuk menentukan batas pemakaian daya sesuai dengan

daya tersambung. KWh meter merupakan alat ukur untuk mengukur energi listrik dalam orde kWh. Gambar 2.3 berikut adalah contoh kWh meter.

Gambar 2.3 KWh Meter Digital 1 Fasa

Elemen alat ukur kWh meter satu fasa ditunjukkan pada Gambar 2.4 dalam bentuk skema. Kumparan arus dihubungkan seri dengan jala-jala, dan kumparan tegangan dihubungkan paralel. Kedua kumparan yang dililitkan pada sebuah kerangka logam dengan desain khusus melengkapi dua rangkaian magnet. Sebuah piringan aluminium ringan digantung di dalam senjang udara medan kumparan arus yang menyebabkan arus pusar mengalir di dalam piringan. Reaksi arus pusar dan medan kumparan tegangan membangkitkan sebuah torsi (aksi motor) terhadap piringan dan menyebabkannya berputar.



Gambar 2.4 Skema KWh Meter Elektromagnet 1 Fasa

Torsi yang dibangkitkan sebanding dengan kuat medan kumparan tegangan dan arus pusar di dalam piringan yang berturut-turut adalah fungsi kuat medan kumparan arus. Berarti jumlah putaran piringan sebanding dengan energi yang telah dipakai oleh beban dalam selang waktu tertentu, dan diukur dalam kiloWatt-jam (kWh). Poros yang menopang piringan aluminium dihubungkan melalui susunan roda gigi ke mekanisme jam dipanel alat ukur, melengkapi suatu pembacaan kWh yang terkalibrasi dalam desimal. Redaman piringan diberikan oleh dua magnet permanen kecil yang ditempatkan saling berhadapan pada sisi piringan. Bila piringan berputar, magnet-magnet permanen menginduksi arus pusar di dalamnya. Arus-arus pusar ini bereaksi dengan medan magnet dari magnet-magnet permanen kecil dan meredam gerakan piringan. Untuk lebih jelasnya dapat dilihat dalam Gambar 2.5.

Gambar 2.5 Konstruksi KWh Meter 1 Fasa

Jumlah energi yang dipergunakan ditunjukkan oleh putaran cakram, dinotasikan dengan simbol kWh yang diberikan dalam unit Watt jam per putaran. Dengan mengetahui nilai kWh, seorang pelanggan dapat menentukan konsumsi daya yang dipergunakan dengan cara menghitung putaran cakram dengan *stopwatch*. Jika waktu yang dibutuhkan cakram dalam detik untuk menyelesaikan satu putaran adalah *t*, dan daya dalam Watt adalah P = 3600 x kWh /t. Contoh, jika kWh = 7.2 dan satu putaran membutuhkan waktu 14.4 detik, maka dayanya adalah 1800 Watt. Metode ini dapat digunakan untuk menentukan konsumsi daya dari peralatan rumah tangga. Jika membeli sebuah kWh meter maka akan tercantum *x* putaran per kWh, artinya untuk mencapai 1 kWh dibutuhkan putaran sebanyak *x* kali putaran dalam setiap jamnya. Contohnya jika 1200 putaran per kWh maka harus ada 1200 putaran setiap jamnya untuk dikatakan sebesar satu kWh.

Jumlah kWh secara kumulatif dihitung dan pada akhir bulan dicatat oleh petugas. Besarnya pemakaian lalu dikalikan dengan tarif dasar listrik (TDL) ditambah dengan biaya abonemen dan pajak menghasilkan jumlah tagihan yang harus dibayarkan setiap bulannya. Sebagian besar meter listrik domestik masih dicatat secara manual, dengan cara perwakilan/utusan dari perusahaan listrik.

2.3 Sensor Tegangan [2]

Sensor tegangan adalah komponen yang digunakan untuk mengkonversi besaran tegangan pada listrik menjadi besaran analog dengan memperkecil nilainya menjadi tegangan referensi sehingga dapat dibaca pada rangkaian elektronik. Sensor tegangan dapat dibuat dengan menggunakan trafo.

Sebuah *transformator* yang sederhana pada dasarnya terdiri dari 2 lilitan atau kumparan kawat yang terisolasi yaitu kumparan primer dan kumparan sekunder. Pada kebanyakan *transformator*, kumparan kawat terisolasi ini dililitkan pada sebuah besi yang dinamakan dengan Inti Besi (*Core*). Ketika kumparan primer dialiri arus AC (bolak-balik) maka akan menimbulkan medan magnet atau *fluks* magnetik disekitarnya. Kekuatan Medan magnet (densitas *fluks* magnet) tersebut dipengaruhi oleh besarnya arus listrik yang dialirinya. Semakin besar arus listriknya semakin besar pula medan magnetnya. Fluktuasi medan magnet yang terjadi di sekitar kumparan pertama (primer) akan menginduksi GGL (Gaya Gerak Listrik) dalam kumparan kedua (sekunder) dan akan terjadi pelimpahan daya dari kumparan primer ke kumparan sekunder. Dengan demikian, terjadilah perubahan taraf tegangan listrik baik dari tegangan rendah menjadi tegangan yang lebih tinggi maupun dari tegangan tinggi menjadi tegangan yang rendah. Sedangkan Inti besi pada *transformator* atau trafo pada umumnya adalah kumpulan lempengan-lempengan besi tipis yang terisolasi dan ditempel berlapis-lapis dengan kegunaanya untuk mempermudah jalannya Fluks Magnet yang ditimbulkan oleh arus listrik kumparan serta untuk mengurangi suhu panas yang ditimbulkan.

Jenis *transformator* yang dapat digunakan untuk menurunkan tegangan AC dan salah satunya adalah trafo nol (engkel). Trafo ini hanya punya besar keluaran tegangan yang hanya satu macam ditambah tegangan nol. Pada dasarnya trafo biasa (engkel) akan menghasilkan tegangan keluaran di dua titik yaitu pada ujung-ujung kumparan sekundernya. Dalam pembuatan sensor tegangan pada Tugas Akhir ini

kami akan menggunakan trafo nol 500 mA. Trafo yang digunakan adalah jenis trafo yang mempunyai nilai arus yang kecil hal ini bertujuan agar sensitivitas tegangan kecil juga dapat dideteksi oleh trafo sehingga sensor tegangan dapat bekerja secara maksimal. Lilitan primer yang dihubungkan dengan tegangan jala-jala sebesar 220 V AC. Sedangkan yang dihubungkan dengan rangkaian pengkondisi adalah lilitan sekunder dan yang diambil untuk sekundernya adalah lilitan nomer 6. Sensor tegangan menggunakan trafo nol 500 mA dapat dilihat pada Gambar 2.6.

Gambar 2.6 Trafo Nol 500mA

2.4 Sensor Arus [3]

Sensor arus merupakan suatu piranti yang digunakan untuk mengukur besaran arus pada suatu sistem tenaga listrik. Sensor ini mengubah arus menjadi suatu besaran listrik (arus / tegangan) yang lebih proporsional untuk selanjutnya digunakan untuk keperluan *metering* maupun proteksi. Di samping untuk pengukuran arus, sensor arus juga digunakan untuk pengukuran jarak jauh, pengukuran daya dan energi. Contoh sensor arus yang umum digunakan adalah adalah *Current Transformer* (CT) atau trafo arus. *Transformator* arus biasa dipasang pada bagian sekunder yang berfungsi sebagai impedansi beban, sehingga *transformator* tidak benar-benar *short circuit.* Di saat arus terlalu tinggi dalam jaringan maka di perlukan CT sebagai pengubah pada pembacaan alat ukur. Jadi yang digunakan adalah progresif arus imbas dari hantaran sebuah rangkaian listrik bolak balik atau AC. Sebuah trafo arus harus bisa menghasilkan konversi arus yang akurat untuk pembacaan alat ukur.

Pada alat yang akan kami buat, sensor arus yang dipakai adalah sensor CT Ta12-100. Sensor ini memiliki kemampuan membaca arus

AC hingga 5 A dengan *range* arus keluaran hingga 5 mA. Modul sensor ini juga sudah dilengkapi dengan resistor 200 Ohm yang mengubah arus menjadi tegangan ukur. Sehingga tegangan keluaran pengukuran yang dapat terbaca adalah maksimum 1V. Sensor arus jenis CT Ta12-100 dapat dilihat pada Gambar 2.7.

Gambar 2.7 CT Ta12-100

Berikut adalah spesifikasi dari modul CT Ta12-100:

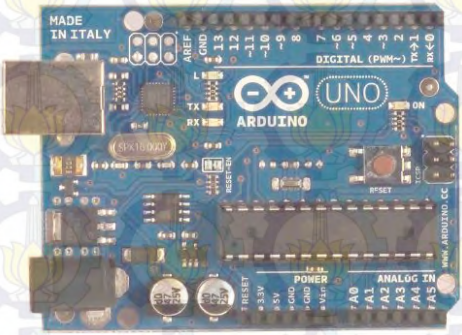
a. Membutuhkan pasokan listrik sebesar 5V
b. Jenis antarmuka : Analog
c. Pin S: *Analog output* ; Pin N: *Not Connected*; Pin G: GND
d. Rasio 1000:1
e. Arus *input*: 0~5A
f. Arus *output*: 0~5mA
g. Resistansi: 200 Ohm
h. Tegangan *sampling*: 1V
i. Memiliki respon yang cepat dengan sensitivitas yang tinggi

2.5 Arduino UNO [4]

Arduino UNO adalah *board* mikrokontroler berbasis ATmega 328. Memiliki 14 pin *input* dari *output* digital dimana 6 pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP *header*, dan tombol reset. Board Arduino UNO dapat disambungkan ke komputer dengan menggunakan kabel USB atau baterai sebagai *supply*.

Arduino UNO memiliki beberapa keunggulan dibanding mikrokontroler lain diantaranya tidak perlu perangkat *chip* programmer karena di dalamnya sudah ada *bootloader* yang akan menangani *upload* program dari komputer, karena sudah memiliki sarana komunikasi USB. Dalam hal koneksi USB-*to-serial* yaitu menggunakan fitur ATmega8U2

yang diprogram sebagai konverter USB-*to-serial* berbeda dengan *board* sebelumnya yang menggunakan *chip* FTDI *driver* USB-*to-serial*, sehingga pengguna laptop yang tidak memiliki *port* serial/RS323 bisa menggunakannya. Selain itu, bahasa pemrograman Arduino relatif mudah karena *software* Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap, serta memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada *board* Arduino. Misalnya *shield* GPS, *ethernet*, *SD Card*, dan lain-lain. Arduino UNO yang digunakan dapat dilihat pada Gambar 2.8



Gambar 2.8 Modul Arduino UNO

Arduino UNO dilengkapi dengan *static random-access memory* (SRAM) berukuran 2 KB untuk memegang data, *flash memory* berukuran 32 KB, dan *erasable programmable read-only memory* (EEPROM) untuk menyimpan program. Secara ringkas spesifikasi Arduino UNO ialah sebagai berikut:

a. Mikrokontroler : ATmega 328
b. Tegangan Operasi : 5 V
c. Tegangan *Input* : 7-12 V (yang direkomendasikan)
d. Tegangan *Input* (min-maks) : 6-20 V
e. Jumlah Pin I/O Digital :14 (6 diantaranya merupakan *output* PWM)
f. Jumlah Pin Analog *Input* : 6
g. Arus DC per Pin I/O : 40 mA
h. Arus DC pada Pin 3,3 V : 50 mA
i. *Flash Memory* : 32 KB (ATmega328) dimana 0,5 KB sebagai *bootloader*
j. SRAM : 2 KB (ATmega328)

k. EEPROM : 1 KB (ATmega328)
l. *Clock Speed* : 16 MHz

Masing-masing dari 14 pin digital di UNO dapat digunakan sebagai *input* atau *output* serta beroperasi dengan daya 5 Volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki *internal pull-up* resistor (secara *default* terputus) dari 20-50 kOhm. Selain itu, beberapa pin memiliki fungsi khusus:

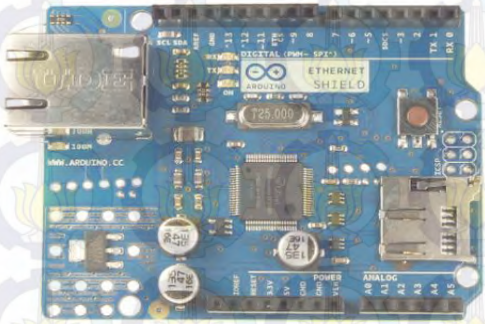
1. *Eksternal Interrupt*: Pin 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu *interrupt* pada nilai yang rendah, dengan batasan tepi naik atau turun, atau perubahan nilai.
2. PWM: Pin 3, 5, 6, 9, 10, dan 11. Menyediakan *output* PWM 8-bit dengan fungsi *analogWrite ()*.
3. SPI *library*. SPI: Pin 10 (SS), 11 (Mosi), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI.
4. LED: Pin 13. Ada *built-in* LED terhubung ke pin digital 13. Ketika pin bernilai nilai *HIGH*, LED *on*, ketika pin bernilai *LOW*, LED *off*.
5. Arduino UNO memiliki 6 masukan analog, berlabel A0 sampai dengan A5, yang masing-masing menyediakan 10 bit dengan resolusi (yaitu 1024 nilai yang berbeda).
6. I2C*:* A4 *(SDA)* dan A5 *(SCL)*. Dukungan I2C (TWI) komunikasi menggunakan *library wire*.
7. *Aref*. Tegangan referensi (0 sampai 5V saja) untuk *input* analog. Digunakan dengan fungsi *analogReference ()*.
8. *Reset*. Bawa baris ini LOW untuk me-*reset* mikrokontroler.

2.6 *Ethernet Shield* [2]

Ethernet Shield adalah modul yang dapat dipasangkan langsung di atas papan arduino untuk menambahkan fungsi LAN/*ethernet* dalam proyek rangkaian elektronika yang menggunakan papan pengembang arduino. *Ethernet shield* adalah modul yang berbasiskan cip *Ethernet* Wiznet W5100. Dengan menggunakan modul komunikasi ini, Arduino dapat dihubungkan dengan perangkat lain yang mendukung protokol TCP/IP atau UDP. Modul ini mendukung sampai dengan 4 koneksi secara bersamaan.

Board Ethernet Shield ini mempunyai sebuah *slot micro-SD*, yang dapat digunakan untuk menyimpan *file* yang dapat diakses melalui jaringan. *Onboard micro-SD card reader* diakses dengan menggunakan SD *library*. Arduino *board* berkomunikasi dengan W5100 dan SD *card*

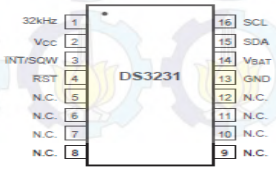
mengunakan bus SPI (*Serial Peripheral Interface*). *Ethernet Shield* dapat dihubungkan dengan modul *WiFi* dengan menggunakan *library* SPI.h dan *Ethernet*.h pada *listing* program Arduino. Komunikasi ini diatur oleh *library* SPI.h dan Ethernet.h. *Bus* SPI menggunakan pin digital 11, 12 dan 13 pada Arduino UNO. Pin digital 10 digunakan untuk memilih W5100 dan pin digital 4 digunakan untuk memilih SD *card*. Pin-pin yang sudah disebutkan sebelumnya tidak dapat digunakan untuk *input/output* umum ketika kita menggunakan *Ethernet shield*. Karena W5100 dan *SD card* berbagi *bus* SPI, hanya salah satu yang dapat aktif pada satu waktu. Modul *Ethernet Shield* yang digunakan dapat dilihat pada Gambar 2.9.



Gambar 2.9 Modul *Ethernet Shield*

2.7 *Real Time Clock* (RTC) [5]

RTC adalah jenis pewaktu yang bekerja berdasarkan waktu yang sebenarnya atau dengan kata lain berdasarkan waktu yang ada pada jam kita. Modul RTC yang dipakai mengunakan DS3231 yang merupakan salah satu tipe IC RTC yang dapat bekerja dalam daya listrik rendah. Di dalamnya berisi waktu jam dan kalender dalam format BCD. Waktu jam dan kalender memberikan informasi detik, menit, jam, hari, tanggal, bulan, dan tahun. Untuk bagian jam dapat berformat 24 jam atau 12 jam. Konfigurasi pin IC DS3231 pada RTC dapat dilihat pada Gambar 2.10.



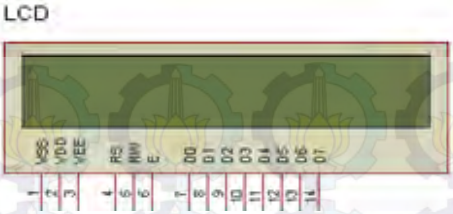
Gambar 2.10 Pin IC DS3231

Berikut adalah fungsi pin dari IC DS3231:

1. Pin 1 (32kHz) – pin ini membutuhkan *pullup* resistor eksternal. Ketika diaktifkan, output beroperasi pada salah satu *power supply*. Jika tidak digunakan pin ini dapat dibiarkan terbuka.
2. Pin 2 Vcc – sebagai *power supply* utama. Pin ini harus dipisahkan menggunakan 0,1µF untuk 1,0µF kapasitor. Jika tidak digunakan, pin ini dihubungkan ke *ground*.
3. Pin 3 INT/SQW – aktif-*low Interrupt* atau *Square-Wave Output* ini memerlukan resistor *pullup* eksternal yang terhubung dengan tegangan 5,5 V atau kurang. Pin multifungsi ini ditentukan oleh keadaan bit INTCN di *Control Register* (0Eh). Jika tidak digunakan, pin ini dapat dibiarkan tidak tersambung.
4. Pin 4 RST – menunjukkan status VCC relatif terhadap spesifikasi VPF. Jika VCC berada dibawah VPF, pin RST ini bernilai rendah. Pin ini bisa diaktifkan dengan permintaan tombol tekan *reset*. Pin ini memiliki internal 50kΩ sebagai resistor *pullup* untuk VCC.
5. Pin 5-12 N.C dan Pin 13 GND – *Not Connected* dan *Ground.*
6. Pin 14 VBAT – bila perangkat menggunakan VBAT sebagai sumber daya utama, pin ini harus dipisahkan menggunakan kapasitor 0,1µF. Bila menggunakan perangkat dengan masukan VBAT.
   sebagai sumber daya cadangan, kapasitor tidak diperlukan. Jika VBAT tidak digunakan, hubungkan dengan GND.
7. Pin 15 SDA (*Serial Data Input/Output*) – merupakan pin *input/output* untuk antarmuka serial 2 kawat. Pin SDA membutuhkan resistor *pull-up* eksternal.
8. Pin 16 SCL (*Serial Clock Input/Output*) – pin ini merupakan *input clock* untuk *interface* serial I2C dan digunakan untuk menyinkronkan perubahan data pada *interface* serial.

2.8 LCD [5]

LCD (*Liquid Crystal Display*) bisa memunculkan gambar atau tulisan dikarenakan terdapat banyak sekali titik cahaya (piksel) yang terdiri dari satu buah kristal cair sebagai sebuah titik cahaya. Modul LCD 16x2 ditunjukkan oleh Gambar 2.11 berikut:



Gambar 2.11 LCD 16x2

Karakteristik modul LCD:

a. Terdapat 16 x 2 karakter huruf yang bisa ditampilkan.
b. Terdapat 192 macam karakter.
c. Terdapat 80 x 8 bit *display* RAM (maksimal 80 karakter).
d. Memiliki kemampuan penulisan dengan 8 bit maupun 4 bit.
e. Dibangun dengan *oscillator* lokal.
f. Satu sumber tegangan 5 Volt.
g. Otomatis *reset* saat tegangan dihidupkan.
h. Bekerja pada suhu 0°C sampai 55°C.

Berdasarkan Gambar 2.11 konfigurasi pin dari modul LCD sebagai berikut:

a. VSS : Sebagai *Ground*
b. VCC : *Supply* 5 V DC
c. RS : *Register Select*
   0= *instruction register*
   1= *data register*
d. R/W : Instruksi baca atau tulis
   0= *mode* tulis
   1= *mode* baca
e. E : *Enable*
   Jika logika 0 = *Enable*
   Jika logika 1 = *Disable*
f. DB0 : *Data bit* ke-0
g. DB1 : *Data bit* ke-1
h. DB2 : *Data bit* ke-2
i. DB3 : *Data bit* ke-3
j. DB4 : *Data bit* ke-4
k. DB5 : *Data bit* ke-5
l. DB6 : *Data bit* ke-6

m. DB7 : *Data bit* ke-7

2.9 Modul I2C LCD *Backpack* [6]

Modul I2C LCD backpack adalah modul yang digunakan untuk mengurangi jumlah pin yang digunakan pada koneksi antara Arduino (atau mikrokontroler lainnya) dengan *character* LCD. Modul I2C LCD *backpack* yang digunakan dapat dilihat pada Gambar 2.12.

Gambar 2.12 Modul I2C LCD *Backpack*

Untuk menggunakan modul ini, sebelumnya kita perlu menambahkan fungsi *library New LiquidCrystal* dengan cara mengunduh dan ekstrak *library* tersebut pada sub direktori dimana *library* arduino berada. Setelah itu kita perlu mengganti nama direktori library *LiquidCrystal* yang lama agar tidak bentrok dengan *library* yang baru. Serta kita tidak membutuhkan sub direktori *__MACOSX*, sehingga kita dapat menghapusnya. Selanjutnya, hubungkan arduino dengan I2C LCD *backpack*. Untuk Arduino Leonardo atau Arduino UNO R3, dimana pin I2C berada pada bagian ujung *digital* pin sehingga mudah dikenali (Pin dengan tulisan SDA dan SCL). Sedangkan untuk Arduino UNO sebelum R3, pin I2C berada pada pin analog A4 (SDA) dan A5 (SCL). Untuk memudahkan, berikut ini adalah Tabel 2.1 yang merupakan tabel koneksi antara I2C LCD *backpack*:

Tabel 2.1 Konfigurasi I2C LCD *Backpack* pada Arduino

I2C LCD Backpack	Arduino UNO	Arduino UNO R3 / Leonardo
GND	GND	GND
VCC	5V	5V
SDA	A4	SDA
SCL	A5	SCL

Setelah menghubungkan Arduino dan I2C LCD *backpack*, Anda perlu mengetahui alamat I2C LCD *backpack*. Gunakan *sketch* I2C scanner untuk mengetahui alamat tersebut (*sketch* dapat ditemukan di situs *playground* Arduino). Lakukan kompilasi dan *upload sketch* pada program Arduino IDE. Setelah proses *upload* ke Arduino selesai, lanjutkan dengan menjalankan Arduino serial monitor. Pada *log serial monitor* akan nampak *output* dari pengalamatan I2C. Pengalamatan ini nantinya perlu ditulis pada program Arduino IDE saat akan memrogram LCD menggunakan I2C. Kelebihan dari penggunaan *library New LiquidCrystal* adalah sifatnya fleksibel dan memungkinkan kita untuk menggunakan beberapa fungsi inisialisasi I2C LCD *backpack* sesuai dengan kebutuhan.

2.10 Modul Relai [7]

Relai adalah komponen listrik yang bekerja berdasarkan prinsip induksi medan elektromagnetis. Jika sebuah penghantar dialiri oleh arus listrik, maka di sekitar penghantar tersebut timbul medan magnet. Medan magnet yang dihasilkan oleh arus listrik tersebut selanjutnya diinduksikan ke logam ferromagnetis. *Driver* relai adalah rangkaian elektronika yang bisa mengendalikan pengoperasian sesuatu dari jarak jauh. Dengan relai ini kita bisa mengontrol dan mengopersikan perangkat dari jarak jauh sehingga tak perlu bergeser atau pindah tempat duduk.

Modul relai yang akan digunakan merupakan relai yang dilengkapi *driver* sehingga dapat dihubungkan dengan Arduino UNO. Pada perancangan alat ini digunakan modul relai 5 V dengan 4 *channel.* Modul ini hanya menggunakan sinyal *input* 5 Volt untuk mengontrol. *Driver relay* ini bersifat *active high* sehingga bekerja apabila tegangan *input* 5 Volt dan tidak bekerja apabila tegangan *input* bernilai 0 Volt.

2.11 *Wireless Outdoor* TP-Link TL-WA5210G [8]

Access Point adalah sebuah node yang telah dikonfigurasi secara khusus pada sebuah WLAN *(Wireless Local Area Network). Access Point* sama seperti *hub/switch* pada jaringan kabel bertindak sebagai pusat pemancar dan penerima untuk sinyal-sinyal radio WLAN. *Wireless Access Point* berfungsi untuk menghubungkan beberapa *wireless* klien dengan perangkat jaringan atau komputer-komputer yang tergabung dalam sebuah jaringan. *Access point* memiliki kemampuan untuk mendistribusikan alamat IP ke *wireless* klien secara otomatis.

Jenis *access point* yang digunakan pada alat ini adalah TP-Link TL-WA5210G. Fiturnya memiliki 12dBi *gain antenna*, daya *output* tinggi dan sensitivitas RX yang tinggi secara signifikan dapat memperluas jangkauan transmisi untuk memberikan koneksi nirkabel yang lebih stabil. Sesuai dengan IEEE 802.11b/g, TL-WA5210G dapat menciptakan jaringan nirkabel hingga 54Mbps, sehingga bagus untuk *surfing* internet dan pertukaran *file*.

Access Point (AP) ini adalah sebuah *wireless outdoor* yang memiliki kecepatan hingga 52Mbps dengan jangkauan jarak yang cukup jauh sehingga memungkinkan AP untuk mendeteksi dan menerima sinyal terlemah. AP ini juga memiliki ketahanan terhadap cuaca buruk, sehingga dapat memberikan koneksi nirkabel yang lebih stabil. Untuk AP ini fungsi router perlu ditambahkan antara WAN nirkabel sisi dan sisi *Ethernet* LAN. Oleh karena itu, pelanggan dapat dengan mudah berbagi koneksi cukup dengan *switch* sederhana. TL-WA5210G didukung dengan menggunakan kabel *Ethernet* untuk secara bersamaan mengirim data dan tenaga listrik ke mana pun AP diletakkan sampai 200 kaki. Bentuk fisik AP TP-Link TL-WA5210G dapat dilihat pada Gambar 2.13.

Gambar 2.13 TP-Link TL-WA5210G

2.12 LabVIEW [9]

LabVIEW (singkatan dari *Laboratory Virtual Instrumentation Engineering Workbench*) adalah perangkat lunak komputer untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali instrumentasi serta automasi industri yang pertama kali dikembangkan oleh perusahaan National Instruments pada tahun 1986. Perangkat lunak

ini dapat dijalankan pada sistem operasi Linux, Unix, Mac OS X dan Windows. Produk dari National Instruments ini berupa *software* pengembangan program aplikasi dan *hardware input-output* untuk keperluan akusisi dan pengendalian. Sedangkan perangkat lunak atau software LabVIEW merupakan sebuah bahasa pemograman *graphical* yang menggunakan simbol (ikon) untuk membuat aplikasi. Sementara itu, *Visual Instruments* (VIs) adalah program LabVIEW yang menirukan instrumen sebenarnya dalam bentuk simbol-simbol. Selanjutnya untuk membuat tampilan program aplikasi LabVIEW, digunakan *tools* dan objek. Tampilan aplikasi ini kemudian dikenal dengan jendela *front panel* yang berisikan kode representasi dari simbol sebagai fungsi untuk mengatur objek. Adapun *source code* simbol tersebut ada dalam tampilan jendela *block diagram*. VI berisi tiga komponen sebagai berikut:

- Panel muka—Melayani antarmuka pemakai.
- Diagram Blok—Berisi *source code* grafis yang menggambarkan fungsi-fungsi VI.
- *Icon* dan *Connector Pane*—Mengidentifikasi VI sedemikian rupa sehingga anda dapat menggunakan VI di dalam VI yang lain atau disebut dengan sub VI atau juga disebut dengan *sub routine* didalam program berbasis teks.

Front panel merupakan penghubung (*interface*) antara pengguna (*user*) dengan program aplikasi. Didalam *front panel* terdapat Kontrol (*Input*) dan Indikator (*Output*). Kontrol pada *front panel* dapat berupa knop, tombol, dial, dan lainnya. Sedangkan untuk indikator (*Output*) dapat berupa LED, grafik dan tampilan lainnya. Kontrol menirukan *input instrument* dan menyuplai data ke diagram blok pada VI yang bersangkutan. Indikator menirukan instrumen keluaran dan menampilkan data yang diperoleh atau dihasilkan oleh diagram blok. *Block diagram* merupakan jendela tempat menuliskan perintah dan fungsi, berisikan *source code* berupa simbol-simbol, *node* dan garis sebagai *data flow* untuk mengeksekusi program termasuk kode dari *front panel*. Pada bagian ini kita dapat menambahkan kode menggunakan grafis yang mewakili fungsi untuk mengendalikan obyek-obyek panel muka. Blok diagram berisi *source code* grafis ini. Obyek-obyek panel muka nampak seperti terminal pada diagram blok. Dalam membuat aplikasi VIs, harus diperhatikan tipe data tiap simbol agar *data flow* dapat berjalan semestinya. Tipe data yang tersedia yaitu numerik, *boolean* dan *string*. Tipe data dari sebuah simbol dapat diketahui dari

warna *node* atau warna kabel ketika dihubungkan ke simbol lainnya. Tampilan utama *software* LabVIEW dapat dilihat pada Gambar 2.14.

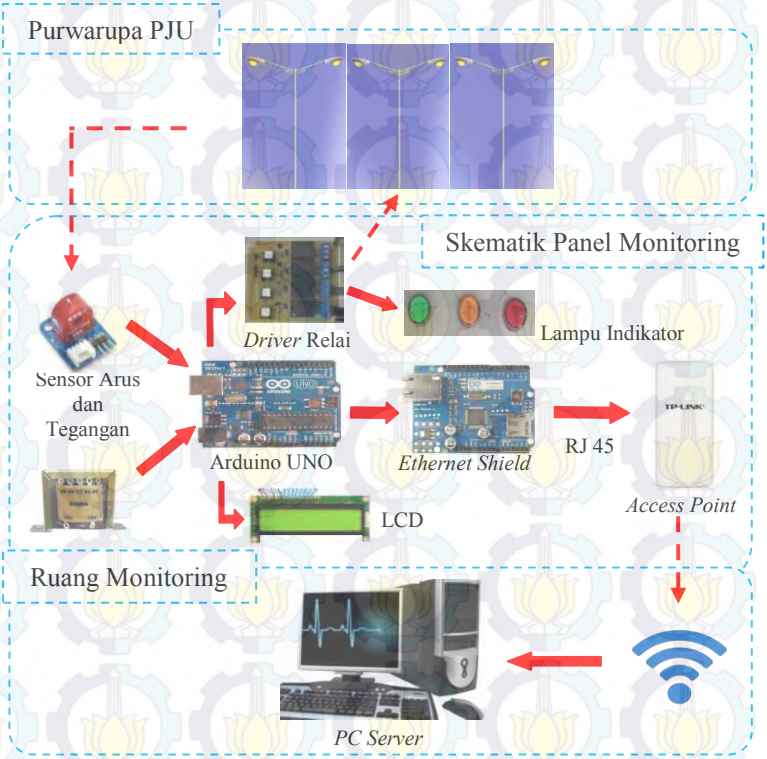
Gambar 2.14 Tampilan Utama *Software* LabVIEW

Halaman ini sengaja dikosongkan

BAB III
PERANCANGAN DAN PEMBUATAN ALAT

Dari teori penunjang yang didapat, selanjutnya dilakukan langkah perancangan dan pembuatan alat yang dibutuhkan pada Tugas Akhir ini. Perancangan yang dilakukan terdiri atas empat bagian, yaitu perancangan sistem keseluruhan, perancangan mekanik, perancangan *hardware* meliputi perancangan rangkaian elektronika, dan perancangan *software* meliputi perancangan program dan *interface* yang digunakan.

3.1 Perancangan Sistem Keseluruhan



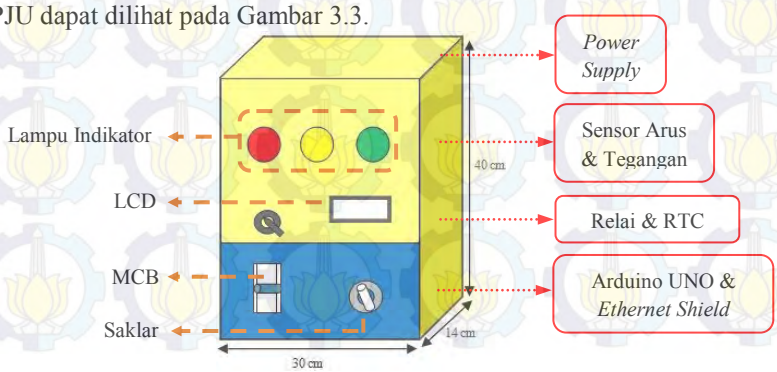
Gambar 3.1 Diagram Fungsional Sistem Alat

Dari Gambar 3.1 Arduino mengirimkan sinyal ke *driver* relai untuk mengaktifkan purwarupa PJU. Kemudian sensor tegangan dan sensor arus akan membaca tegangan dan arus yang terpakai oleh beban lampu pada purwarupa PJU. Data hasil pembacaan dari sensor tegangan dan sensor arus diolah oleh Arduino kemudian di tampilkan pada LCD dan juga disimpan pada *microSD*. Penyimpanan pada *microSD* ditujukan untuk mem*back-up* pembacaan data, apabila *server* mengalami *trouble.* Hasil pembacaan kemudian dikirimkan melalui *Access Point* (AP) yang terhubung dengan *Ethernet shield* melalui kabel RJ45 lalu diterima dan ditampilkan pada PC *server* melalui *interface* LabVIEW. Data yang dikirimkan ke *interface* LabVIEW kemudian disimpan pada PC dengan format *.csv.

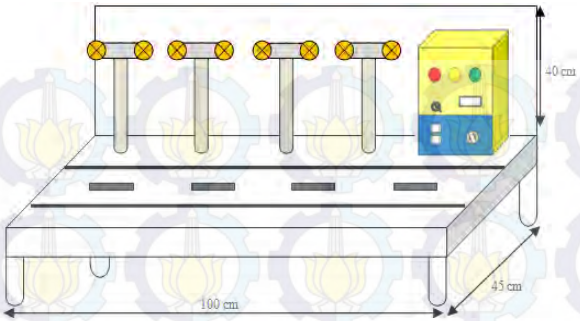
Pada Gambar 3.1 terdapat 3 indikator lampu. Lampu hijau menunjukkan bahwa daya yang terpakai dalam keadaan normal. Apabila pemakaian daya pada beban menengah, maka Arduino mengirimkan sinyal ke *driver* relai untuk mengaktifkan lampu kuning, dan apabila pemakaian daya pada beban penuh, maka Arduino mengirimkan sinyal ke *driver* relai untuk mengaktifkan lampu kuning dan merah.

3.2 Perancangan Mekanik

Perancangan mekanik berupa purwarupa panel monitoring daya PJU dan purwarupa PJU. Pemodelan ruang panel monitoring terbuat dari kotak besi berukuran 30x14x40cm. Purwarupa ruang panel monitoring daya pada alat Tugas Akhir ini dapat dilihat pada Gambar 3.2. Untuk purwarupa PJU kami menggunakan 8 lampu dengan daya 60 Watt yang disusun diatas triplek berukuran 100x45x40 cm. Purwarupa PJU dapat dilihat pada Gambar 3.3.



Gambar 3.2 Purwarupa Panel Monitoring Daya PJU



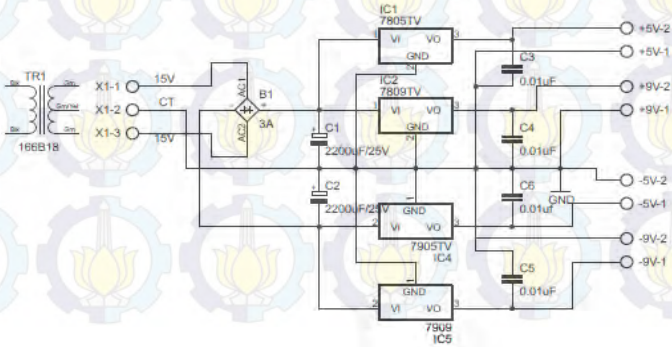
Gambar 3.3 Purwarupa PJU

3.3 Perancangan *Hardware*

Perancangan *hardware* yang dibahas terdiri dari perancangan *power supply,* perancangan sensor tegangan, modul sensor arus, LCD, RTC, *driver* relai, Arduino UNO, dan *setting* sistem komunikasi *WiFi*.

3.3.1 Perancangan *Power Supply*

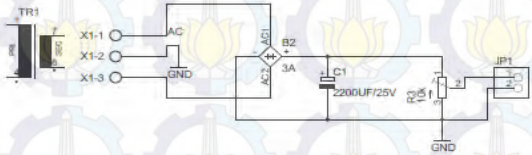
Rangkaian *power supply* yang dirancang adalah *power supply* yang dapat menghasilkan tegangan variasi sebesar ±5V dan ±9V DC. Power supply yang dibuat menggunakan trafo CT 2A. Untuk mendapatkan tegangan ±5V kami menggunakan IC LM7805 dan LM7905, sedangkan untuk mendapatkan tegangan ±9V kami menggunakan IC LM7809 dan LM7909. Alat ini dibuat untuk mensuplai rangkaian LCD, RTC, dan *driver* relai. Rancangan rangkaian *power supply* untuk Tugas Akhir ini dapat dilihat pada Gambar 3.4.



Gambar 3.4 Rangkaian *Power Supply*

3.3.2 Perancangan Sensor Tegangan

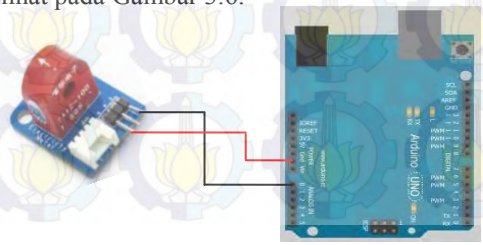
Sensor tegangan merupakan alat yang digunakan untuk mendeteksi besar tegangan yang melalui suatu peralatan listrik. Perancangan sensor tegangan pada Tugas Akhir ini menggunakan trafo nol 500 mA yang bekerja pada tegangan 220V AC. Sensor tegangan ini menggunakan *transformator stepdown* untuk mentransformasikan nilai tegangan sebesar 220V AC menjadi tegangan 6V AC. Kemudian sisi sekunder trafo dihubungkan ke *diode bridge* 3A untuk disearahkan menjadi tegangan DC. Untuk mengurangi *ripple* dari sinyal keluaran *diode bridge*, maka rangkaian sensor tegangan ini diberi rangkaian *filter* kapasitor yang berupa kapasitor 2200uF/25V. Untuk mendapatkan keluaran yang dapat dibaca oleh Arduino (menggunakan referensi 5V), maka rangkaian sensor tegangan ini diberi rangkaian pembagi tegangan yang terdiri dari sebuah *multitune* 10 kOhm yang dapat mengatur tegangan keluaran dari sensor tegangan yang kami buat. Skema rangkaian sensor tegangan dapat dilihat pada Gambar 3.5.



Gambar 3.5 Rangkaian Sensor Tegangan

3.3.3 Modul Sensor Arus

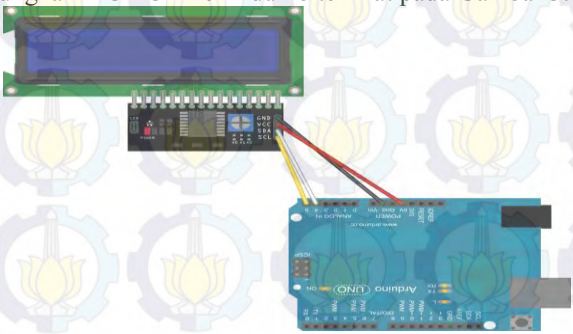
Sensor arus yang digunakan pada Tugas Akhir ini adalah sensor arus tipe CT Ta12-100. Modul ini memiliki 3 pin yaitu pin *Analog Output* (S), pin *Not Connected* (N), dan pin *ground* (G). Pin S dihubungkan ke *port* A0 pada Arduino UNO, dan pin G dihubungkan ke *port* Gnd pada Arduino UNO. Peletakkan sensor arus pada Arduino UNO dapat dilihat pada Gambar 3.6.



Gambar 3.6 Cara Menghubungkan Sensor Arus ke Arduino

3.3.4 Perancangan LCD dengan I2C

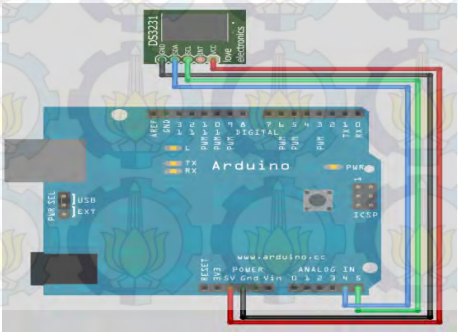
LCD yang digunakan pada alat ini adalah 16x2. Maksudnya adalah 16 kolom dan 2 baris. Tegangan yang diberikan pada LCD tersebut adalah sebesar 5 V DC. LCD ini dapat menampilkan gambar atau tulisan karena didalam LCD terdapat titik-titik cahaya. LCD ini dipakai sebagai tampilan status purwarupa yang dijalankan. Penggunaan I2C LCD *backpack* ini bertujuan untuk mengurangi jumlah pin yang digunakan pada koneksi antara Arduino dengan *character LCD*. Cara Menghubungkan I2C LCD ke Arduino terlihat pada Gambar 3.7.



Gambar 3.7 Cara Menghubungkan I2C LCD ke Arduino

3.3.5 Perancangan RTC

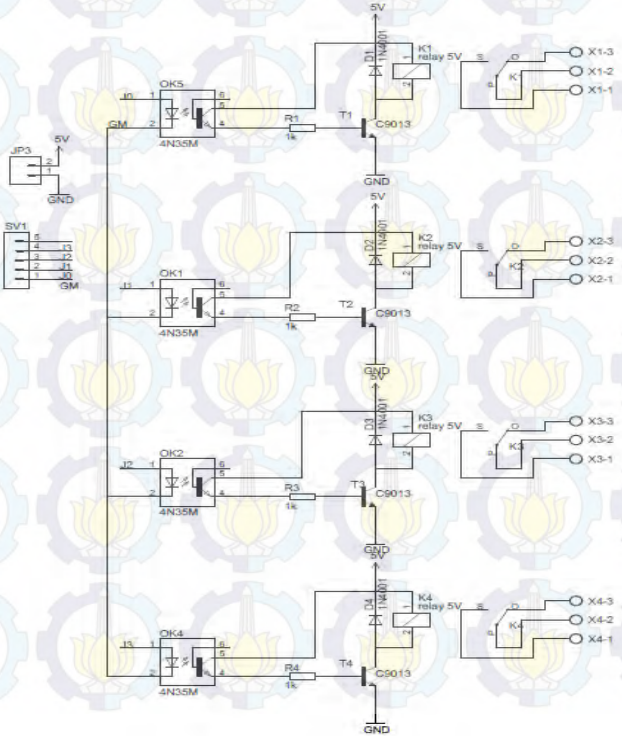
Pada Tugas Akhir ini menggunakan modul RTC DS3231. Cara menghubungkan modul ke Arduino yaitu pin VCC modul dihubungkan ke *port* 5V, pin SDA ke *port* A4, pin SCL ke *port* A5, dan pin GND ke *port* GND. Cara menghubungkan modul ke Arduino dapat dilihat pada Gambar 3.8.



Gambar 3.8 Cara Menghubungkan Modul RTC pada Arduino

3.3.6 Perancangan *Driver* Relai

Pada perancangan alat ini digunakan relai 5 V dengan 4 *channel.* Rancangan relai yang seperti ini dapat digunakan pada Arduino UNO. *Driver* relai ini hanya menggunakan tegangan *input* 5 Volt untuk mengontrol. *Driver* relai ini bersifat *active high* sehingga bekerja apabila *digital output* dari Arduino UNO bernilai 1 (tegangan 5 Volt) dan tidak bekerja apabila *digital output* dari Arduino UNO bernilai 0 (tegangan 0 Volt). Pada rangkaian *driver* relai ini dilengkapi dengan *optocoupler* tipe 4N35. *Optocoupler* ini berfungsi sebagai isolator internal untuk memisahkan Arduino dengan sistem rangkaian yang dikontrol sehingga apabila terjadi lonjakan tegangan yang berada pada beban tidak akan masuk kebagian pengolah data (Arduino). Fungsi dari *driver* relai ini adalah sebagai saklar pada lampu indikator beban. Rangkaian perancangan relai dapat dilihat pada Gambar 3.9.

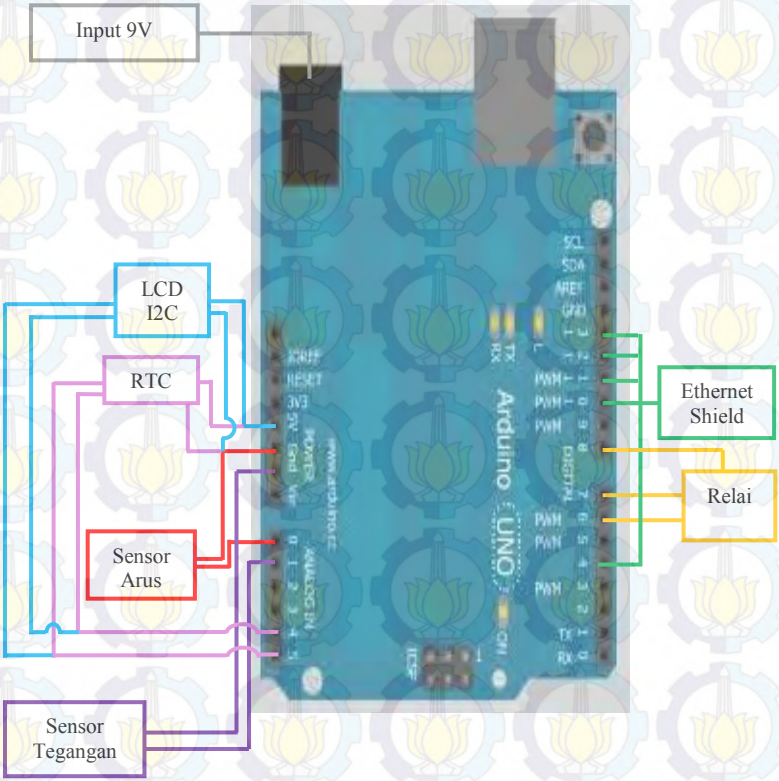


Gambar 3.9 Rangkaian *Driver* Relai

3.3.7 Penggunaan Modul Arduino UNO

Dalam penggunaannya, perangkat keras Arduino UNO ini memakai sistem minimum ATmega328. Data yang diterima akan diproses dan ditransmisikan ke PC melalui komunikasi *WiFi*.

Modul Arduino UNO diberikan tegangan *input* sebesar 9 V DC dari adaptor 9 V;2 A. Untuk penggunaan kaki-kaki pada Arduino UNO, pada *port* A0 digunakan untuk menerima data dari sensor arus, *port* A1 digunakan untuk menerima data dari sensor tegangan. *Port* A4-A5 untuk LCD-I2C dan RTC, *port* D6-D8 untuk relai, dan *port* D4 serta D10-D13 untuk *Ethernet shield*. Penggunaan *port* Arduino UNO dapat dilihat pada Gambar 3.10.

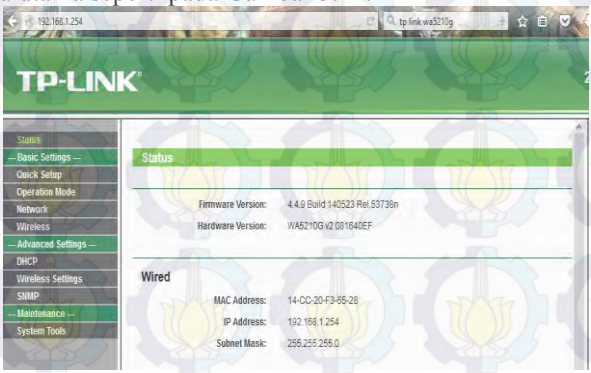


Gambar 3.10 Penggunaan *Port* pada Arduino UNO

3.3.8 *Setting* Sistem Komunikasi

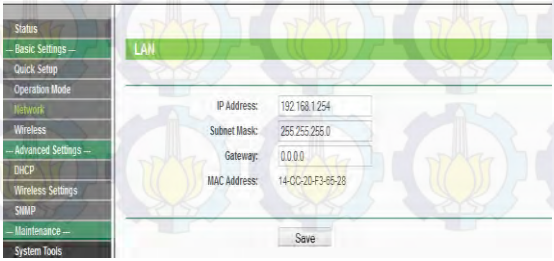
Access Point (AP) adalah sebuah alat yang berfungsi sebagai penghubung beberapa *wireless* klien dengan perangkat jaringan atau komputer-komputer yang tergabung dalam sebuah jaringan. Sebelum digunakan AP harus dilakukan *setting* terlebih dahulu pada *IP address, subnet mask*, dan *default gateway*. Langkah-langkah *setting* awal pada AP adalah sebagai berikut:

1. Buka aplikasi *web browser*
2. Ketikkan IP *default* AP yaitu 192.168.1.254
3. Masukkan *user name*: "admin", *password*: "admin" (*default* AP) kemudian tekan "*enter*". Setelah itu akan muncul tampilan menu utama seperti pada Gambar 3.11.



Gambar 3.11 Tampilan Menu Utama AP

4. Setelah masuk ke menu utama TP-LINK, klik "*Operation Mode*" lalu pilih "*Access Point"* kemudian "*save*".
5. Lalu pilih menu "*Network*" dan lakukan penyetelan seperti Gambar 3.12. Kemudian klik "*save*".



Gambar 3.12 Penyetelan Alamat IP AP

3.4 Perancangan *Software*

Perancangan perangkat lunak ini menggunakan *software* Arduino IDE untuk memprogram Arduino dan *software* LabVIEW 2014 yang digunakan untuk menampilkan hasil pembacaan daya PJU.

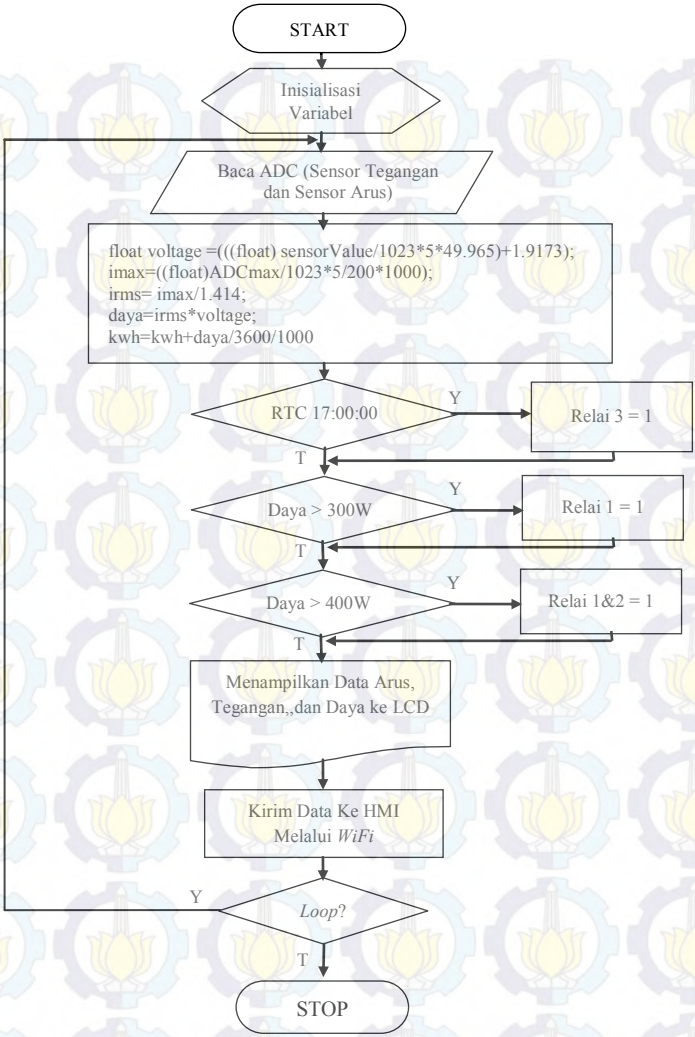
3.4.1 Program pada Arduino IDE

Program pada Arduino UNO menggunakan program Arduino IDE dengan bahasa pemrograman bahasa C. Arduino IDE adalah sebuah *editor* yang digunakan untuk menulis program, meng-*compile*, dan mengunggah ke papan Arduino UNO. Arduino IDE terdiri dari *editor* teks untuk menulis kode, area pesan, *console* teks, *toolbar* dengan tombol-tombol untuk fungsi umum, dan sederetan menu.

Software yang ditulis menggunakan Arduino dinamakan *sketches*. *Sketches* ini ditulis di *editor* teks dan disimpan dengan *file* yang berekstensi .ino. *Editor* teks ini mempunyai fasilitas untuk *cut/paste* dan *search/replace*. Area pesan berisi umpan balik ketika menyimpan dan mengunggah *file*, dan juga menunjukkan jika terjadi *error*. *Flow Chart* pemrogramannya seperti terlihat pada Gambar 3.13.

Dari diagram alir Gambar 3.13, algoritma dari perancangan perangkat lunak untuk Arduino ini adalah:

1. Arduino aktif melakukan inisialisasi variabel data sensor arus dan tegangan.
2. Membaca sensor.
3. Apabila RTC menunjukkan nilai 17:00:00, maka relai 3 menyala.
4. Data sensor dibandingkan dengan kondisi yang telah diatur yaitu 300 Watt.
5. Apabila daya yang diukur sensor melebihi 300 Watt, maka Arduino akan memberi perintah pada *driver* relai untuk mengaktifkan relai 1.
6. Apabila daya yang diukur sensor melebihi 400 Watt, maka Arduino akan memberi perintah pada *driver* relai untuk mengaktifkan relai 1 dan 2.
7. Menampilkan data sensor tegangan, arus, dan daya ke LCD.
8. Data yang terukur dikirimkan ke HMI melalui *WiFi*.
9. Program akan berhenti jika Arduino tidak ter*supply*.



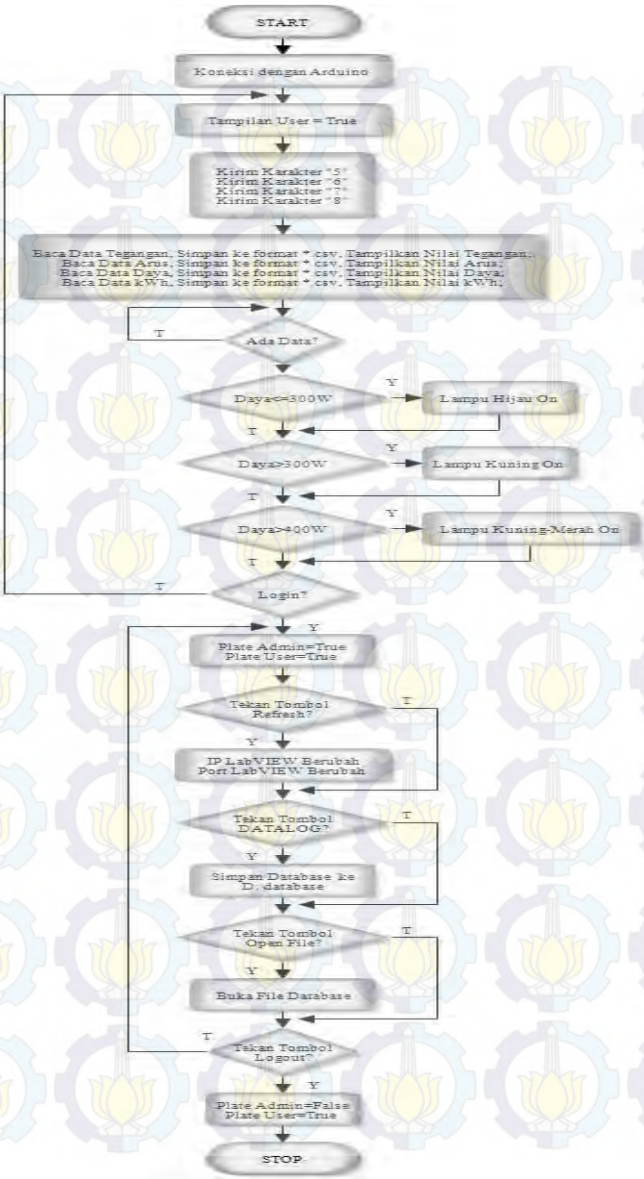
Gambar 3.13 Diagram Alir Arduino IDE

3.4.2 LabVIEW 2014

Dalam pemrograman LabVIEW dikenal istilah *Visual Instruments* (VIs) yang merupakan aplikasi yang menirukan instrumen sebenarnya dalam bentuk simbol-simbol.

LabVIEW yang digunakan pada perancangan HMI monitoring ini adalah LabVIEW versi 2014. Diagram alir untuk tampilan *software* LabVIEW 2014 dapat dilihat pada Gambar 3.14, dengan algoritmanya sebagai berikut :

1. Ketika program dijalankan, maka secara otomatis LabVIEW terhubung dengan Arduino dan tampilan *user* bernilai *true*.
2. Ketika HMI mengirimkan karakter “5”, maka Arduino akan membalas dengan mengirimkan data tegangan. Ketika HMI mengirimkan karakter “6”, maka Arduino akan membalas dengan mengirimkan data arus. Ketika HMI mengirimkan karakter “7”, maka Arduino akan membalas dengan mengirimkan data daya. Dan ketika HMI mengirimkan karakter “8”, maka Arduino akan membalas dengan mengirimkan data kWh.
3. Data yang dikirimkan Arduino selanjutnya ditampilkan dan disimpan ke dalam format *.csv.
4. Jika daya bernilai <=300 W, maka lampu hijau menyala.
5. Jika daya bernilai >300 W, maka lampu kuning menyala.
6. Jika daya bernilai >400 W, maka lampu merah-kuning menyala.
7. Apabila *login username* dan *password* benar maka admin bisa langsung mengakses aplikasi pada *form admin*. Jika tidak melakukan *login* maka *plate user* tetap tampil.
8. Jika tombol *refresh* pada *form admin* ditekan, maka data ID pelanggan berubah.
9. Jika tombol *datalog* ditekan, maka *database* akan tersimpan pada folder D:\database.
10. Jika tombol *open file* ditekan maka *database* yang tersimpan akan terbuka.
11. Jika tombol *logout* ditekan, maka tampilan admin tertutup dan tampilan user tetap berjalan.
12. Sistem ini akan terus berjalan dan akan berhenti ketika tombol stop ditekan.



Gambar 3.14 Diagram Alir LabVIEW

3.4.3 Tampilan Monitoring

Tampilan monitoring yang telah dibuat menggunakan *software* LabVIEW 2014 yang terdiri dari dua *form* yang memiliki tampilan dan fungsi yang berbeda.

a. Kolom *Login.*

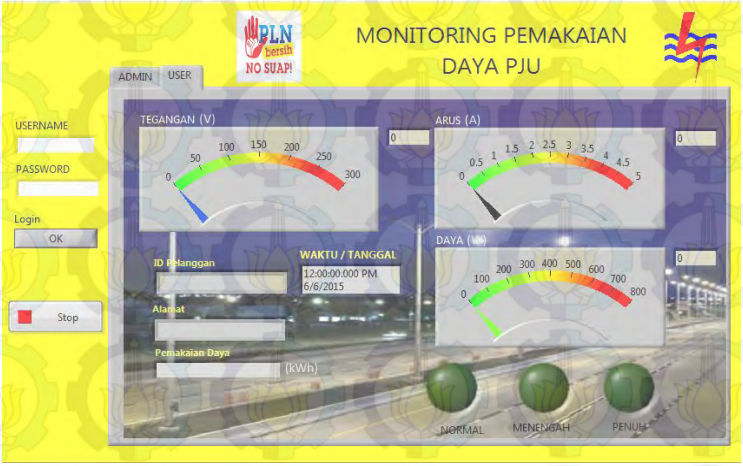
Kolom ini dibuat untuk memberikan keterbatasan akses terhadap aplikasi, sehingga keamanan data yang diakses aplikasi terjaga dengan baik.

b. Tampilan *User*

Tampilan *user* ini berfungsi untuk menampilkan data ID pelanggan, alamat, waktu, tanggal, nilai tegangan, nilai arus, nilai daya, pemakaian daya, dan lampu indikator.

c. Tampilan *Admin*

Tampilan *admin* ini berfungsi untuk menampilkan data ID pelanggan, alamat, waktu, tanggal, nilai tegangan, nilai arus, nilai daya, pemakaian daya, lampu indikator, dan mengganti IP serta *port* PJU, dan mengakses *database*.



Gambar 3.15 Tampilan Monitoring Pemakaian Daya PJU

Halaman ini sengaja dikosongkan

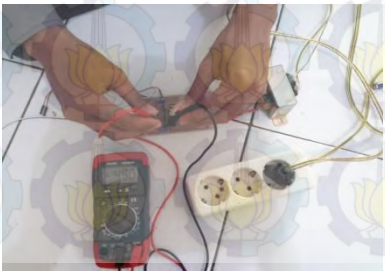
BAB IV
PENGUJIAN DAN ANALISA DATA

Bab ini membahas tentang pengujian alat yang dibuat, adapun tujuan pengujian tersebut adalah untuk mengetahui hasil dari pembuatan alat dalam kegiatan Tugas Akhir ini. Pengujian alat ini dilakukan tiap bagian dan secara keseluruhan. Gambar 4.1 berikut ini menampilkan *hardware* secara keseluruhan.

Gambar 4.1 Tampilan Fisik *Hardware* Keseluruhan

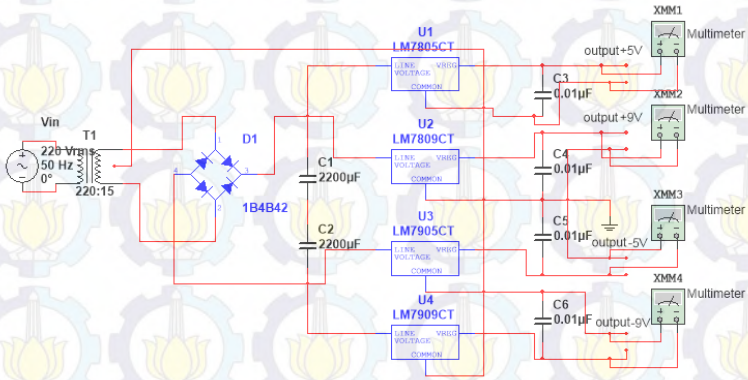
4.1 Pengujian *Power Supply*

Power supply diperlukan untuk mencatu daya *driver relai*, LCD, dan RTC. *Driver relai,* RTC, dan LCD memerlukan tegangan sebesar 5V DC. Gambar 4.2 berikut merupakan gambar pengujian *power supply*.



Gambar 4.2 Pengujian *Power Supply*

Pengujian dilakukan dengan cara menyalakan *power supply* dengan tegangan *input* yang berbeda-beda yakni 200VAC, 220VAC dan 240VAC melalui variak, kemudian mengukur hasil tegangan keluarannya menggunakan multimeter. Skematik pengujian *power supply* serta hasil pengujian dari *power supply* dapat dilihat pada Gambar 4.3 dan Tabel 4.1.



Gambar 4.3 Skematik Pengujian *Power Supply*

Tabel 4.1 Hasil Pengujian *Power Supply*

Output	Vin (AC)	Vout Multimeter (DC)	Kesalahan
+ 5V DC	200V	4,92V	1,6%
	220V	4,92V	1,6%
	240V	4,92V	1,6%
+ 9V DC	200V	8,96V	0,4%
	220V	8,96V	0,4%
	240V	8,96V	0,4%
- 5V DC	200V	-5,03V	0,59%
	220V	-5,03V	0,59%
	240V	-5,03V	0,59%
-9V DC	200V	-9,01V	0,1%
	220V	-9,01V	0,1%
	240V	-9,01V	0,1%

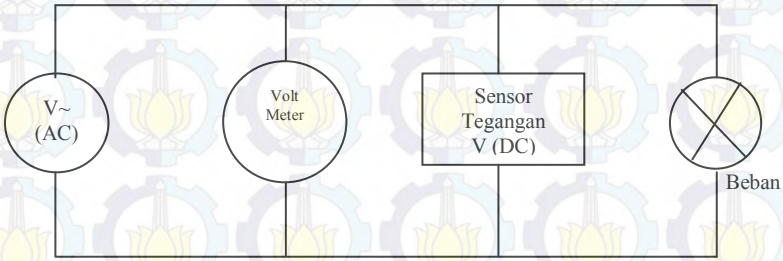
Persentase kesalahan pengukuran sensor tegangan didapat dari Persamaan 4.1 berikut:

$$\frac{-}{} \qquad \text{.....} \quad (4.1)$$

Dari Tabel 4.1 bisa disimpulkan bahwa *power supply* layak untuk digunakan.

4.2 Pengujian Sensor Tegangan

Pada pengujian sensor tegangan ini kami menggunakan perubahan tegangan dari variak. *Input* variak bersumber dari tegangan PLN dan *output* dapat diubah–ubah. Diagram pengujian sensor dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram Pengujian Sensor Tegangan

Dalam pengujian sensor, diambil berbagai perubahan tegangan *output* variak. Hasil pengujian sensor dapat dilihat pada Tabel 4.2.

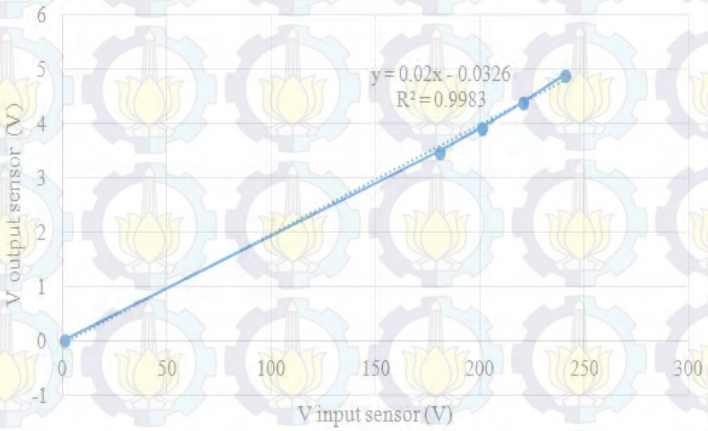
Tabel 4.2 Hasil Pengujian Sensor Tegangan

Beban	V*input* (AC)	V*output* (DC)
100 Watt	180 V	3,47 V
100 Watt	200 V	3,90 V
100 Watt	220 V	4,38 V
100 Watt	240 V	4,87 V

Dari data pengujian dan pengukuran tegangan *output* dari sensor tegangan, dapat dilihat bahwa *output* cukup linier sehingga sensor tegangan ini dapat digunakan. Dari hasil tersebut dapat dibuat suatu persamaan karakteristik sensor tegangan. Dari data pengujian sensor tegangan dapat diketahui rumus kelinierannya dengan menggunakan rumus *sloop* dan *intercept* pada *Ms Excel.* Persamaan 4.2 berikut adalah hasil yang didapat beserta grafik karakteristiknya:

$$y = 0{,}02x - 0{,}0326 \quad (4.2)$$

Keterangan :
x = Nilai tegangan *input* sensor tegangan (VAC)
y = Nilai tegangan *output* sensor tegangan (VDC)

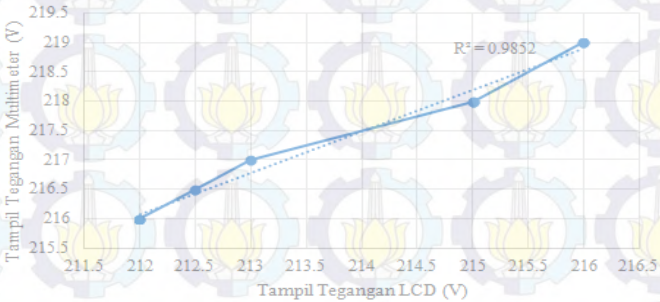


Gambar 4.5 Karakteristik Sensor Tegangan

Seperti pada Gambar 4.5, nilai tegangan *output* bergerak linier sesuai dengan kenaikan tegangan *input* dengan koefisien korelasi sebesar 0,9983. Jadi sensor layak digunakan sebagai sensor tegangan. Berikut adalah tabel pengujian persentase kesalahan pengukuran sensor tegangan beserta grafiknya:

Tabel 4.3 Persentase Kesalahan Pengukuran Sensor Tegangan

Beban	Tegangan Multimeter	Tegangan Tampil LCD	Kesalahan (%)
100 W	219 V	216 V	1,37
200 W	219 V	216 V	1,37
260 W	218 V	215 V	1,38
320 W	218 V	215 V	1,38
380 W	216 V	213 V	1,39
440 W	217 V	214 V	1,38
500 W	216 V	212 V	1,85
560 W	216 V	212 V	1,85
Rata-Rata Kesalahan			1,5



Gambar 4.6 Kesalahan Pengukuran Sensor Tegangan

Persentase kesalahan pengukuran sensor tegangan didapat dari Persamaan 4.3 berikut:

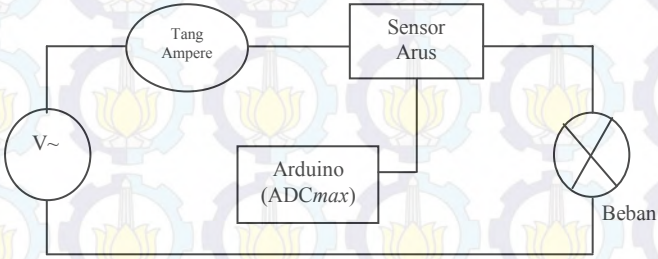
$$\text{-} \qquad \text{.....(4.3)}$$

Dari Tabel 4.3 dan Gambar 4.6 bisa disimpulkan bahwa sensor tegangan layak untuk digunakan karena rata-rata kesalahan dari sensor tegangan adalah 1,5% dengan koefisien korelasi sebesar 0,9852.

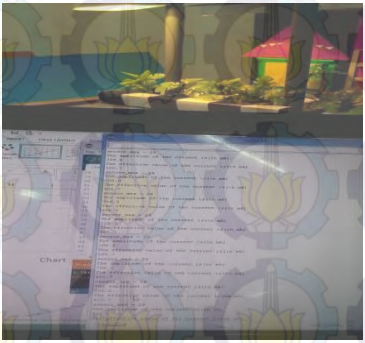
4.3 Pengujian Sensor Arus

Pada pengujian sensor arus ini kami menggunakan perubahan beban lampu pijar 60 Watt sebanyak 8 buah dengan tegangan *input* tetap

yaitu 220V AC. Diagram pegujian sensor arus dapat dilihat pada Gambar 4.7. Dalam pengujian sensor dilakukan dengan merubah beban lampu, lalu melakukan pembacaan ADC*max* dari sensor arus melalui serial monitor Arduino seperti pada Gambar 4.8. Hasil pengujian sensor arus dapat dilihat pada Tabel 4.4.



Gambar 4.7 Diagram Pengujian Sensor Arus



Gambar 4.8 Pengujian Sensor Arus

Tabel 4.4 Hasil Pengujian Sensor Arus

Beban	Arus (Tang Ampere)	ADC *Max*
120 W	0,47 A	29
240 W	0,96 A	59
360 W	1,43 A	89
480 W	1,9 A	118

Dari Tabel 4.4 tersebut dapat dibuat suatu perumusan untuk mendapatkan nilai V*max* dari ADC*max* melalui Persamaan 4.4.

$$\frac{\quad}{\quad} \qquad \text{.....(4.4)}$$

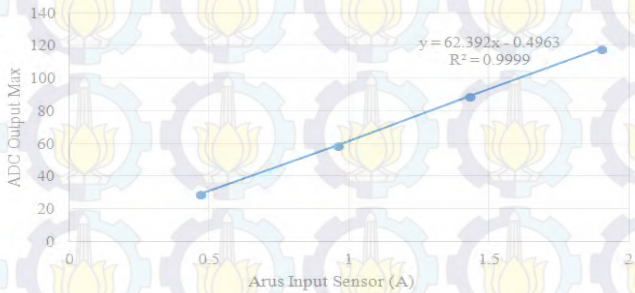
Selanjutnya nilai V*max* tersebut digunakan untuk mendapatkan nilai dari I*max* sensor. Berdasarkan hukum Ohm, I*max* didapatkan dari Persamaan 4.5 berikut:

$$\frac{\quad}{\quad} \qquad \text{.....(4.5)}$$

Nilai R adalah nilai resistor yang terdapat pada modul sensor arus, yaitu 200 Ohm. Dari Persamaan 4.4 dan 4.5 didapatkan perumusan penghitungan I*max* dan Irms untuk program Arduino seperti Persamaan 4.6 dan Persamaan 4.7.

$$\left(\left(\frac{\quad}{\quad}\right)\right) \qquad \text{.....(4.6)}$$

$$\left(\frac{I}{\quad}\right) \qquad \text{.....(4.7)}$$



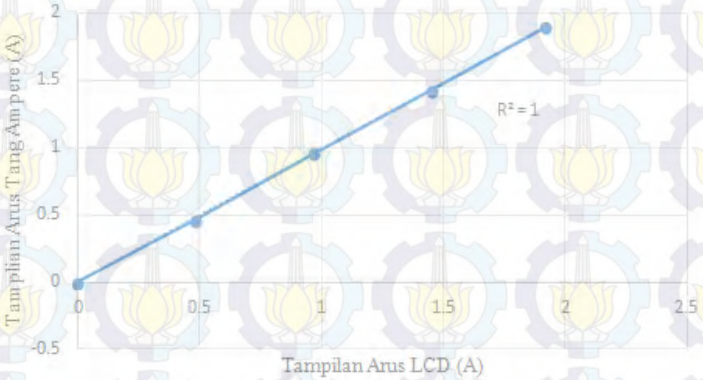
Gambar 4.9 Karakteristik Sensor Arus

Dari Gambar 4.9 dapat dilihat bahwa output dari sensor arus linier sesuai dengan kenaikan arus *input* dengan koefisien korelasi sebesar 0,9999. Berikut adalah tabel pengujian persentase kesalahan pengukuran sensor arus beserta grafiknya:

Tabel 4.5 Persentase Kesalahan Pengukuran Sensor Arus

Beban	Arus (Tang Ampere)	Arus (Tampilan LCD)	Kesalahan (%)
120 W	0,47 A	0,48 A	2,08
240 W	0,96 A	0,97 A	1,03

Beban	Arus (Tang Ampere)	Arus (Tampilan LCD)	Kesalahan (%)
360 W	1,43 A	1,45 A	1,38
480 W	1,9 A	1,92 A	1,04
Rata-Rata Kesalahan			1,38



Gambar 4.10 Kesalahan Pengukuran Sensor Arus

Persentase kesalahan pengukuran sensor arus didapat dari Persamaan 4.8 berikut:

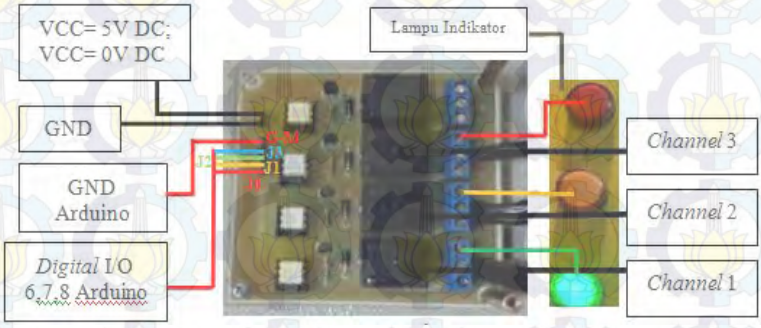
$$\frac{\text{- Tang}}{} \quad \text{.....(4.8)}$$

Dari Tabel 4.5 dan Gambar 4.10 bisa disimpulkan bahwa sensor arus layak untuk digunakan karena rata-rata kesalahan dari sensor arus adalah 1,38% dengan koefisien korelasi sama dengan 1.

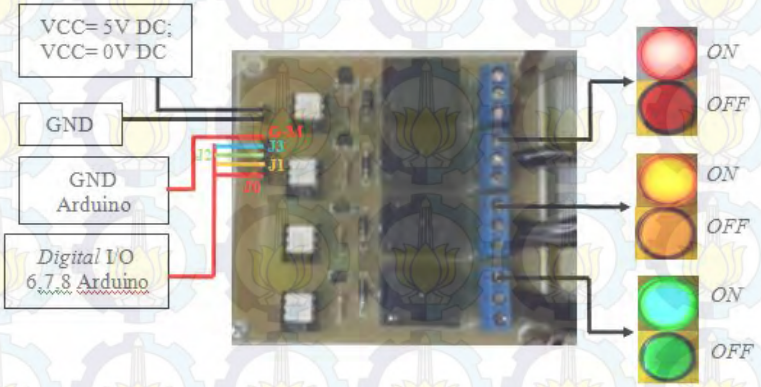
4.4 Pengujian Relai

Pada pengujian ini, kami menguji *channel* relai yang digunakan dengan menghubungkan pin VCC dan GND pada *power supply* +5V. Pin G-M dihubungkan pada *ground* Arduino dan pin J0-J2 dihubungkan pada pin D6-D8 pada Arduino. Pada saat pengujian, ketika pin J0-J2 (relai *channel* 1,2,3) diberi logika 1 (5V) maka relai aktif ditandai dengan lampu indikator menyala, dan jika pin J0-J2 diberi logika 0 (0V)

maka relai tidak aktif ditandai dengan lampu indikator mati. Skema pengujian *driver* relai dan kondisi lampu indikator saat pengujian dapat dilihat pada Gambar 4.11 dan Gambar 4.12. Dengan pengujian ini, kami dapat mengetahui bahwa relai yang telah dibuat dapat berfungsi.



Gambar 4.11 Skema Pengujian *Driver* Relai



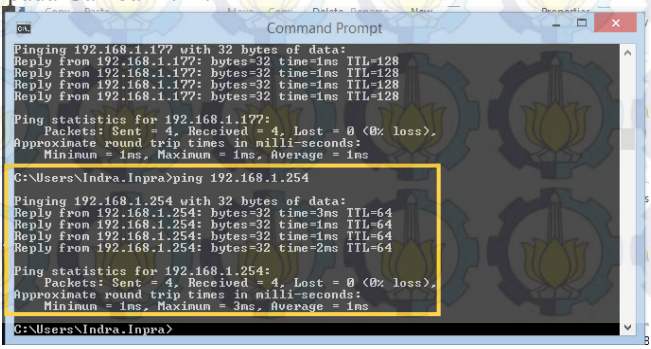
Gambar 4.12 Kondisi Lampu Indikator saat Pengujian *Driver* Relai

4.5 Pengujian *WiFi* TP-LINK

Untuk mengetahui apakah *WiFi* TP-LINK dapat digunakan perlu dilakukan suatu pengujian. Pengujian meliputi pengujian koneksi dan pengujian pegiriman data *WiFi*. Pengujian koneksi dilakukan untuk melihat apakah *WiFi* dapat terhubung dengan komputer *server* atau tidak. Pengujian koneksi dapat dilihat pada Gambar 4.13.

Gambar 4.13 Koneksi *WiFi* TP-LINK

Pada kotak dialog seperti Gambar 4.13 di atas, kita pilih *WiFi* yang akan kita gunakan kemudian klik pada tombol *connect* dan tunggu hingga status *WiFi* berubah menjadi *connected* yang menandakan komputer sudah terhubung dengan *WiFi*. Setelah terkoneksi dengan *WiFi* TP-LINK perlu diuji apakah *WiFi* sudah benar-benar terkoneksi atau belum. Uji koneksi dapat menggunakan pengujian komunikasi juga dapat dilakukan dengan cara menggunakan perintah *cmd*. Pengujian ini disebut juga sebagai *Ping Test*. *Ping Test* dilakukan dengan cara membuka kotak dialog *run* dan mengetikkan perintah *cmd*. Setelah itu akan muncul kotak dialog dan kita ketikkan perintah *Ping* 192.168.1.254 (IP *WiFi* yang sudah di-*setting* sebelumnya. Apabila koneksi berhasil, *WiFi* akan memberikan respon *reply*. Hasil *Ping Test* lewat *cmd* bisa dilihat pada Gambar 4.14.



Gambar 4.14 Hasil *Ping Test*

Pengujian *WiFi* ini bertujuan untuk menguji kemampuan dari *WiFi* dalam mengirimkan data untuk variasi jarak yang diberikan. Hasil uji dari *WiFi* ini dapat dilihat pada Tabel 4.6.

Tabel 4.6 Hasil Pengujian *WiFi*

Jarak PC-*WiFi* (Meter)	Ketinggian *WiFi*-Tanah (Meter)	Penghalang	Hasil
1	0	Tidak Ada	Terhubung
10	0	Tidak Ada	Terhubung
30	0	Tidak Ada	Terhubung
80	0	Tembok	Terhubung
100	0	Tembok	Terhubung
300	0	Tembok	Terputus
300	2	Tidak ada	Terhubung
500	2	Tidak ada	Terhubung
700	2	Tidak ada	Terputus

Jadi, *WiFi* yang digunakan dapat mengirimkan data hingga jarak 500 m dalam kondisi tanpa penghalang.

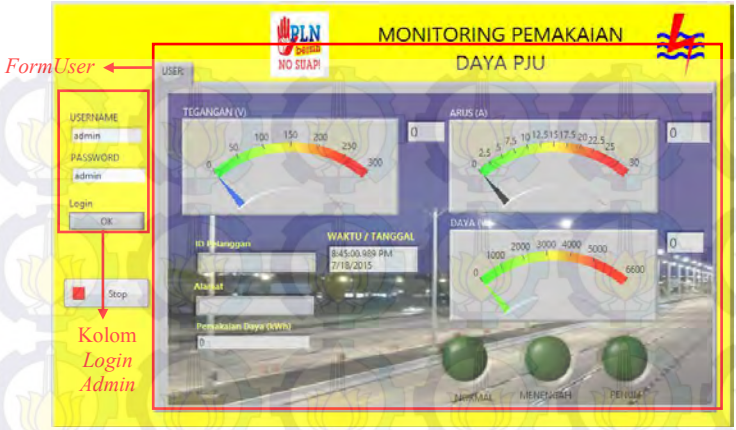
4.6 Pengujian HMI

HMI dibuat menggunakan *software* LabVIEW 2014. Pengujian untuk HMI ini meliputi dari kolom *login*, tampilan monitoring, dan *database*.

4.6.1 Pengujian Kolom *Login*

Kolom *login* adalah sistem keamanan dalam sebuah aplikasi perusahaan yang merupakan salah satu prioritas utama. Hal ini diperlukan untuk mencegah jika ada seseorang yang tidak bertanggung jawab menyalahgunakan sistem tersebut.

Untuk itulah dibutuhkan sebuah halaman *login* pada HMI yang ditampilkan. Pemberian *username* dan *password* bisa meminimalisir orang-orang yang tidak mempunyai kewenangan untuk berbuat sesuatu yang merugikan orang banyak. Untuk mengakses *form admin*, kita perlu kk “d ” d k *username* dan *password* seperti pada Gambar 4.15.



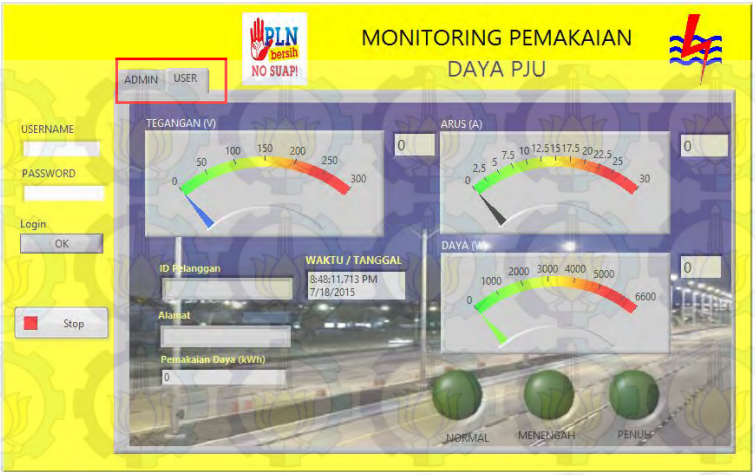
Gambar 4.15 Tampilan Kolom *Login Admin* dan *Form User*

4.6.2 Pengujian Tampilan Monitoring

Tampilan LabVIEW terdiri dari *form admin* dan *user*. *Form us*er merupakan tampilan awal dari HMI. Tampilan *form user* dapat dilihat pada Gambar 4.15.

Setelah berhasil melakukan proses *login,* maka proses selanjutnya ialah masuk ke tampilan *form admin* monitoring pemakaian daya PJU. Tampilan *form admin* dapat dilihat pada Gambar 4.16 dan Gambar 4.17.

Gambar 4.16 Tampilan 1 *Form Admin*



Gambar 4.17 Tampilan 2 *Form Admin*

4.6.3 Pengujian *Database*

Seluruh proses akuisisi data akan masuk ke dalam sebuah *database*. *Database* ini disimpan pada PC dengan format *.csv. Pengujian *database* dilakukan untuk memastikan bahwa penyimpanan data ke dalam *database* tersebut berjalan dengan baik. Dalam *database* ini dapat dilihat rekaman hasil monitoring dengan melihat tanggal dan waktu pengambilan data arus, tegangan, daya, dan pemakaian daya, serta status indikator lampu. Tampilan *database* jika dibuka dengan aplikasi *Ms.Excel* pada PC dapat dilihat pada Gambar 4.18.

	A	B	C	D	E	F	G	H
1	Time	Tegangan (V)	Arus (A)	Daya (W)	Pemakaian Daya (kWh)	HIJAU	KUNING	MERAH
2	7/7/2015 17:59:57.694	219.787	0.478	104.924	0.09943	1	0	0
3	7/7/2015 18:59:59.941	218.811	0.495	108.429	0.20004	1	0	0
4	7/7/2015 20:00:03.944	218.078	0.495	107.825	0.30224	1	0	0
5	7/7/2015 21:00:19.943	208.31	1.451	301.866	0.44252	0	1	0
6	7/7/2015 22:00:05.228	199.03	2.065	411.054	0.72588	0	1	1

Kondisi Beban Normal

Kondisi Beban Menengah

Kondisi Beban Penuh

Gambar 4.18 Pengujian *Database* pada PC

4.7 Pengujian Sistem Secara Keseluruhan

Pengujian sistem secara keseluruhan merupakan gabungan dari pengujian unit yang saling berhubungan. Pengujian ini dilakukkan untuk mengetahui integritas antar unit dalam satu sistem dengan tujuan sistem yang dibangun sesuai dengan perancangan sistem. Pengujian keseluruhan meliputi pengujian LCD, pengujian komunikasi, dan pengujian HMI. Gambar 4.19 berikut adalah pengujian alat secara keseluruhan.

Gambar 4.19 Pengujian Alat Keseluruhan

Pengujian dimulai dengan menyalakan sistem dan melihat tampilan LCD apakah berfungsi dengan baik atau tidak. Ketika ada permasalahan pada sistem, pada LCD akan muncul tampilan acak. Sedangkan apabila sistem dapat bekerja dengan baik maka LCD akan menunjukkan tampilan awal seperti pada Gambar 4.20.

Gambar 4.20 Tampilan Awal LCD

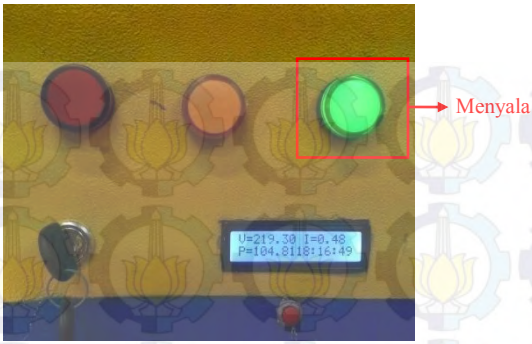
Setelah muncul tampilan pada Gambar 4.20 maka tampilan LCD akan berubah seperti pada Gambar 4.21. Penyetelan jam pada awal tampilan adalah 16:59:00. Penyetelan ini dilakukan secara *software* melalui Arduino IDE.

Gambar 4.21 Tampilan Monitoring pada LCD

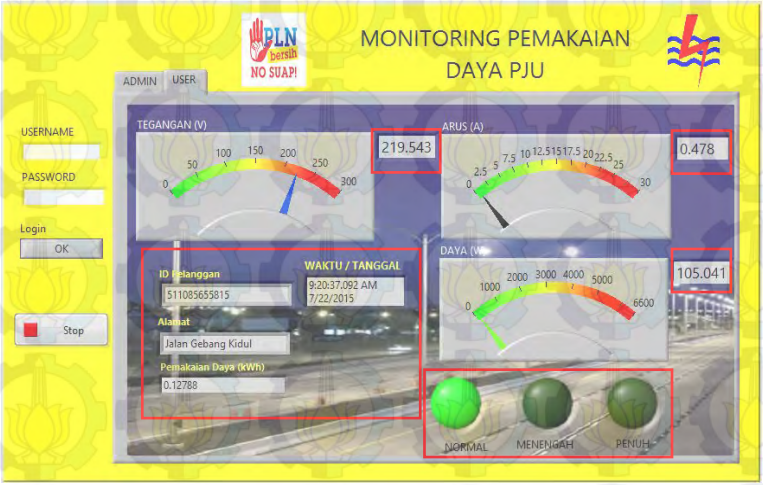
Tepat ketika jam pada LCD menunjukkan pukul 17:00:00 maka secara otomatis lampu pada dua tiang purwarupa PJU menyala (lihat Gambar 4.22). Penyalaan lampu disetel untuk menyala selama 5 jam sehingga pada pukul 22:00:00 (lihat Gambar 4.34) maka lampu pada purwarupa PJU akan mati secara otomatis.

Gambar 4.22 Penyalaan Purwarupa PJU

Selanjutnya menguji HMI secara keseluruhan, dengan melakukan pengujian dalam tiga kondisi yakni saat pemakaian daya pada kondisi normal, saat kondisi pemakaian daya menengah, dan pada kondisi pemakaian daya berlebih (penuh). Pada kondisi pertama ketika dua tiang purwarupa PJU menyala, nilai daya berada pada keadaan normal, lampu indikator hijau pada panel akan menyala dan nilai arus, tegangan, dan daya akan tertampil pada LCD. Kemudian HMI juga menampilkan nilai arus, tegangan, daya, dan pemakaian daya serta indikator lampu hijau menyala. Kondisi pertama ini ditunjukkan pada Gambar 4.23 dan Gambar 4.24.

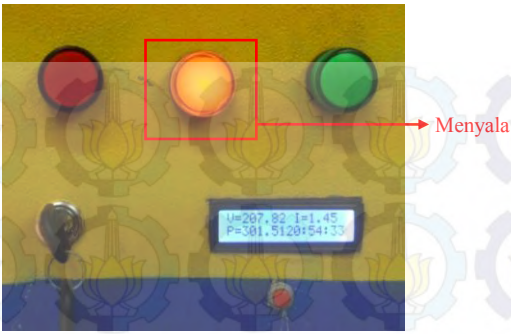


Gambar 4.23 Tampilan LCD dan Indikator pada Kondisi Pertama

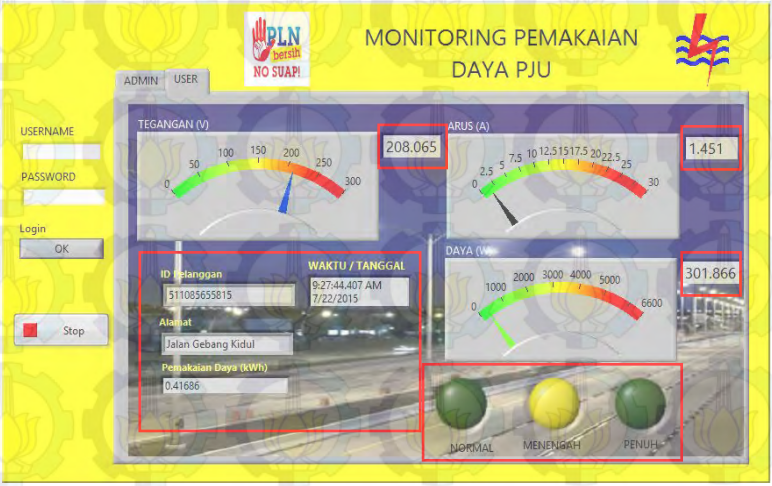


Gambar 4.24 Tampilan HMI pada Kondisi Pertama

Pada kondisi kedua yaitu ketika saklar tiang ketiga dinyalakan, nilai daya berada pada keadaan beban menengah (daya > 300 Watt). Pada kondisi ini lampu indikator kuning pada panel akan menyala dan nilai arus, tegangan, dan daya akan tertampil pada LCD kemudian HMI juga menampilkan nilai arus, tegangan, daya, dan pemakaian daya serta indikator lampu kuning menyala. Kondisi kedua ini ditunjukkan pada Gambar 4.25 dan Gambar 4.26.

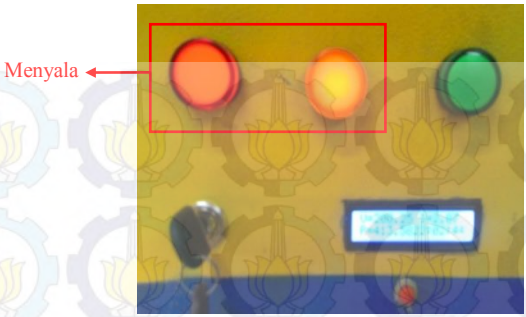


Gambar 4.25 Tampilan LCD dan Indikator pada Kondisi Kedua

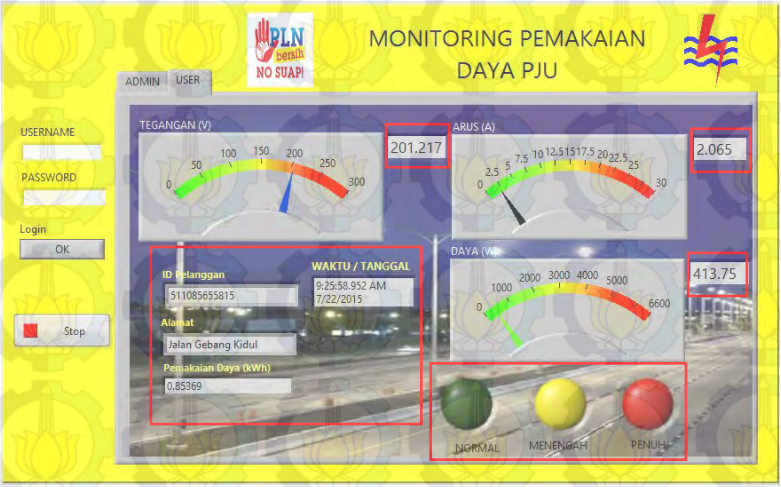


Gambar 4.26 Tampilan HMI pada Kondisi Kedua

Pada kondisi ketiga yaitu ketika saklar tiang keempat dinyalakan, nilai daya berada pada keadaan beban penuh (daya > 400 Watt). Pada kondisi ini lampu indikator kuning dan merah pada panel akan menyala serta nilai arus, tegangan, dan daya akan tertampil pada LCD kemudian HMI juga menampilkan nilai arus, tegangan, daya, dan pemakaian daya serta indikator lampu kuning-merah menyala. Kondisi ketiga ini ditunjukkan pada Gambar 4.27 dan Gambar 4.28.



Gambar 4.27 Tampilan LCD dan Indikator pada Kondisi Ketiga



Gambar 4.28 Tampilan HMI pada Kondisi Ketiga

Seluruh data yang terukur, kemudian disimpan dalam *database* pada PC dan juga *microSD*. *Database* yang tersimpan pada PC dapat dilihat pada Gambar 4.18 dan Gambar 4.29 berikut menunjukkan *database* yang tersimpan pada *microSD*.

	A	B	C	D	E	F
1	Time	Tegangan (V)	Arus (A)	Daya (W)	Pemakaian Daya (kWh)	Indikator Beban
2	7/7/2015 17:00	0	0	0	0	HIJAU
3	7/7/2015 18:00	219.543	0.478	104.924	0.09952	HIJAU
4	7/7/2015 19:00	219.055	0.495	108.429	0.20004	HIJAU
5	7/7/2015 20:00	218.078	0.495	107.825	0.30224	HIJAU
6	7/7/2015 21:00	208.554	1.451	302.575	0.44093	KUNING
7	7/7/2015 22:00	199.03	2.065	411.054	0.72588	KUNING MERAH

Normal

Menengah

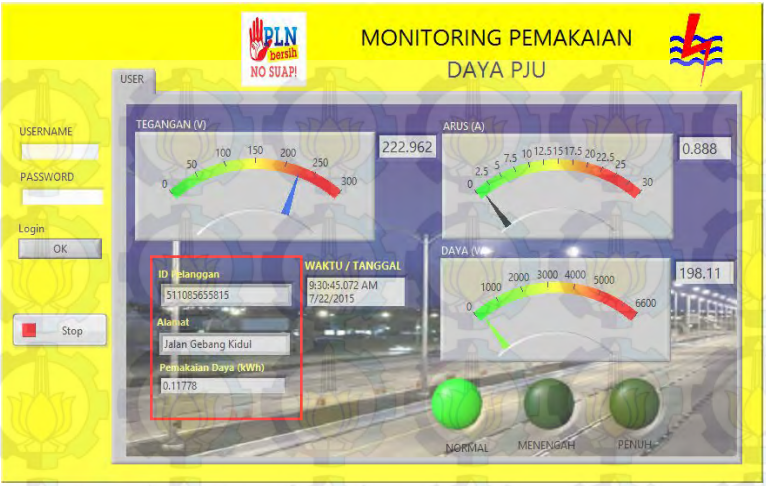
Penuh

Gambar 4.29 Tampilan *Database* yang Tersimpan pada *MicroSD*

Selain itu, kami juga menguji *form* monitoring ini untuk fungsi multi-jaringan. Pengujian dapat dilihat pada Gambar 4.30, Gambar 4.31, Gambar 4.32, dan Gambar 4.33.

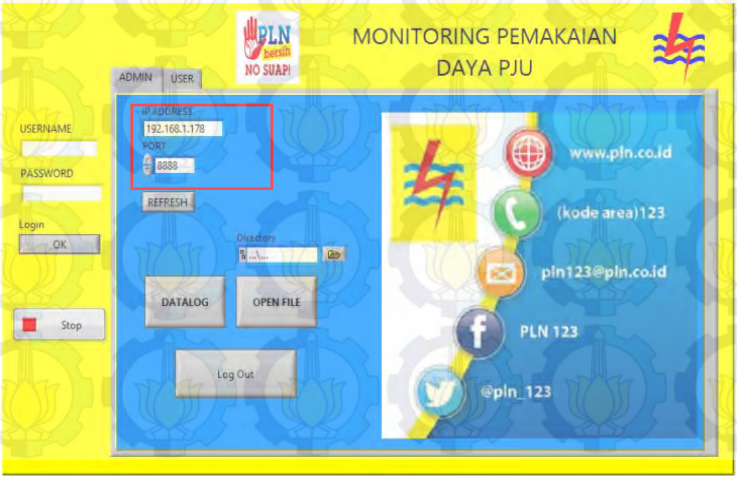
Gambar 4.30 *Login* ke Data Pelanggan Jalan Gebang Kidul

Setelah mengetikkan alamat IP 192.168.1.177, klik tombol "*refresh*". k *form admin* menampilkan data pelanggan di Jalan Gebang Kidul seperti Gambar 4.31.

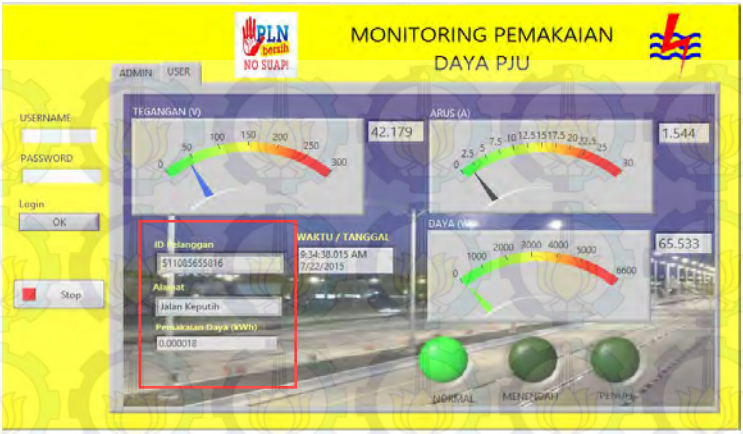


Gambar 4.31 Data Pelanggan Jalan Gebang Kidul

Apabila mengetikkan alamat IP 192.168.1.178 seperti Gambar 4.32 lalu k b "*refresh*" k *form admin* menampilkan data pelanggan di Jalan Keputih seperti pada Gambar 4.33.



Gambar 4.32 *Login* ke Data Pelanggan Jalan Keputih



Gambar 4.33 Data Pelanggan Jalan Keputih

Gambar 4.34 Kondisi Purwarupa PJU Mati

4.8 Analisa Keseluruhan

Dari pengujian sistem secara keseluruhan, semua fungsi sistem dapat bekerja. Sensor mampu merespon terhadap perubahan kondisi beban seperti yang dilakukan dalam prosedur pengujian dan fungsi alat, LCD dapat menampilkan hasil pembacaan sensor. Data yang terukur dapat dikirimkan ke *server* dan HMI dapat menampilkan data sensor dan menyimpannya ke dalam *database* pada PC dan *microSD*. Sistem monitoring ini dapat digunakan untuk mengakses data pelanggan PJU lain dengan menggunakan IP yang berbeda. Namun masih terdapat kelemahan yakni alat ini masih belum dilengkapi dengan sensor faktor daya. Dengan demikian dapat disimpulkan bahwa alat dapat berfungsi normal.

Halaman ini sengaja dikosongkan

BAB V
PENUTUP

Setelah melakukan perencanaan dan pembuatan alat serta pengujian dan analisis, maka dapat ditarik kesimpulan dan saran dari kegiatan yang telah dilakukan.

5.1 Kesimpulan

Dari Tugas Akhir yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut:

1. Alat yang dibuat dapat memonitoring pemakaian daya PJU secara otomatis dan *online* melalui PC *server*.
2. Data hasil monitoring dapat disimpan ke dalam *database* pada PC dan *microSD* berupa *file* dengan format **.csv* untuk mempermudah perekapan data pemakaian daya PJU.
3. Hasil kalibrasi sensor arus dan tegangan yang digunakan menunjukan bahwa sensor tersebut layak digunakan. Karena, hasil persentase kesalahan rata-rata dari sensor arus sebesar 1,38% dan sensor tegangan sebesar 1,5%.
4. Tampilan monitoring dapat menampilkan data berupa ID pelanggan, alamat pelanggan, arus, tegangan, daya, pemakaian daya, dan indicator beban.
5. Monitoring ini juga dapat digunakan untuk mengakses pemakaian daya pelanggan PJU lainnya dengan menggunakan IP yang berbeda.

5.2 Saran

Dengan memperhatikan beberapa kelemahan dari proyek Tugas Akhi rini, maka diberikan beberapa saran yang sekiranya dapat dikembangkan pada masa yang akan datang demi kesempurnaan dari proyek Tugas Akhir ini. Adapun beberapa saran tersebut yaitu:

1. Studi Kelayakan Tugas Akhir harus lebih diperhatikan dan melibatkan pihak PLN, sehingga alat yang dibuat sesuai dengan kebutuhan PLN.
2. Pengambilan data lebih banyak dan berulang, untuk memastikan kemampuan alat serta keakuratan alat.
3. Perlu menambahkan sensor faktor daya pada alat ini.
4. Dalam memonitoring data yang ada sebaiknya dilakukan secara bertahap.

5. Lebih cermat dalam pemilihan jenis *Access Point* (AP), karena jarak jangkauan dari AP sangat berpengaruh terhadap keberhasilan pengiriman data.

DAFTAR PUSTAKA

[1] Fadly Irawan, Ahmad., "Analisis Peningkatan Efisiensi Penerangan Jalan Umum (PJU) Di Kabupaten Jember", ***Skripsi***, Teknik Elektro, Universitas Brawijaya, 2013.

[2] Ramadhan, Reza., "Prototipe Manuver Beban Penyulang Pada Pelanggan VIP Dengan Arduino", ***Tugas Akhir***, D3 Teknik Elektro, ITS Surabaya, 2014.

[3] ....., ***Electricity Meter (Analog)***, <http://wiki.iteadstudio.com/Electricity_Meter%28Analog%29>, diakses tanggal 20 April 2015.

[4] Kadir, Abdul. 2012, ***Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya Menggunakan Arduino***, Yogyakarta: Penerbit ANDI.

[5] Cintia Pratiwi, Shely., "Prototipe Monitoring Kinerja Rumah Pompa Dengan Media Modem GSM", ***Tugas Akhir***, D3 Teknik Elektro, ITS Surabaya, 2014.

[6] Ladyada. 2014, ***I2C/SPI LCD Backpack***, Adafruit Industries.

[7] Riza Fathur, R., "Alat Monitoring Suhu Dan Kelembaban Pada Peralatan Hubung Bagi Tegangan Rendah (Phb-Tr) Melalui Media Modem GSM", ***Tugas Akhir***, D3 Teknik Elektro, ITS Surabaya, 2013.

[8] ....., ***TP-Link TL-WA5210G***, <http://www.tp-link.com/en/products/details/cat-4581_TL-WA5210G.html>, diakses tanggal 20 Mei 2015.

[9] Artanto, Dian. 2012, ***Interaksi Arduino dan LabVIEW***, Jakarta: Gramedia.

Halaman ini sengaja dikosongkan

LAMPIRAN A
PROGRAM

***Listing* Program Arduino IDE**

```
//Relay
#define RELAY1  6
#define RELAY2  7
#define RELAY3  8

//program lcd
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#define I2C_ADDR    0x27 // <<----- Add your address here.  Find it
from I2C Scanner
#define BACKLIGHT_PIN     3
#define En_pin  2
#define Rw_pin  1
#define Rs_pin  0
#define D4_pin  4
#define D5_pin  5
#define D6_pin  6
#define D7_pin  7
int n = 1;
LiquidCrystal_I2C
            lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pi
n,D7_pin);

//Ethernnet
#include <SPI.h>        // needed for Arduino versions later than 0018
#include <Ethernet.h>
#include <EthernetUdp.h>
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 1, 177); //177 untuk PJU 511085655815 /// 178
untuk PJU 511085655816
unsigned int localPort = 8888;      // local port to listen on
// buffers for receiving and sending data
```

```
char packetBuffer[UDP_TX_PACKET_MAX_SIZE]; //buffer to hold
incoming packet,
int data_masuk;

// An EthernetUDP instance to let us send and receive packets over UDP
EthernetUDP Udp;
//////////////////////////////////////
#define sensor_arus A0 // Analog input pin that sensor is attached to
float imax;             //amplitude current
float irms;       //effective current
float voltage;
float daya;
float kwh;
char volt[10],amp[10],watt[10],total[10];
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
String indikator="";

//sd card
#include<SD.h>
const int chipSelect = 4;

//RTC
#include "Wire.h"
#define DS3231_I2C_ADDRESS 0x68
//jam
  // Convert normal decimal numbers to binary coded decimal
  byte decToBcd(byte val)
    {
    return( (val/10*16) + (val%10) );
    }
  // Convert binary coded decimal to normal decimal numbers
  byte bcdToDec(byte val)
    {
    return( (val/16*10) + (val%16) );
    }
////////////////set time data logger//////////////////
int h = 17, m = 0, s = 0;
////////////////####################////////////////////
void setup()
```

```
{
  kwh=0;
  Wire.begin();
  // set the initial time here:
  // DS3231 seconds, minutes, hours, day, date, month, year
  setDS3231time(0,59,16,2,7,7,15);

  Serial.begin(9600);
  while (!Serial) {
   ; // wait for serial port to connect. Needed for Leonardo only
  }

  Serial.print("Initializing SD card...");
  // make sure that the default chip select pin is set to
  // output, even if you don't use it:
  pinMode(10, OUTPUT);

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }
  Serial.println("card initialized.");
  printheader();
  // Switch on the backlight
  lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
  lcd.setBacklight(HIGH);
  lcd.home (); // go home
  pinMode(RELAY1, OUTPUT);
  pinMode(RELAY2, OUTPUT);
  pinMode(RELAY3, OUTPUT);

  //Logika awal relay
  digitalWrite(RELAY1,LOW);
  digitalWrite(RELAY2,LOW);
  digitalWrite(RELAY3,LOW);

  lcd.begin(16,2);
```

```
  Ethernet.begin(mac, ip);
  Udp.begin(localPort);
  Serial.begin(9600);
  //initial status
  lcd.print("Monitoring PJU");
  lcd.setCursor(0,1);
  lcd.print("PT PLN Persero");
  delay(2000);
  lcd.clear();
}

//////////////////////PROGRAM VOID////////////////
void printheader() // this function prints header to SD card
{
   File dataFile = SD.open("DATALOG.csv", FILE_WRITE);
   if (dataFile)
   {
      dataFile.println("Date & Time, Tegangan, Arus, Daya, Pemakaian
Daya, Indikator Beban");
      dataFile.close();
   }
}
void setDS3231time(byte second, byte minute, byte hour, byte
dayOfWeek, byte
dayOfMonth, byte month, byte year)
{
  // sets time and date data to DS3231
  Wire.beginTransmission(DS3231_I2C_ADDRESS);
  Wire.write(0); // set next input to start at the seconds register
  Wire.write(decToBcd(second)); // set seconds
  Wire.write(decToBcd(minute)); // set minutes
  Wire.write(decToBcd(hour)); // set hours
  Wire.write(decToBcd(dayOfWeek)); // set day of week (1=Sunday,
7=Saturday)
  Wire.write(decToBcd(dayOfMonth)); // set date (1 to 31)
  Wire.write(decToBcd(month)); // set month
  Wire.write(decToBcd(year)); // set year (0 to 99)
  Wire.endTransmission();
}
```

```
void readDS3231time(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year)
{
  Wire.beginTransmission(DS3231_I2C_ADDRESS);
  Wire.write(0); // set DS3231 register pointer to 00h
  Wire.endTransmission();
  Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
  // request seven bytes of data from DS3231 starting from register 00h
  *second = bcdToDec(Wire.read() & 0x7f);
  *minute = bcdToDec(Wire.read());
  *hour = bcdToDec(Wire.read() & 0x3f);
  *dayOfWeek = bcdToDec(Wire.read());
  *dayOfMonth = bcdToDec(Wire.read());
  *month = bcdToDec(Wire.read());
  *year = bcdToDec(Wire.read());
}
///////////////////////////////////////////////////
  void sensor(){
  //sensor arus
  int ADCmax;
  ADCmax = getMaxValue();
  imax=((float)ADCmax/1023*5/200*1000);
  irms=imax/1.414;
  //sensor tegangan
  int sensorValue = analogRead(A1);
  voltage = (((float)sensorValue/1023*5*49.965)+1.9173);
  if (voltage==1.9173)
  voltage=0;
  // daya
  daya=irms*voltage;
  Serial.print(kwh);
  }
  void pins_init()
{
```

```
  pinMode(sensor_arus, INPUT);
}
/*Function: Sample for 1000ms and get the maximum value from the
SIG pin*/

int getMaxValue()

{
  int sensorValue;          //value read from the sensor
  int ADCmax = 0;
  uint32_t start_time = millis();
  while((millis()-start_time) < 1000)//sample for 1000ms
  {
    sensorValue = analogRead(sensor_arus);
    if (sensorValue > ADCmax)
    {
          /*record the maximum sensor value*/
          ADCmax = sensorValue;
    }
  }
  return ADCmax;
}
//////////////##########################/////////////////
  void logikarelay(){
  readDS3231time(&second, &minute, &hour, &dayOfWeek,
&dayOfMonth, &month,
  &year);
  if (daya<300)
  indikator="HIJAU";
  if (daya>300){digitalWrite(RELAY1,HIGH);
  indikator="KUNING";}
  else{digitalWrite(RELAY1,LOW);}
  if (daya>400){
  digitalWrite(RELAY2,HIGH);
  indikator="KUNING MERAH";}
  else{digitalWrite(RELAY2,LOW);}
  if (hour==17 && minute==0 && second==0){
  digitalWrite(RELAY3,HIGH);}
  if (hour==17 && minute==5 && second==0)
```

```
  digitalWrite(RELAY3,LOW);
  }
//////////////#####################//////////////////
void ethernet(){
  dtostrf(voltage,4,3,volt);
  dtostrf(irms,4,3,amp);
  dtostrf(daya,4,3,watt);
  dtostrf(kwh,4,5,total);
  int packetSize = Udp.parsePacket();
  if (packetSize)
  {
    Serial.print("Received packet of size ");
    Serial.println(packetSize);
    Serial.print("From ");
    IPAddress remote = Udp.remoteIP();
    for (int i = 0; i < 4; i++)
    {
     Serial.print(remote[i], DEC);
     if (i < 3)
     {
       Serial.print(".");
     }
    }
    Serial.print(", port ");
    Serial.println(Udp.remotePort());
    // read the packet into packetBufffer
    Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
    Serial.println("Contents:");
    Serial.println(packetBuffer);
    data_masuk= atoi (packetBuffer);
    // send a reply, to the IP address and port that sent us the packet we
received
    if (data_masuk==5)
    {
     Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
     Udp.write(volt);
     Udp.endPacket();
    }
    else if (data_masuk==6)
```

```
    {
     Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
     Udp.write(amp);
     Udp.endPacket();
    }
    else if (data_masuk==7)
    {
     Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
     Udp.write(watt);
     Udp.endPacket();
    }
    else if (data_masuk==8)
    {
     Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
     Udp.write(total);
     Udp.endPacket();
    }
   }
  }
///////////////////////////////////////////
void tampilkan(){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("V=");
  lcd.setCursor(2,0);
  lcd.print(voltage,2);
  lcd.setCursor(9,0);
  lcd.print("I=");
  lcd.setCursor(11,0);
  lcd.print(irms,2);
  lcd.setCursor(0,1);
  lcd.print("P=");
  lcd.setCursor(2,1);
  lcd.print(daya,1);
}
////////////////////////////////////////////////
void displayTime()
{
```

```
  readDS3231time(&second, &minute, &hour, &dayOfWeek,
&dayOfMonth, &month,
  &year);
  // send it to the serial monitor
  lcd.setCursor(8,1);
  lcd.print(hour, DEC);
  // convert the byte variable to a decimal number when displayed
  //lcd.setCursor(10,1);
  lcd.print(":");
  if (minute<10)
  {
    lcd.print("0");
  }
  //lcd.setCursor(12,1);
  lcd.print(minute, DEC);
  //lcd.setCursor(13,1);
  lcd.print(":");
  if (second<10)
  {
    lcd.print("0");
  }
  lcd.print(second, DEC);
  Serial.print(second, DEC);
  Serial.print(" ");
  Serial.print(dayOfMonth, DEC);
  Serial.print("/");
  Serial.print(month, DEC);
  Serial.print("/");
  Serial.print("20");
  Serial.println(year, DEC);
}
////////##############################/////////
void sdcard(){
  File dataFile = SD.open("DATALOG.csv", FILE_WRITE);
  if (dataFile) {
    dataFile.print(dayOfMonth, DEC);
    dataFile.print("/");
    dataFile.print(month, DEC);
    dataFile.print("/");
```

```
    dataFile.print(year, DEC);
    dataFile.print("  ");
    dataFile.print(hour, DEC);
    dataFile.print(":");
    dataFile.print(minute, DEC);
    dataFile.print(":");
    dataFile.print(second, DEC);
    dataFile.print(",");
    dataFile.print(volt);
    dataFile.print(",");
    dataFile.print(amp);
    dataFile.print(",");
    dataFile.print(watt);
    dataFile.print(",");
    dataFile.print(total);
    dataFile.print(",");
    dataFile.println(indikator);
    dataFile.close();
    }
}
void loop()
{
  kwh=kwh+daya/3600/1000;
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  readDS3231time(&second, &minute, &hour, &dayOfWeek,
&dayOfMonth, &month,
  &year);
  if(s>=60 &&m >=60 && h>=24)
  {
    s = s-60;  //reset s (seconds) to 0 after one minute or more has passed
    m++;  // add one minute after 60 seconds have passed
    m = m-60;
    h = h-24;
  }
  if(hour==h+1 || hour==0)
  {
    h = hour;  //change value of h to actual hour after one hour or one day
  }
  if (hour==h && minute==m && second==s){
```

```
  sdcard();
  m++;
  }
  sensor();
  logikarelay();
  ethernet();
  tampilkan();
  displayTime();
}
```

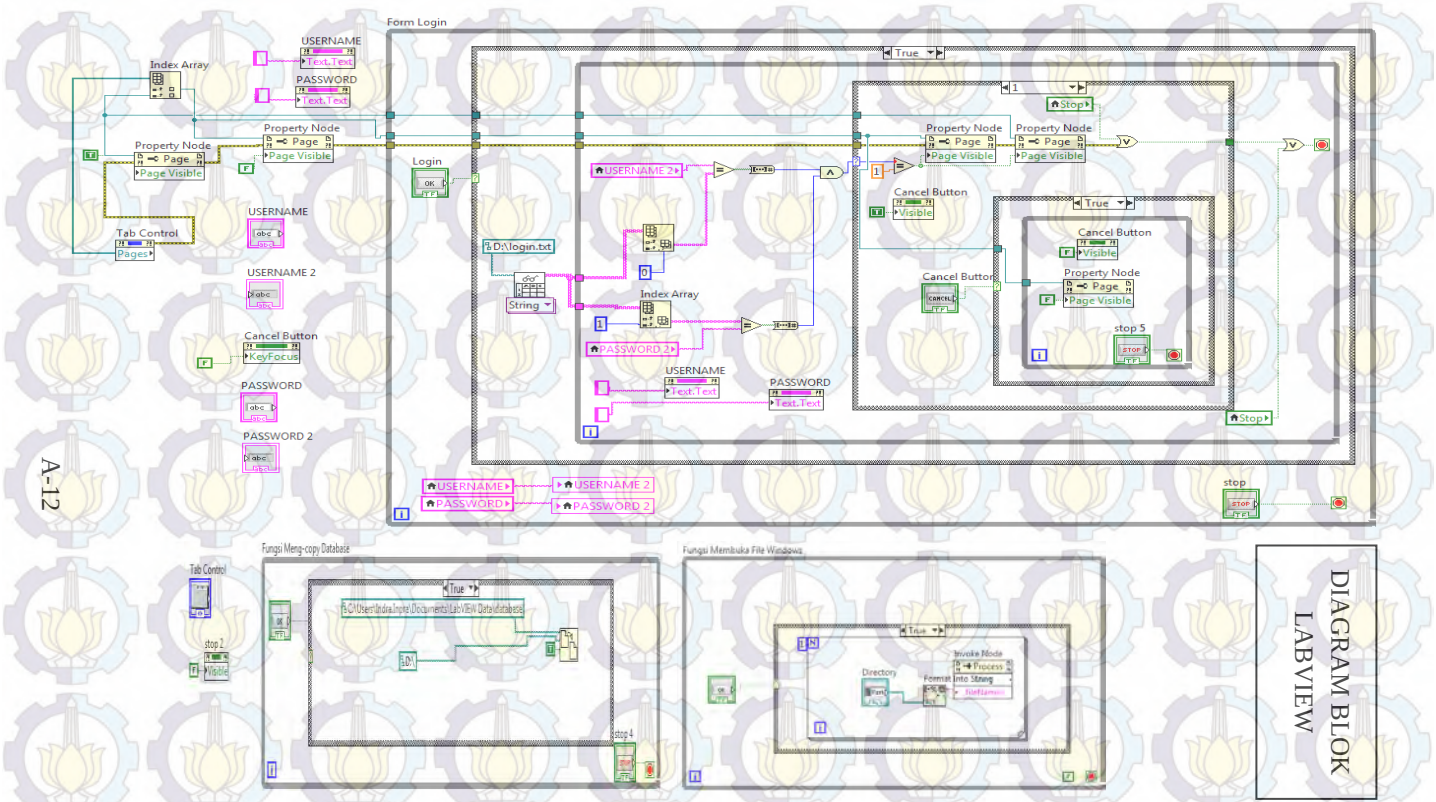
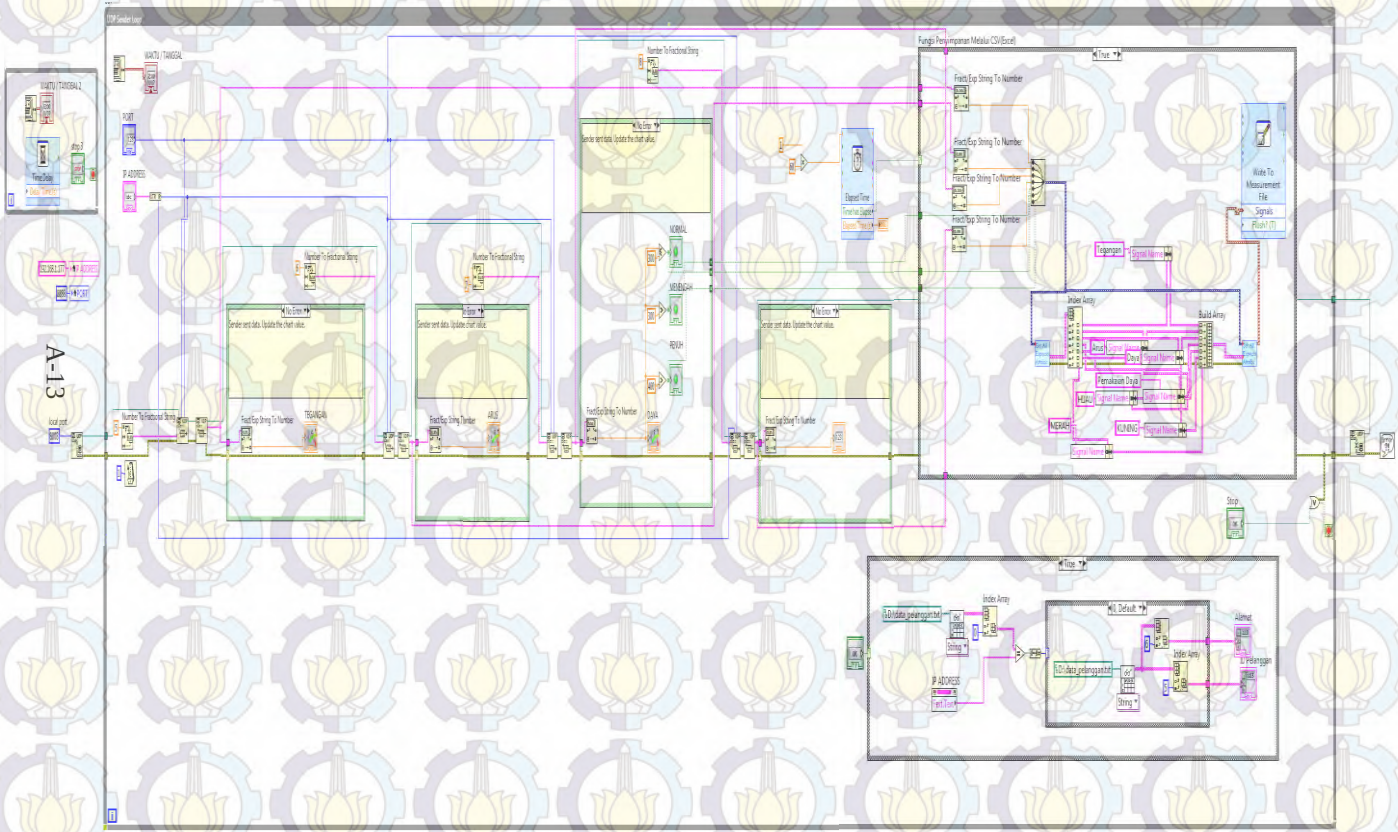


DIAGRAM BLOK
LABVIEW


TCP Sender Loop
JAM TU / TANGGAL
PORT
IP ADDRESS
Number To Fractional String
Sender sent data. Update the chart value.
TEGANGAN
ARUS
NORMAL
MENENGAH
RENDAH
DAYA
Elapsed Time
Fungsi Penyimpanan Melalui CSV(Excel)
Fract/Exp String To Number
Write To Measurement File
Tegangan
Index Array
Build Array
Pemakaian Daya
HIJAU
KUNING
MERAH
Stop
Alamat
Tegangan
Default

A-13

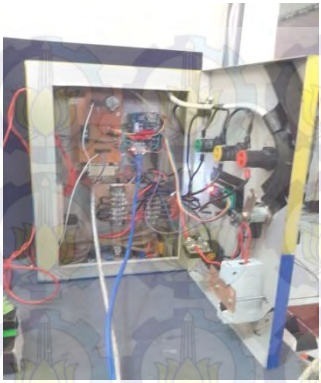
Halaman ini sengaja dikosongkan

LAMPIRAN B
DOKUMENTASI ALAT DAN PENGUJIAN

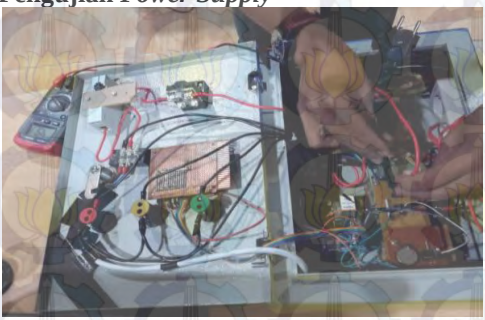
Sistem Hardware Keseluruhan

Tampak Luar Panel Monitoring

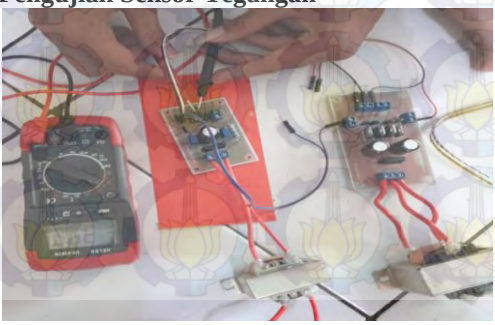
Tampak Dalam Panel Monitoring



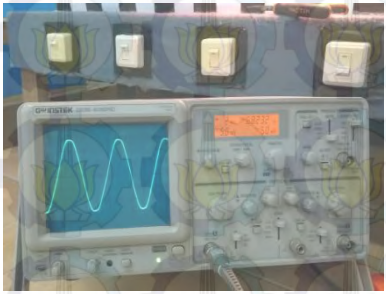
Pengujian *Power Supply*



Pengujian Sensor Tegangan

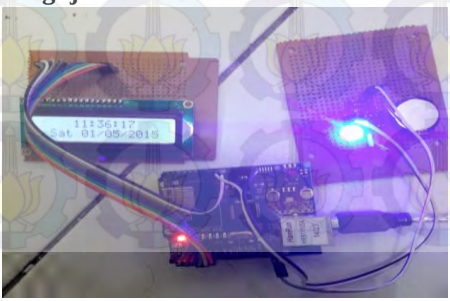


Pengujian Sensor Arus

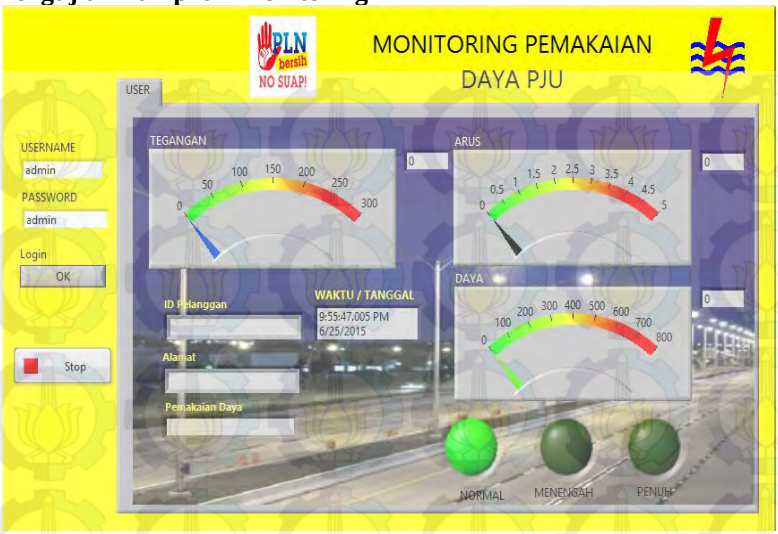


Pengujian Ethernet Shield

Pengujian LCD dan RTC



Pengujian Tampilan Monitoring



Tampilan *Form Admin*

Tampilan *Database* pada PC

	A	B	C	D	E	F	G	H
1	Time	Tegangan	Arus	Daya	Pemakaian Daya	HIJAU	KUNING	MERAH
2	7/7/2015 17:59:57.694	219.787	0.478	104.924	0.09943	1	0	0
3	7/7/2015 18:59:59.941	218.811	0.495	108.429	0.20004	1	0	0
4	7/7/2015 20:00:03.944	218.078	0.495	107.825	0.30224	1	0	0
5	7/7/2015 21:00:20.000	208.31	1.451	301.866	0.44252	0	1	0
6	7/7/2015 22:00:20.768	200.86	2.051	411.964	0.85447	0	1	1

Tampilan *Database* pada *microSD*

	A	B	C	D	E	F
1	Date & Time	Tegangan	Arus	Daya	Pemakaian Daya	Indikator Beban
2	7/7/2015 17:00	0	0	0	0	HIJAU
3	7/7/2015 18:00	219.543	0.478	104.924	0.09952	HIJAU
4	7/7/2015 19:00	219.055	0.495	108.429	0.20004	HIJAU
5	7/7/2015 20:00	218.078	0.495	107.825	0.30224	HIJAU
6	7/7/2015 21:00	208.554	1.451	302.575	0.44093	KUNING
7	7/7/2015 22:00	199.03	2.065	411.054	0.72588	KUNING MERAH

Halaman ini sengaja dikosongkan

LAMPIRAN C
DATASHEET

Datasheet ATmega328

Features

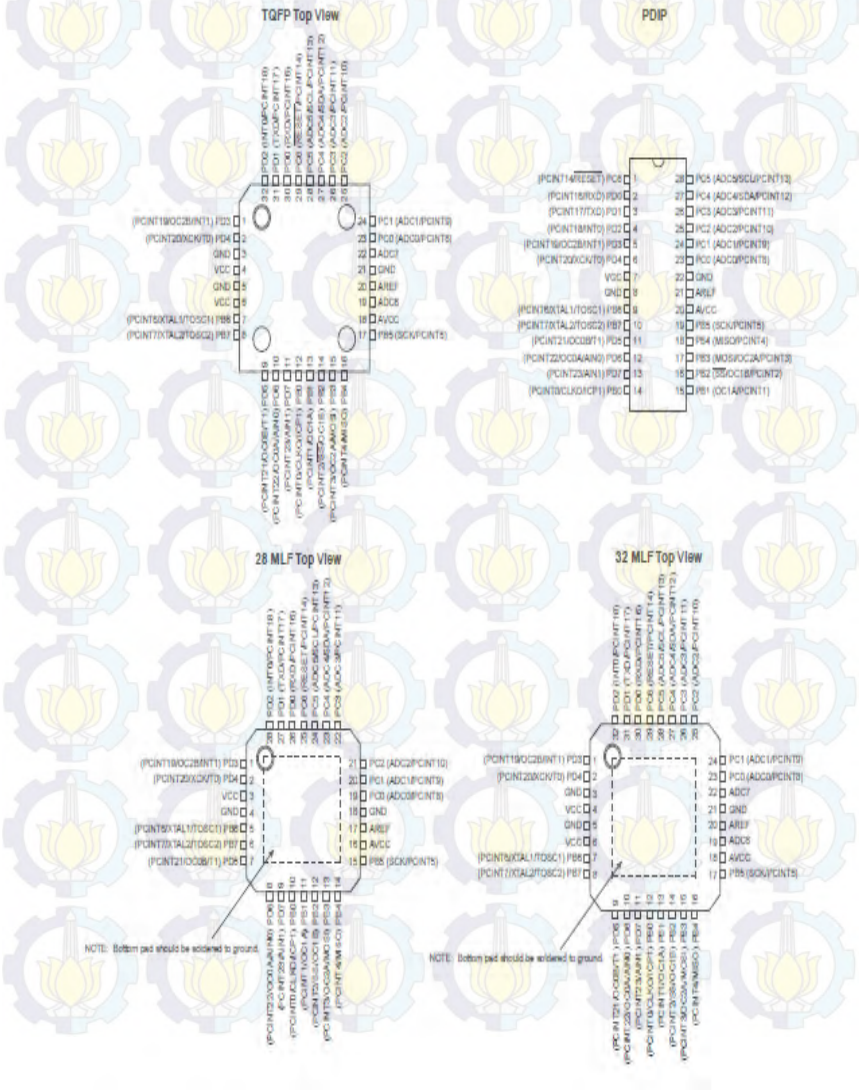
- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash progam memory (ATmega48PA/88PA/168PA/328P)
  - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
  - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    - Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 µA
  - Power-save Mode: 0.75 µA (including 32 kHz RTC)

8-bit AVR® Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash

ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P

1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P



1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 82 and "System Clock and Clock Options" on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/$\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 28-3 on page 318. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 85.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 88.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC}.

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

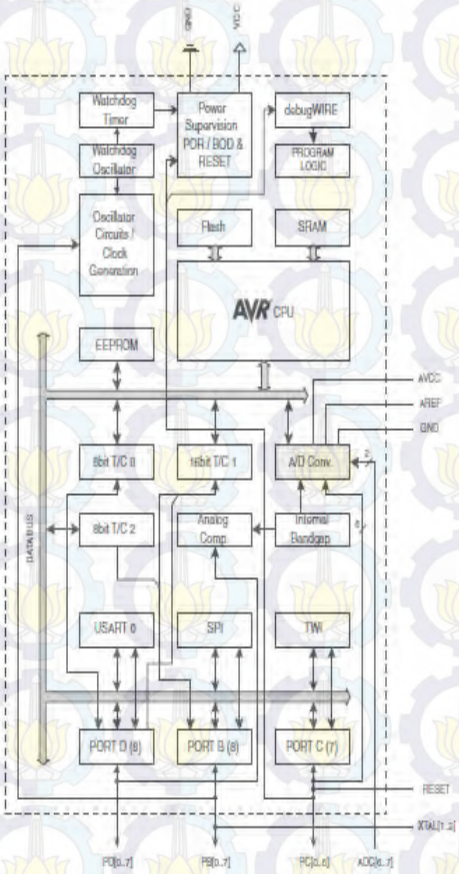
In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

2. Overview

The ATmega48PA/88PA/168PA/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting

architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48PA/88PA/168PA/328P provides the following features: 4K/8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48PA/88PA/168PA/328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48PA/88PA/168PA/328P AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

2.2 Comparison Between ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P

The ATmega48PA, ATmega88PA, ATmega168PA and ATmega328P differ only in memory sizes, boot loader support, and interrupt vector sizes. Table 2-1 summarizes the different memory and interrupt vector sizes for the three devices.

Table 2-1. Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48PA	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88PA	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168PA	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instruction words/vector

ATmega88PA, ATmega168PA and ATmega328P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega48PA, there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.

3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on http://www.atmel.com/avr.

4. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

5. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

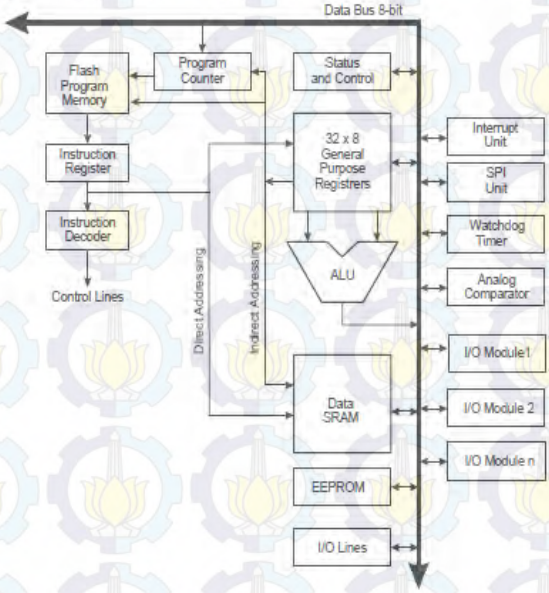
For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

6. AVR CPU Core

6.1 Overview

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 6-1. Block Diagram of the AVR Architecture



***Datasheet* Arduino UNO**

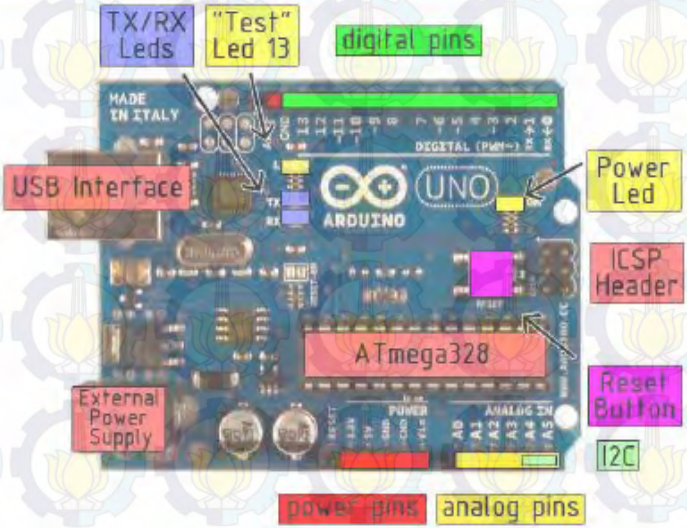
Technical Specification

EAGLE files: arduino-duemilanove-uno-design.zip Schematic: arduino-uno-schematic.pdf

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



Power

The Arduino Uno 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 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 Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno 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).** Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** 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: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

- **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.

The Uno has 6 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 is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the Wire library.

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.

See also the mapping between Arduino pins and Atmega328 ports.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial 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 Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

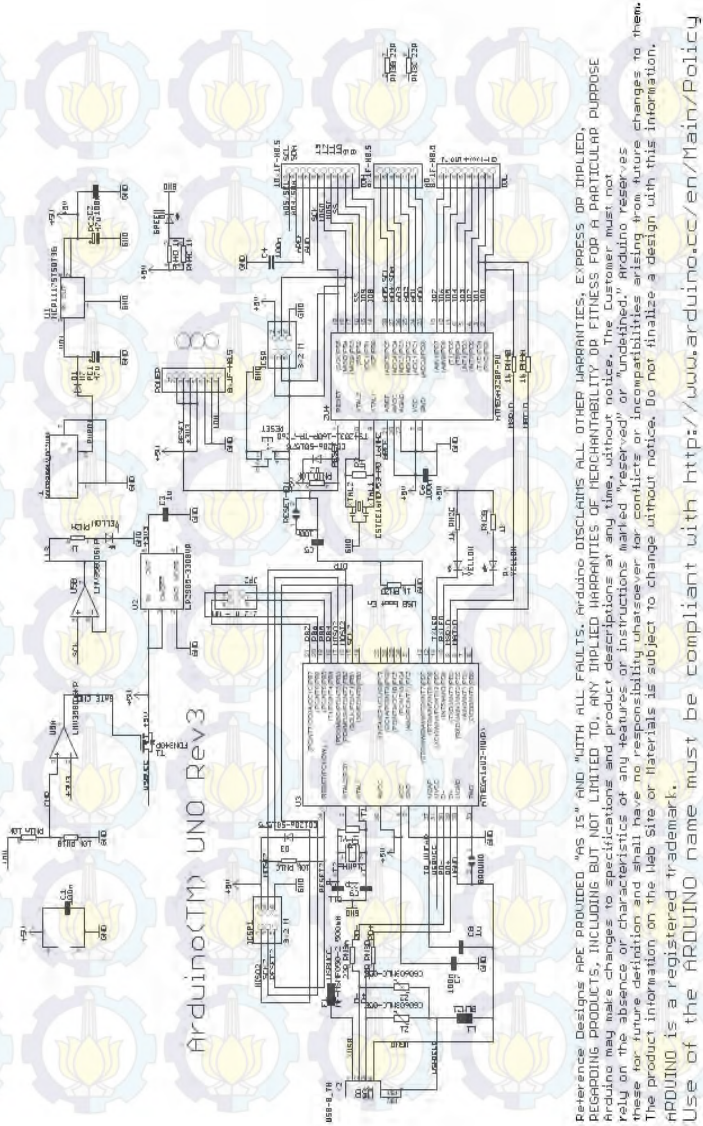
The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ ATmega328" from the **Tools** > **Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno 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.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

***Schematic* Arduino UNO**

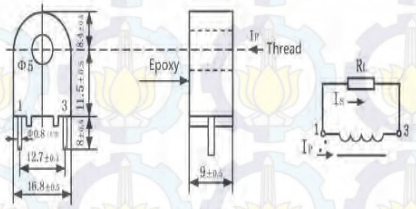


Datasheet **Sensor Arus CT Ta-12 100**

Technical Parameter

Model	Ratio	Input Currenton	Output Current	Sample Resistance	Sampling Voltage	Nonlinear Scale	Phase Drift	Working Frequency	Working Temperature	Anti-electricity Level
TA12-100	1000:1	0~5A	0~5mA	200Ω	1V	≤0.3%	≤10′	20Hz~20KHz	-55℃~+85℃	6KV AC/1min
TA12-200	2000:1	0~5A	0~2.5mA	800Ω	2V	≤0.2%	≤5′	20Hz~20KHz	-55℃~+85℃	6KV AC/1min
TA12L-100	1000:1	0~5A	0~5mA	200Ω	1V	≤0.3%	≤10′	20Hz~20KHz	-55℃~+85℃	6KV AC/1min
TA12L-200	2000:1	0~5A	0~2.5mA	800Ω	2V	≤0.2%	≤5′	20Hz~20KHz	-55℃~+85℃	6KV AC/1min

Schematic

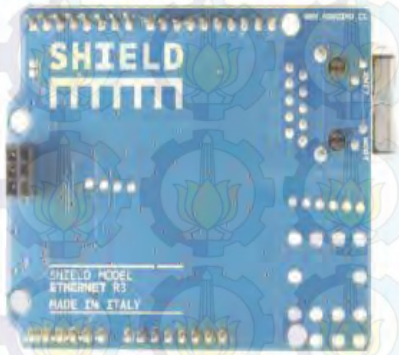


***Datasheet* Ethernet Shield**

ARDUINO

Arduino Ethernet Shield

Arduino Ethernet Shield R3 Front



Arduino Ethernet Shield R3 Back

Download: arduino-ethernet-shield-06-schematic.pdf, arduino-ethernet-shield-06-reference-design.zip

Overview

The Arduino Ethernet Shield connects your Arduino to the internet in mere minutes. Just plug this module onto your Arduino board, connect it to your network with an RJ45 cable (not included) and follow a few simple instructions to start controlling your world through the internet. As always with Arduino, every element of the platform – hardware, software and documentation – is freely available and open-source. This means you can learn exactly how it's made and use its design as the starting point for your own circuits. Hundreds of thousands of Arduino boards are already fueling people's creativity all over the world, everyday. Join us now, Arduino is you!

- Requires and Arduino board (not included)
- Operating voltage 5V (supplied from the Arduino Board)
- Ethernet Controller: W5100 with internal 16K buffer
- Connection speed: 10/100Mb
- Connection with Arduino on SPI port

Description

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the Wiznet W5100 ethernet chip (datasheet). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the Ethernet library to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The most recent revision of the board exposes the 1.0 pinout on rev 3 of the Arduino UNO board.

The Ethernet Shield has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled.

There is an onboard micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Uno and Mega (using the Ethernet library). The onboard microSD card reader is accessible through the SD Library. When working with this library, SS is on Pin 4. The original revision of the shield contained a full-size SD card slot; this is not supported.
The shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up.
The current shield has a Power over Ethernet (PoE) module designed to extract power from a conventional twisted pair Category 5 Ethernet cable:

- IEEE802.3af compliant
- Low output ripple and noise (100mVpp)
- Input voltage range 36V to 57V
- Overload and short-circuit protection
- 9V Output
- High efficiency DC/DC converter: typ 75% @ 50% load
- 1500V isolation (input to output)

NB: the Power over Ethernet module is proprietary hardware not made by Arduino, it is a third party accessory. For more information, see the datasheet
The shield does not come with the PoE module built in, it is a separate component that must be added on.
Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 11, 12, and 13 on the 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 or the SPI interface won't work.
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.
The shield provides a standard RJ45 ethernet jack.
The reset button on the shield resets both the W5100 and the Arduino board.
The shield contains a number of informational LEDs:

- PWR: indicates that the board and shield are powered
- LINK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FULLD: indicates that the network connection is full duplex
- 100M: 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
- COLL: flashes when network collisions are detected

The solder jumper marked "INT" can be connected to allow the Arduino board to receive interrupt-driven notification of events from the W5100, but this is not supported by the Ethernet library. The jumper connects the INT pin of the W5100 to digital pin 2 of the Arduino.
See also: getting started with the ethernet shield and Ethernet library reference

Datasheet WiFi TPLink TL-WA5210G

HARDWARE FEATURES	
Interface	1 10/100Mbps Auto-Sensing RJ45 Port(Auto MDI/MDIX, PoE) 1 External Reverse SMA Connector One Grounding Terminal
Button	Reset Button
External Power Supply	12VDC / 1.0A Linear PSU
Wireless Standards	IEEE 802.11g, IEEE 802.11b
Antenna	12dBi Dual-Polarized Directional Antenna
Dimensions (W x D x H)	10.4 × 4.7 × 3.2 in. (265x120x83mm)
Antenna Beamwidth	Horizontal: 60° Vertical: 30°
Protection	15kV ESD Protection, 4000V Lightning Protection Grounding Terminal Integrated
WIRELESS FEATURES	
Frequency	2.4-2.4835GHz
Signal Rate	11g: Up to 54Mbps(dynamic) 11b: Up to 11Mbps(dynamic)
Reception Sensitivity	802.11g 54M: -76dBm 48M: -78dBm 36M: -82dBm 12M: -91dBm 9M:-92dBm 802.11b 11M:-90dBm 5.5M:-92dBm 1M:-98dBm
Transmit Power	<20dBm (EIRP, For countries using CE Standards) <27dBm (Peak Output Power, For countries using FCC Standards)
Wireless Modes	AP Router Mode AP Client Router Mode (WISP Client) AP/Client/WDS Bridge/Repeater mode
Wireless Functions	WDS Bridge, Wireless Statistics
Wireless Security	SSID Enable/Disable MAC Address Filter 64/128/152-bit WEP Encryption WPA/WPA2/WPA-PSK/WPA2-PSK (AES/TKIP) Encryption
Wireless Range	2km with Integrated Antenna 15km Maximum (High gain directional antenna required)
Advanced Functions	Up to 60 meters PoE is supported Provides 4-level signal LED indicator
OTHERS	
Certification	CE, FCC, RoHS
Package Contents	TL-WA5210G Power Supply Unit Power Injector Resource CD Quick Installation Guide
System Requirements	Microsoft Windows 98SE, NT, 2000, XP, Vista™ or Windows 7, MAC OS, NetWare, UNIX or Linux.
Environment	Operating Temperature: -30°C~70°C (-22°F~158°F) Storage Temperature: -40°C~70°C (-40°F~158°F) Operating Humidity: 10%~90% non-condensing Storage Humidity: 5%~95% non-condensing

DS3231

Extremely Accurate I²C-Integrated RTC/TCXO/Crystal

General Description

The DS3231 is a low-cost, extremely accurate I²C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an $\overline{AM}$/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I²C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of V_{CC} to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the $\overline{RST}$ pin is monitored as a pushbutton input for generating a µP reset.

Benefits and Features

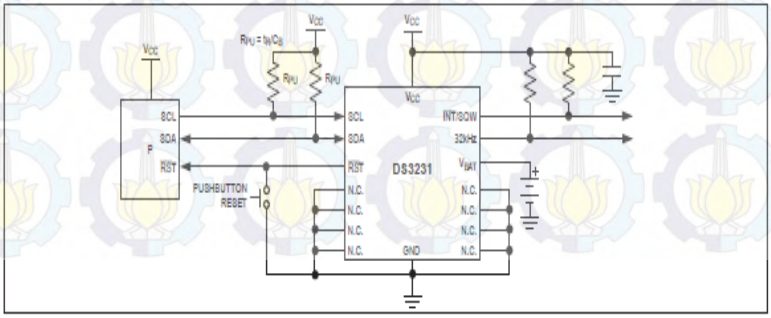
- Highly Accurate RTC Completely Manages All Timekeeping Functions
  - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
  - Accuracy ±2ppm from 0°C to +40°C
  - Accuracy ±3.5ppm from -40°C to +85°C
  - Digital Temp Sensor Output: ±3°C Accuracy
  - Register for Aging Trim
  - $\overline{RST}$ Output/Pushbutton Reset Debounce Input
  - Two Time-of-Day Alarms
  - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
  - Fast (400kHz) I²C Interface
- Battery-Backup Input for Continuous Timekeeping
  - Low Power Operation Extends Battery-Backup Run Time
  - 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to +70°C) and Industrial (-40°C to +85°C)
- Underwriters Laboratories® (UL) Recognized

Applications

- Servers
- Telematics
- Utility Power Meters
- GPS

Ordering Information and Pin Configuration appear at end of data sheet.

Typical Operating Circuit



Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground.....-0.3V to +6.0V
Junction-to-Ambient Thermal Resistance (θ_{JA}) (Note 1)73°C/W
Junction-to-Case Thermal Resistance (θ_{JC}) (Note 1).....23°C/W
Operating Temperature Range
DS3231S.....0°C to +70°C
DS3231SN.....-40°C to +85°C

Junction Temperature.....+125°C
Storage Temperature Range.....-40°C to +85°C
Lead Temperature (soldering, 10s).....+260°C
Soldering Temperature (reflow, 2 times max).....+260°C
(see the *Handling, PCB Layout, and Assembly* section)

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to www.maximintegrated.com/thermal-tutorial.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Recommended Operating Conditions

($T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		2.3	3.3	5.5	V
	V_{BAT}		2.3	3.0	5.5	V
Logic 1 Input SDA, SCL	V_{IH}		0.7 x V_{CC}		V_{CC} + 0.3	V
Logic 0 Input SDA, SCL	V_{IL}		-0.3		0.3 x V_{CC}	V

Electrical Characteristics

(V_{CC} = 2.3V to 5.5V, V_{CC} = Active Supply (see Table 1), $T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.) (Typical values are at V_{CC} = 3.3V, V_{BAT} = 3.0V, and T_A = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Active Supply Current	I_{CCA}	(Notes 4, 5)	V_{CC} = 3.63V			200	µA
			V_{CC} = 5.5V			300	
Standby Supply Current	I_{CCS}	I²C bus inactive, 32kHz output on, SQW output off (Note 5)	V_{CC} = 3.63V			110	µA
			V_{CC} = 5.5V			170	
Temperature Conversion Current	$I_{CCACONV}$	I²C bus inactive, 32kHz output on, SQW output off	V_{CC} = 3.63V			575	µA
			V_{CC} = 5.5V			650	
Power-Fail Voltage	V_{PF}			2.45	2.575	2.70	V
Logic 0 Output, 32kHz, INT/SQW, SDA	V_{OL}	I_{OL} = 3mA				0.4	V
Logic 0 Output, RST	V_{OL}	I_{OL} = 1mA				0.4	V
Output Leakage Current 32kHz, INT/SQW, SDA	I_{LO}	Output high impedance		-1	0	+1	µA
Input Leakage SCL	I_{LI}			-1		+1	µA
RST Pin I/O Leakage	I_{OL}	RST high impedance (Note 6)		-200		+10	µA
V_{BAT} Leakage Current (V_{CC} Active)	I_{BATLKG}				25	100	nA

Electrical Characteristics (continued)

(V_{CC} = 2.3V to 5.5V, V_{CC} = Active Supply (see Table 1), T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Typical values are at V_{CC} = 3.3V, V_{BAT} = 3.0V, and T_A = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Output Frequency	f_{OUT}	V_{CC} = 3.3V or V_{BAT} = 3.3V			32.768		kHz
Frequency Stability vs. Temperature (Commercial)	$\Delta f/f_{OUT}$	V_{CC} = 3.3V or V_{BAT} = 3.3V, aging offset = 00h	0°C to +40°C			±2	ppm
			>40°C to +70°C			±3.5	
Frequency Stability vs. Temperature (Industrial)	$\Delta f/f_{OUT}$	V_{CC} = 3.3V or V_{BAT} = 3.3V, aging offset = 00h	-40°C to <0°C			±3.5	ppm
			0°C to +40°C			±2	
			>40°C to +85°C			±3.5	
Frequency Stability vs. Voltage	$\Delta f/V$				1		ppm/V
Trim Register Frequency Sensitivity per LSB	Δf/LSB	Specified at:	-40°C		0.7		ppm
			+25°C		0.1		
			+70°C		0.4		
			+85°C		0.8		
Temperature Accuracy	Temp	V_{CC} = 3.3V or V_{BAT} = 3.3V		-3		+3	°C
Crystal Aging	$\Delta f/f_O$	After reflow, not production tested	First year		±1.0		ppm
			0–10 years		±5.0		

Electrical Characteristics

(V_{CC} = 0V, V_{BAT} = 2.3V to 5.5V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Active Battery Current	I_{BATA}	$\overline{EOSC}$ = 0, BBSQW = 0, SCL = 400kHz (Note 5)	V_{BAT} = 3.63V			70	µA
			V_{BAT} = 5.5V			150	
Timekeeping Battery Current	I_{BATT}	$\overline{EOSC}$ = 0, BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = V_{BAT} (Note 5)	V_{BAT} = 3.63V		0.84	3.0	µA
			V_{BAT} = 5.5V		1.0	3.5	
Temperature Conversion Current	I_{BATTC}	$\overline{EOSC}$ = 0, BBSQW = 0, SCL = SDA = 0V or SCL = SDA = V_{BAT}	V_{BAT} = 3.63V			575	µA
			V_{BAT} = 5.5V			650	
Data-Retention Current	I_{BATTDR}	$\overline{EOSC}$ = 1, SCL = SDA = 0V, +25°C				100	nA

AC Electrical Characteristics

($V_{CC} = V_{CC(MIN)}$ to $V_{CC(MAX)}$ or $V_{BAT} = V_{BAT(MIN)}$ to $V_{BAT(MAX)}$, $V_{BAT} > V_{CC}$, $T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f_{SCL}	Fast mode	100		400	kHz
		Standard mode	0		100	
Bus Free Time Between STOP and START Conditions	t_{BUF}	Fast mode	1.3			µs
		Standard mode	4.7			
Hold Time (Repeated) START Condition (Note 7)	$t_{HD:STA}$	Fast mode	0.6			µs
		Standard mode	4.0			
Low Period of SCL Clock	t_{LOW}	Fast mode	1.3			µs
		Standard mode	4.7			
High Period of SCL Clock	t_{HIGH}	Fast mode	0.6			µs
		Standard mode	4.0			
Data Hold Time (Notes 8, 9)	$t_{HD:DAT}$	Fast mode	0		0.9	µs
		Standard mode	0		0.9	
Data Setup Time (Note 10)	$t_{SU:DAT}$	Fast mode	100			ns
		Standard mode	250			
START Setup Time	$t_{SU:STA}$	Fast mode	0.6			µs
		Standard mode	4.7			
Rise Time of Both SDA and SCL Signals (Note 11)	t_R	Fast mode	20 + 0.1C_B		300	ns
		Standard mode			1000	
Fall Time of Both SDA and SCL Signals (Note 11)	t_F	Fast mode	20 + 0.1C_B		300	ns
		Standard mode			300	
Setup Time for STOP Condition	$t_{SU:STO}$	Fast mode	0.6			µs
		Standard mode	4.7			
Capacitive Load for Each Bus Line	C_B	(Note 11)			400	pF
Capacitance for SDA, SCL	$C_{I/O}$			10		pF
Pulse Width of Spikes That Must Be Suppressed by the Input Filter	t_{SP}			30		ns
Pushbutton Debounce	PB_{DB}			250		ms
Reset Active Time	t_{RST}			250		ms
Oscillator Stop Flag (OSF) Delay	t_{OSF}	(Note 12)		100		ms
Temperature Conversion Time	t_{CONV}			125	200	ms

Power-Switch Characteristics

($T_A = T_{MIN}$ to T_{MAX})

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{CC} Fall Time; $V_{PF(MAX)}$ to $V_{PF(MIN)}$	t_{VCCF}		300			µs
V_{CC} Rise Time; $V_{PF(MIN)}$ to $V_{PF(MAX)}$	t_{VCCR}		0			µs
Recovery at Power-Up	t_{REC}	(Note 13)		250	300	ms

LAMPIRAN D
DAFTAR RIWAYAT PENULIS

Nama	:	Sanynita Kiskindy
TTL	:	Mataram, 22 Agustus 1994
Jenis Kelamin	:	Perempuan
Agama	:	Islam
Alamat Rumah	:	Jl. Merdeka XI/12 Pagesangan Baru-Mataram
E-mail	:	kiskindy@gmail.com
Hobi	:	Bersepeda

RIWAYAT PENDIDIKAN

2000 – 2006	:	SD Negeri 41 Mataram
2006 – 2009	:	SMP Negeri 2 Mataram
2009 – 2012	:	SMA Negeri 1 Mataram
2012 – sekarang	:	D3 Teknik Elektro-ITS Surabaya

PENGALAMAN ORGANISASI

1. Sekretaris Departemen Kominfo HIMAD3TEKTRO 2013/2014
2. Asisten Sekretaris Direktur BSO IECC BEM ITS 2013/2014
3. Kabiro Info dan Redaksi Departemen Kominfo HIMAD3TEKTRO 2014/2015

Halaman ini sengaja dikosongkan

DAFTAR RIWAYAT PENULIS

Nama	:	Indra Insan Prasetyo
TTL	:	Probolinggo, 9 Januari 1994
Jenis Kelamin	:	Laki - Laki
Agama	:	Islam
Alamat Rumah	:	Jl. Gatot Subroto 109 Probolinggo
E-mail	:	indra_inpra@yahoo.co.id
Hobi	:	Bermain Drum

RIWAYAT PENDIDIKAN

2000 – 2006	:	SDN Mangunharjo 1 Probolinggo
2006 – 2009	:	SMP Negeri 1 Probolinggo
2009 – 2012	:	SMA Negeri 1 Probolingo
2012 – sekarang	:	D3 Teknik Elektro-ITS Surabaya

PENGALAMAN ORGANISASI

1. Sie Pendidikan Komunitas Kampus Peduli Surabaya 2012/2013
2. Koor Pendidikan Komunitas Kampus Peduli Surabaya 2013/2014
3. Staff PSDM HIMAD3TEKTRO 2013/2014
4. Staff Sosmas BEM ITS 2013/2014
5. Ketua HIMAD3TEKTRO 2014/2015

Halaman ini sengaja dikosongkan