

BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

RANCANG BANGUN DAN MONITORING KETINGGIAN AIR DAN

KEKERUHAN BERBASIS *ARDUINO MEGA 2560* DENGAN

MENGGUNAKAN VTSCADA

Proses pembuatan benda kerja alat Tugas Akhir yang berjudul “RANCANG BANGUN DAN MONITORING KETINGGIAN AIR DAN KEKERUHAN BERBASIS *ARDUINO MEGA 2560* DENGAN MENGGUNAKAN VTSCADA)” ini, dibagi menjadi 2 bagian yaitu:

1. Pembuatan perangkat keras (*hardware*)

Pada tahap pertama ini meliputi semua proses pembuatan perangkat keras untuk merealisasikan hasil dari rancangan yang telah dibuat menjadi sistem yang siap dioperasikan.

2. Pembuatan perangkat lunak (*software*)

Dan untuk tahap kedua ini yaitu mencakup semua hal yang berkaitan dengan perangkat lunak bagi sistem. Pada masing-masing bagian mempunyai tujuan yang sama yaitu agar kedua bagian yang merupakan satu kesatuan sistem yang akan dibuat dapat saling melengkapi satu sama lain sehingga tercipta suatu sistem yang baik. Tetapi langkah awal dari pembuatan alat tugas akhir ini adalah perencanaan yang matang dan konsep yang jelas tentang

aplikasi apa yang akan dibuat. Agar kendala-kendala yang tidak diinginkan pada proses pembuatan dapat diperhitungkan terlebih dahulu.

4.1. Proses Pembuatan Perangkat Keras (hardware)

Pembuatan perangkat keras meliputi 2 bagian, yaitu pembuatan perangkat elektronika dan mekanik.

1. Pembuatan perangkat elektronika

Pembuatan perangkat elektronika ini merupakan pembuatan semua sistem yang berkaitan dengan perakitan elektronika yang meliputi perencanaan rangkaian, percobaan sementara, pembuatan rangkaian, serta pemasangan komponen.





2. Pembuatan bagian mekanik

Yang meliputi perencanaan bagian mekanik, pembuatan kerangka (tempat) untuk rangkaian, perakitan modul rangkaian pada kerangka (tempat) rangkaian, dan pembuatan label petunjuk penggunaan.








Pada pembuatan perangkat keras (*hardware*) ini dibutuhkan peralatan dan bahan-bahan untuk mendukung proses pengerjaannya. Tabel 4.1 4.2 menunjukkan daftar alat dan bahan-bahan yang dibutuhkan dalam membuat alat Tugas Akhir ini. Tabel 4.1 dibawah ini adalah Daftar Alat Pembuatan Bagian *Hardware*.

4.1.1 Alat Dan Bahan






Tabel 4.1 Daftar Alat Pembuatan Bagian *Hardware*

NO.	NAMA	GAMBAR	SPESIFIKASI	JUMLAH
1.	Pensil		Staedler 2B	1 buah
2.	Pengapus Pensil		Staedler 2B	1 buah
3.	Spidol Permanent		Snaowman G-12	1 buah (Warna Merah)
4.	Penggaris		Mika Butterfly	1 buah (Ukuran 30cm)

Lanjutan Tabel 4.1

5.	Multimeter		Digital Sanwa	1 buah
6.	Solder dan Blower		Solderingiron SSOI	1 set
7.	Pasta Solder		Lotfett	12 gram
8.	Atraktor (penyedot timah)		EDEN	1 buah
9.	Tang Pengupas Kabel		Nakai Model V	1 buah
10.	Cutter		Kenko L500	1 buah
11.	Obeng Kombinasi		NO.JK – 6032D	1 Set







Lanjutan Tabel 4.1

12.	Mesin Bor Tangan		KENMASTER Mesin Bor Bolak Balik + Mata Bor Besi Kayu Beton + Sekrup	1 set
13.	Gergaji Kayu		Tekiro	1 buah
14.	Pemotong kaca		Glass Cutter Finder	1 set
15.	Palu		Palu Kecil Tekiro	1 buah
16.	Paku		Paku 2inchi	Secukupnya








Tabel 4.2 Daftar Bahan-Bahan Pembuatan Bagian *Hardware*

NO.	NAMA	GAMBAR	SPESIFIKASI	UKURAN
1.	klem		Klem pipa	Tebal 1"



Lanjutan Tabel 4.2

2.	Isolasi		NATIONAL TAPE	Tebal 3mm
3.	Selang			3/8mm
4.	Lem Kaca		Sealent	1pcs
5.	Kayu Triplek		Multiplek	Ukuran 9 mm
6.	Sekrup		Sekrup Kayu	Secukupnya
7.	Spiral Kabel		KSS KS-10	Secukupnya

Lanjutan Tabel 4.2

8.	Kabel Tie		KSS CV-100	Secukupnya
9.	Kabel Jack DC arduino		Hard Plastic T- type Connector	2,1x5,5mm
10.	Kabel		Kabel Pelangi	Secukupnya
11.	PIN Header		Konektor <i>Female</i> dan <i>male</i>	Secukupnya
12.	Timah Solder		Tenol Paragon 100 M 0,8mm	Secukupnya
13.	Amplas		Amplas Serbaguna	Secukupnya
14.	Kayu			2,5x0,5 cm 1,8x1,8 cm

Lanjutan Tabel 4.2

15.	Botol		Botol bekas lemon water	1 buah
16.	Filter aquarium		Ultra Filter Aquarium	1 buah
17.	Y pipa		Brass Tee Connectors Barbed	10mm
18.	T pipa		Brass Tee Connectors Barbed	10mm

4.1.2 Pembuatan Perangkat Elektronika

Pembuatan perangkat elektronika ini terdiri atas beberapa langkah yaitu:

4.1.2.1 Perencanaan Rangkaian

Dalam perencanaan rangkaian ini dilakukan untuk mendapatkan rangkaian sesuai dengan yang dibutuhkan. Hal ini dilakukan dengan mencari data-data tentang prinsip dasar dari komponen utama dan komponen bantu yang akan

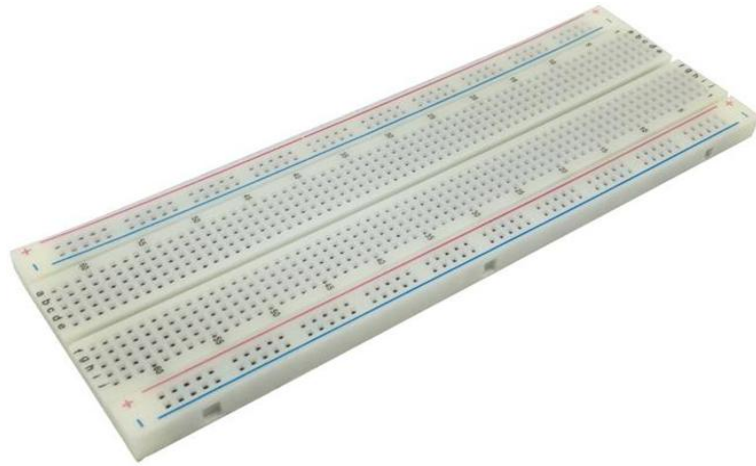
digunakan dalam rangkaian serta menentukan komponen-komponen yang akan digunakan. Setelah itu membuat gambar skema rangkaian baik untuk per-modul ataupun rangkaian sistem secara keseluruhan.

Pembuatan bagian elektronika terdiri atas beberapa langkah yaitu perencanaan rangkaian, percobaan sementara, pembuatan rangkaian pada papan *Printed Circuit Board* (PCB), serta pemasangan komponen. Dalam perencanaan ini terdapat dua rangkaian utama, yaitu:

4.1.2.2 Percobaan Sementara

Dalam pembuatan system ataupun rangkaian pada tugas akhir ini memerlukan rangkaian yang sesuai dan benar dengan spesifikasi yang diinginkan sehingga rangkaian dapat berjalan sesuai dengan yang diinginkan. Untuk itu perlu adanya percobaan dan penelitian terlebih dahulu untuk mendapatkan rangkaian tersebut. Setelah rangkaian dirancang sesuai dengan skema, maka rangkaian tersebut perlu di uji coba untuk mengetahui bagaimanakah rangkain yang sudah di rangkai dan apakah rangkaian dapat bekerja sesuai dengan apa yang diinginkan. Percobaan dilakukan dengan menggunakan papan rangkaian percobaan (*proto board/project board*) terlebih dahulu.

Gambar 4.1 adalah *proto board* yang digunakan untuk percobaan sementara pada alat simulasi ini



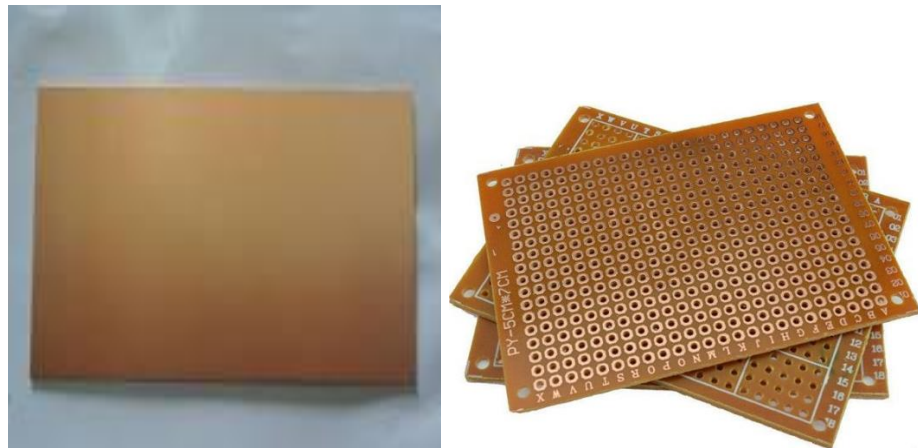
Gambar 4.1 *Proto Board (Project Board)*

Pada proses ini dilakukan agar jika terjadi kesalahan pada pemasangan komponen atau hasil keluaran tidak sesuai dengan keinginan, komponen dapat diganti dengan mudah. Setelah rangkaian dan hasilnya sesuai dengan yang dikehendaki baru dibuat pola Papan Rangkaian.

4.1.2.3 Pembuatan Rangkaian

Langkah selanjutnya setelah percobaan rangkaian sistem adalah pembuatan rangkaian sesungguhnya setelah mengetahui susunan rangkaian yang benar. Dalam langkah ini digunakan sebuah papan sirkuit untuk tempat penyambungan agar antara komponen dapat terhubung sesuai gambar rangkaian yang telah dibuat sebelumnya. Papan sirkuit yang digunakan adalah *printed circuit board* (PCB).

Gambar 4.2 adalah jenis PCB yang digunakan pada simulasi alat ini.



Gambar 4.2 *printed circuit board PCB*

Dengan menggunakan jenis PCB ini pekerjaan dalam pembuatan alat tugas akhir ini akan lebih mudah dan membantu.

4.1.2.4 Pemasangan Komponen

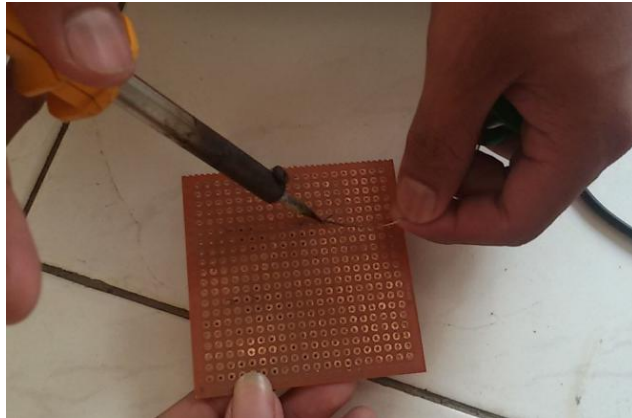
Tahapan terakhir ini adalah memasang komponen yang telah dibuat oleh penulis. Pemasangan ini digunakan agar rangkaian yang telah dibuat, menjadi sebuah modul yang dapat digunakan dibagian proyek tugas akhir yang sedang dikerjakan.

Diantara pemasangan tersebut adalah untuk memasang catu daya (*power supply*) dan komparator yang akan digunakan. Berikut adalah langkah-langkah pemasangannya:

A. Pemasangan Komponen Catu Daya

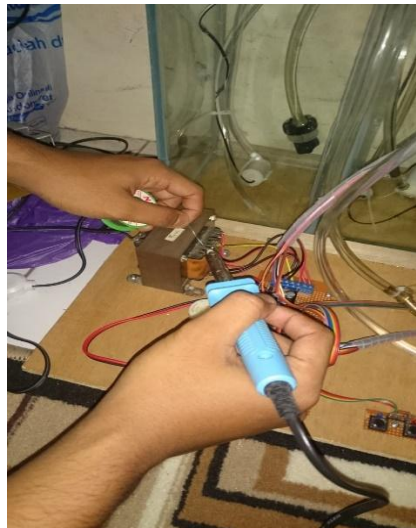
- Memasang dan menyolder komponen-komponen aktif, yaitu diode, kapasitor.

Perlu diperhatikan bahwa pada saat pemasangan komponen ini posisi kaki-kakinya tidak boleh tertukar atau salah posisi pada PCB lubang. **Gambar 4.3** menunjukkan proses pemasangan dan penyolderan komponen pada PCB lubang.



Gambar 4.3 Pemasangan dan Penyolderan Komponen Aktif

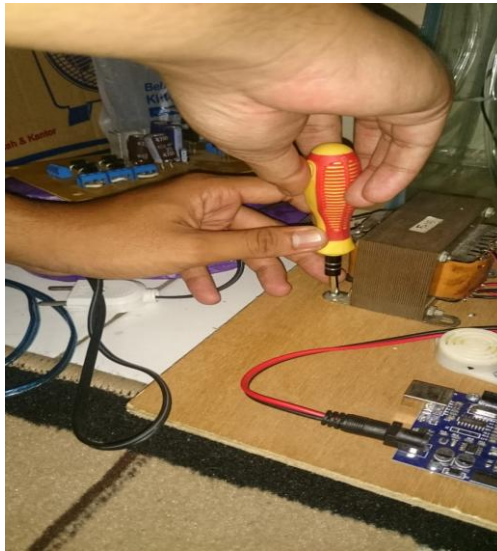
- Penyambungan komponen dioda dengan transformator 5A. Dimana 12V pada trafo untuk rangkaian 12VDC dan 5V pada trafo untuk rangkaian 5VDC, serta CT sebagai *ground* untuk rangkaian 12VDC dan 5VDC. **Gambar 4.4** menunjukkan proses penyambungan/penyolderan rangkaian dengan transformator.



Gambar 4.4 Penyambungan dengan Transformator

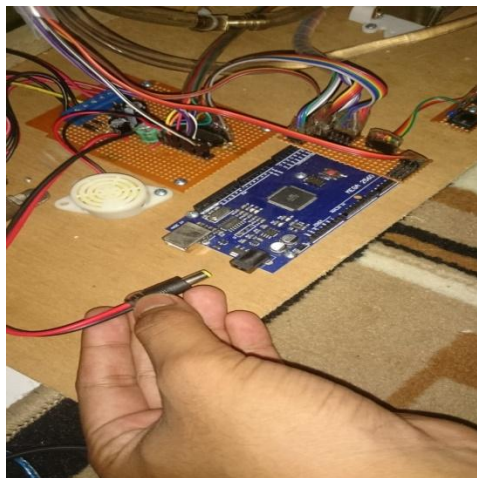
- Pemasangan semua komponen dan transformator pada *box*, sebagai modul *power supply*. Diukur terlebih dahulu untuk letak trafo agar tertata rapih.

Gambar 4.5 menunjukkan proses pemasangan trafo pada box.



Gambar 4.5 Pemasangan Transformator

- Pemasangan kabel *input* untuk kabel jack dc arduino mega 2560. **Gambar 4.6** menunjukkan proses pemasangan kabel jack sebagai input dari arduino mega2560.



Gambar 4.6 Pemasangan Kabel Input Arduino mega2560

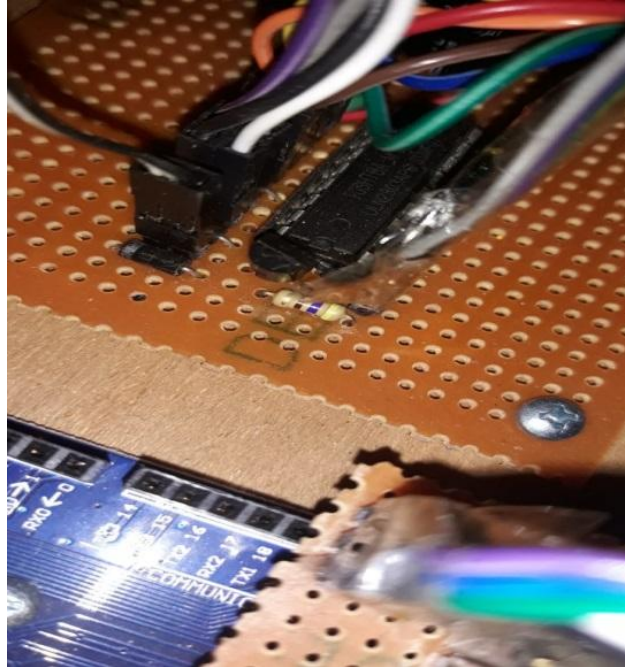
Tabel 4.3 Komponen Power Supply

NO.	KOMPONEN	SPESIFIKASI	JUMLAH/UKURAN
1.	Transformator	Merk "King" 5 A	1 buah
2.	Dioda	2 A 3 A	3 buah
3.	Kapasitor	1000 μ F / 35 V 2200 μ F / 35 V 4700 μ F / 35 V	1 buah
4.	Jack DC	2,1x5,5mm	1 buah
5.	Kabel serabut	1 mm	secukupnya
6.	Mur/baut halus	2mm	secukupnya
7.	Baut kasar	1 cm	Secukupnya

B. Rangkaian IC ULN 2803

Rangkaian IC ULN 2803 ini mendapat *input* tegangan 12 VDC. Sumber tersebut berasal dari catu daya 12 VDC yang nantinya akan digunakan untuk mengontak komponen yang memerlukan tegangan 12VDC. Karena pin yang dipakai cukup banyak maka dalam rangkaian ini menggunakan IC ULN 2803 dalam perangkaiannya. Adapun gambar dan daftar komponen rangkaian IC ULN 2803.

Gambar 4.7 merupakan gambar rangkaian IC ULN 2803 yang telah dirangkai.



Gambar 4.7 Rangkaian *Driver* IC ULN2803 yang Telah Dirangkai

C. Rangkaian *Push Button*

Rangkaian *push button* ini menggunakan rangkaian *debouncer* yang mendapat input 5 VDC. Sumber tersebut berasal dari pin *VCC* arduino dan kemudian memberi input ke pin *digital* arduino. Daftar komponen rangkaian *debouncer* ditunjukkan pada tabel 4.4 Gambar 4.8 merupakan gambar rangkaian *debouncer* yang telah dirangkai.

Rangkaian *push button* ini digunakan untuk menghidupkan alat me-reset alarm jika terjadi kebocoran. Untuk menghidupkan dan me-reset alarm alat bersumber dari pin 41- 49 digital dari arduino. Daftar komponen rangkaian *push button* ditunjukkan pada tabel 4.4 .

Tabel 4.4 Daftar Komponen Rangkaian *push button*

No	Nama Alat	Spesifikasi	Jumlah
1	Resistor	1 k Ω	2 buah
2	Lampu Led	Biru Dan Merah	2 buah
4	Micro Switch Push Button	4 kaki	2 buah

Gambar 4.8 merupakan rangkaian *push button* yang telah di rangkai.



Gambar 4.8 Rangkaian *push button* yang telah dirangkai

4.2 Pembuatan Program

Penulisan program yang digunakan penulis dalam alat Tugas Akhir “RANCANG BANGUN MONITORING KETINGGIAN AIR DAN KEKERUHAN PADA PDAM BERBASIS *ARDUINO MEGA 2560* DENGAN TAMPILAN SISTEM *SCADA* DAN *HUMAN MACHINE INTERFACE (HMI)*” ini dilaksanakan setelah diagram alur selesai dirancang. Dalam sistem ini terdapat dua program, yaitu Program Arduino dan Program HMI. Perangkat lunak yang

digunakan dalam pembuatan program Arduino adalah bahasa pemrograman C, untuk HMI adalah VTScada.

4.2.1 Program Arduino Mega 2560

Dalam pembuatan program Mikrokontroller dari Arduino Mega2560 ini, perangkat lunak yang digunakan adalah Arduino. Langkah kerja untuk membuat program yaitu:

1. Menyiapkan alat dan bahan yang diperlukan.
2. Kemudian sambungkan pada pin yang tersedia pada arduino mega 2560.

Tabel 4.5 dibawah ini pemasangan pin pada arduino mega 2560.

Tabel 4.5 pemasangan pin pada arduino mega 2560

PIN	Keterangan
A15	Sensor Kekeruhan
D19	Sensor Water Flow Meter 3
D20	Sensor Water Flow Meter 2
D21	Sensor Water Flow Meter 1
D22	Sensor Ultrasonic 1 (Trig)
D23	Sensor Ultrasonic 1 (Echo)
D24	Sensor Ultrasonic 2 (Trig)
D25	Sensor Ultrasonic 2 (Echo)
D26	Sensor Ultrasonic 3 (Trig)
D27	Sensor Ultrasonic 2 (Echo)
D28	Buzzer

Lanjutan Tabel 4.5

D29	Solenoid On/Off Valve 3
D30	Solenoid On/Off Valve 2
D31	Solenoid On/Off Valve 1
D32	Pompa 2
D33	Pompa 1
D34	Pompa 3
D41 - 49	Push Botton On dan Reset

3. Buka software Arduino untuk memulai program:
 - a. Klik kanan pada icon Arduino lalu pilih open atau bisa juga double klik pada icon Arduino. **Gambar 4.9** merupakan tampilan software arduino.

**Gambar 4.9** Tampilan Software Arduino

- b. Tampilan pertama saat membuka aplikasi tersebut.

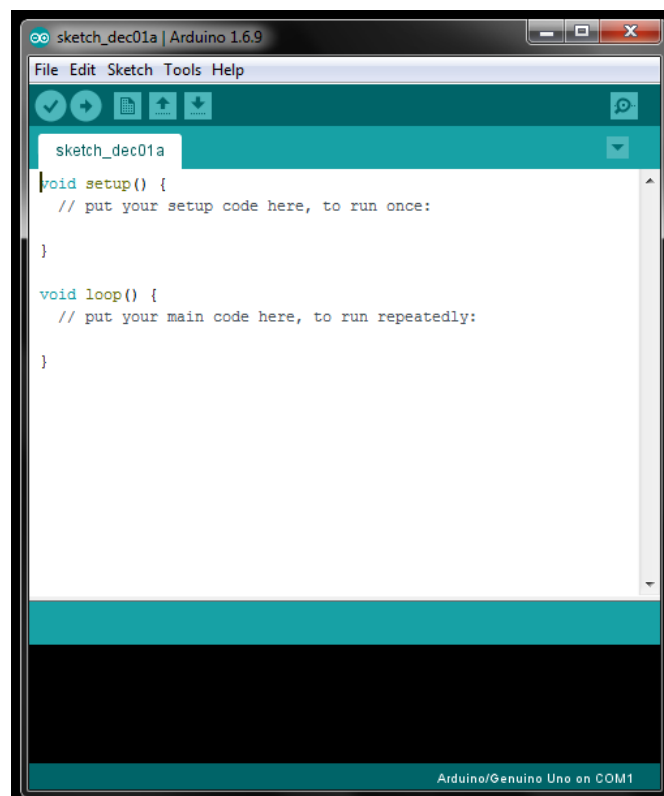
Gambar 4.10 merupakan gambar tampilan awal pada software arduino.




Gambar 4.10 Tampilan Software Arduino setelah dibuka

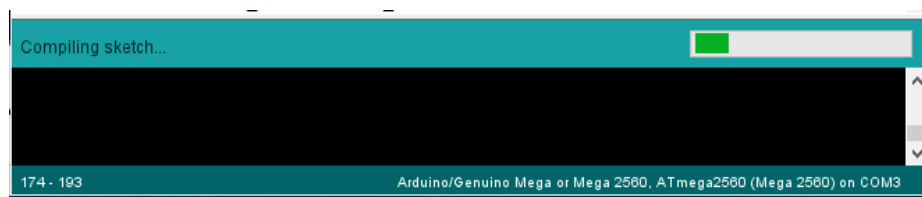
- c. Setelah itu, muncul jendela menu utama pada layar desktop seperti dibawah ini.

Gambar 4.11 merupakan tampilan saat membuat program pada software arduino.



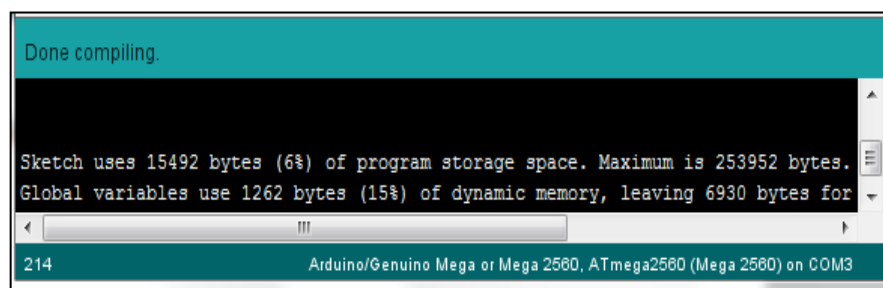
Gambar 4.11 Tampilan untuk mengisi program Software Aduino

- d. Setelah program selesai ditulis, lakukan *compiling* dengan menekan tombol  sehingga muncul proses pada gambar 4.12. Hal itu dilakukan untuk mengecek kebenaran *sketch*. Tunggulah beberapa detik, apabila *sketch* sudah benar maka akan muncul pemberitahuan seperti pada gambar 4.12.



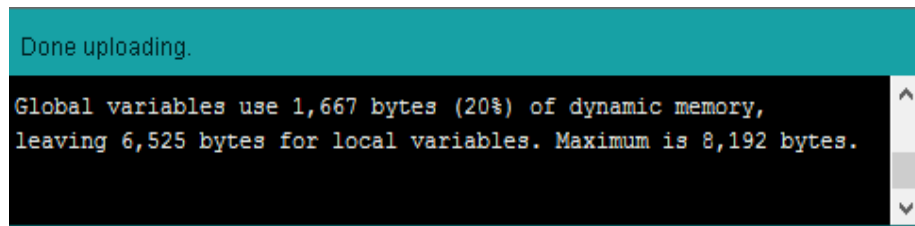
Gambar 4.12 Proses *Compiling*

Gambar 4.13 merupakan program yang berhasil di *uploading*.



Gambar 4.13 *Compiling* Berhasil

Lakukan *uploading* program ke mikrokontroler dengan menekan tombol . Setelah *uploading* selesai, maka akan muncul pemberitahuan seperti pada gambar 4.14.



Gambar 4.14 *Uploading* Berhasil

4. Kemudian, ketik program yang sudah kita siapkan di software tersebut.

Dibawah ini adalah program keseluruhan alat yang telah selesai dibuat.

```
#define PC Serial
```

```
#define FiltTank 0
```

```
#define cityTank_1 1
```

```
#define cityTank_2 2
```

```
#define StorageTank 1
```

```
#define ReFiltTank 2
```

```
#define bayPass 0
```

```
#define turbSens A15
```

```
const uint32_t samplingPeriode = 100;

const uint32_t maxError = 100;           // nilai selisih WaterFlow In &
Out indikasi kebocoran

const uint32_t tout_bocor = 3000;       // waktu memastikan kebocoran
(milisecond)

char buf[100];

uint32_t t,

    t_bocor,

    t_sample;

const int INTch[] = {2, 3, 4};

const int INTpin[] = {21,20,19};

uint32_t c_[3],

    t_[3],

    c_copy_N[3],
```

```
t_copy_N[3],

c_copy_L[3],

t_copy_L[3],

c_flow[3],

t_flow[3],

rate[3];

int32_t errorFlow;

float filtError;

byte penuh[3];

boolean bocor;

long jarak[3];

float fJarak[3];      // FT   CT1   CT2      // kosong = tinggi, penuh =
rendah

const float jarakPenuh[3] = {100.0, 100.0, 100.0}; // pengaturan nilai tangki
kosong

const float jarakKosong[3] = {300.0, 300.0, 300.0}; // pengaturan nilai tangki
penuh
```

```
#define batasKeruh 300 // keruh = rendah, jernih = tinggi

#define batasJernih 400

float kekeruhan;

boolean keruh;

#define buzzer 33

int buzcount;

const int selPin[] = {31, 29, 28};

const int pumpPin[] = {30, 32, 34};

float LPF(float N,float L,float sm) {

    return (N*sm) + (L*(1.0-sm));

}
```



```
void EISR_0(void) {c_[0]++; t_[0]=millis();}
```

```
void EISR_1(void) {c_[1]++; t_[1]=millis();}
```

```
void EISR_2(void) {c_[2]++; t_[2]=millis();}
```

```
void (*isrFunc[3])(void) = {EISR_0,EISR_1,EISR_2};
```

```
const int echoPin[] = {22, 24, 26};           // deklarasi pin echo masing-masing
```

```
SR04
```

```
const int trigPin[] = {23, 25, 27};         // deklarasi pin triger masing-masing
```

```
SR04
```

```
//----- subrutin pengukuran SR04
```

```
long readSR04(int trig_,int echo_){
```

```
long duration;
```

```
digitalWrite(trig_, LOW); // pin triger LOW
```

```
delayMicroseconds(5); // tunda 5 mikrosecond
```

```
digitalWrite(trig_, HIGH); // pin triger HIGH
```

```
delayMicroseconds(10); // tunda 10 mikrosecond
```

```
digitalWrite(trig_, LOW); // pin triger LOW

duration = pulseIn(echo_, HIGH); // baca panjang pulsa HIGH pin echo

return duration / 5;

}

void initSR04(int trig_,int echo_){ // inialisasi pin SR04

    pinMode(trig_, OUTPUT); // pin triger sebagai output

    pinMode(echo_, INPUT); // pin eco sebagai input

}

boolean sistemON;

#define LED_sys 49

#define TBL_sys 47

#define ctrl_com 45
```

```
#define TBL_bocor 43
```

```
#define LED_bocor 41
```

```
void setup() {
```

```
  int ch;
```

```
  PC.begin (9600);
```

```
  for(ch=0; ch<3;ch++) {
```

```
    pinMode(INTpin[ch],INPUT_PULLUP);
```

```
    attachInterrupt(INTch[ch], isrFunc[ch], RISING);    // set External Interupt
```

```
  }
```

```
  for(ch=0; ch<3; ch++) initSR04(trigPin[ch],echoPin[ch]); // inisialisasi SR04
```

```
  for(ch=0; ch<3;ch++) {
```

```
    pinMode(selPin[ch],OUTPUT);    // inisialisasi selenoid
```

```
pinMode(pumpPin[ch],OUTPUT);           // inisialisasi pompa
}

pinMode(INTpin[ch],INPUT_PULLUP);

pinMode(buzzer,OUTPUT);                // inisialisasi buzzer

pinMode(TBL_sys,INPUT_PULLUP);         // inisialisasi tombol

pinMode(TBL_bocor,INPUT_PULLUP);

pinMode(ctrl_com,OUTPUT);              // inisialisasi COMON

pinMode(LED_sys,OUTPUT);               // inisialisasi LED

pinMode(LED_bocor,OUTPUT);

digitalWrite(ctrl_com,LOW);

t = millis();

t_sample = t;

}
```

```
void loop() {  
  
    int ch;  
  
    if(!sistemON) {if(digitalRead(TBL_sys)==LOW) sistemON = true;}  
  
    if(bocor)    {if(digitalRead(TBL_bocor)==LOW) bocor = false;}  
  
  
    digitalWrite(LED_sys, sistemON);  
  
    digitalWrite(LED_bocor, bocor);  
  
  
    t = millis();  
  
    if((t-t_sample)>samplingPeriode) {  
  
        t_sample = t;  
  
  
        for(ch=0; ch<3; ch++) {  
  
            c_copy_L[ch] = c_copy_N[ch];  
  
            t_copy_L[ch] = t_copy_N[ch];  
  
        }  
    }  
}
```

```

for(ch=0; ch<3; ch++) {

    c_copy_N[ch] = c_[ch];

    t_copy_N[ch] = t_[ch];

}

for(ch=0; ch<3; ch++) {

    c_flow[ch] = c_copy_N[ch] - c_copy_L[ch];

    t_flow[ch] = t_copy_N[ch] - t_copy_L[ch];

    if(t_flow[ch]!=0) {

        // flow(mL/mnt) = freq * resolusi * 1 * 1

        // pulse * 1(L) * 60000(ms) * 1000(mL)

        // -----

        // periode(ms) * 5880 * 1(mnt) * 1(L)

        uint32_t Num = c_flow[ch] * 500000L;

        uint32_t Den = t_flow[ch] * 49L;

        rate[ch] = Num / Den;

    }

    else rate[ch] = 0;

```

```

}

errorFlow = rate[FiltTank] - rate[cityTank_1] - rate[cityTank_2];

filtError = LPF((float)errorFlow,filtError,0.01);

if(filtError<maxError) {

    t_bocor = t;

}

else if((t-t_bocor)>=tout_bocor) {

    bocor = true;

    t_bocor = t;

}

for(ch=0; ch<3; ch++) {

    jarak[ch] = readSR04(trigPin[ch],echoPin[ch]);    // baca jarak pada gate
yang sedang dipantau

    fJarak[ch] = LPF((float)jarak[ch],fJarak[ch],0.1);

    if(fJarak[ch]<jarakPenuh[ch])    penuh[ch] = 1;

    else if(fJarak[ch]>jarakKosong[ch]) penuh[ch] = 0;

```

```
}
```

```
kekeruhan = LPF((float)analogRead(turbSens),kekeruhan,0.1);
```

```
if(kekeruhan<batasKeruh)    keruh = true;
```

```
else if(kekeruhan>batasJernih) keruh = false;
```

```
if(sistemON) {
```

```
    if(penuh[FiltTank]==1) {
```

```
        digitalWrite(pumpPin[StorageTank],LOW);
```

```
        if(keruh) {
```

```
            digitalWrite(pumpPin[FiltTank],LOW);
```

```
            digitalWrite(selPin[bayPass],LOW);
```

```
            digitalWrite(selPin[cityTank_1],LOW);
```

```
            digitalWrite(selPin[cityTank_2],LOW);
```

```
        }
```

```
    } else {
```

```
        digitalWrite(pumpPin[FiltTank],HIGH);
```



```
if(bocor) {

    digitalWrite(selPin[cityTank_1],LOW);

    digitalWrite(selPin[cityTank_2],LOW);

    if((penuh[cityTank_1]==1)|(penuh[cityTank_2]==1)) {

        digitalWrite(selPin[bayPass],LOW);

        sistemON = false;

    }

    else digitalWrite(selPin[bayPass],HIGH);

}

else {

    digitalWrite(selPin[bayPass],LOW);

    for(ch=cityTank_1; ch<=cityTank_2; ch++) {

        if(penuh[ch]==1) digitalWrite(selPin[ch],LOW);

        else digitalWrite(selPin[ch],HIGH);

    }

    if((penuh[cityTank_1]==1)&(penuh[cityTank_2]==1)) sistemON = false;

}
```

```
    }  
  
  }  
  
  else {  
  
    digitalWrite(pumpPin[StorageTank],HIGH);  
  
    digitalWrite(pumpPin[FiltTank],LOW);  
  
    digitalWrite(selPin[bayPass],LOW);  
  
    digitalWrite(selPin[cityTank_1],LOW);  
  
    digitalWrite(selPin[cityTank_2],LOW);  
  
  }  
  
  
  digitalWrite(pumpPin[ReFiltTank],(keruh) ? HIGH : LOW);  
  
  if(bocor) {  
  
    if(buzcount==0)    digitalWrite(buzzer,HIGH);  
  
    else if(buzcount==5) digitalWrite(buzzer,LOW);  
  
    buzcount++;  
  
    if(buzcount>=10) buzcount = 0;  
  
  }
```

```
    else digitalWrite(buzzer,LOW);

}

else {

    digitalWrite(pumpPin[StorageTank],LOW);

    digitalWrite(pumpPin[FiltTank],LOW);

    digitalWrite(pumpPin[ReFiltTank],LOW);

    digitalWrite(selPin[bayPass],LOW);

    digitalWrite(selPin[cityTank_1],LOW);

    digitalWrite(selPin[cityTank_2],LOW);

    digitalWrite(buzzer,LOW);

    keruh = false;

    bocor = false;

    buzcount = 0;

}

// serial monitor

PC.print(filtError,0);  PC.write('\t');           // eror debit
```

```
for(ch=0; ch<3; ch++) {  
  
    PC.print(fJarak[ch],0);  PC.write('\t');           // jarak sensor - air  
  
}  
  
PC.print(kekeruhan,0);  PC.write('\t');           // kekeruhan  
  
PC.print(sistemON,DEC);  PC.write('\t');           // sistem ON  
  
PC.print(bocor,DEC);  PC.write('\t');           // kebocoran  
  
PC.write('[');  
  
for(ch=0; ch<3; ch++) {  
  
    PC.print(digitalRead(pumpPin[ch]),DEC);  PC.write(',');  // status pompa  
  
}  
  
PC.write(']');  
  
PC.write('{');  
  
for(ch=0; ch<3; ch++) {  
  
    PC.print(digitalRead(selPin[ch]),DEC);  PC.write(',');  // status selenoid  
  
}  
  
PC.write('}');  
  
PC.write('\n');
```

}

}

4.2.2 Pembuatan Tampilan dan Penyambungan *Monitoring* menggunakan VTSCADA

Software VTScada digunakan untuk membuat desain tampilan monitoring dan pengalamatan peralatan untuk proses kontroling peralatan.

Langkah-langkah mengoperasikan aplikasi VTScada adalah sebagai berikut:

- A. Membuka Aplikasi VTScada yang sudah terinstal pada Personal Computer.



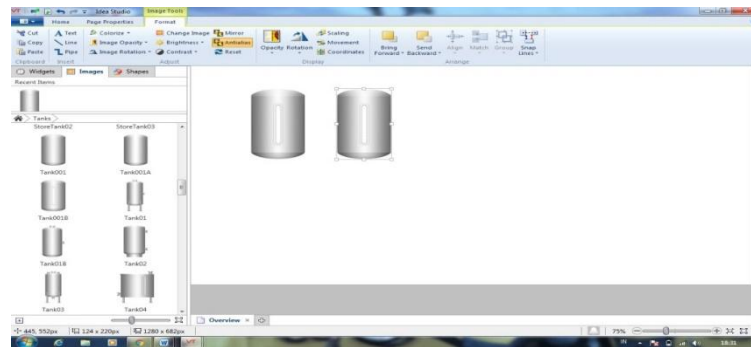
Gambar 4.15 Membuka VTScada

- B. Kemudian akan muncul tampilan seperti gambar 4.16, dan pilih folder yang sudah dibuat desainnya.



Gambar 4.16 Membuka VTScada

1. Kemudian pilih file yang akan dibuat lalu akan muncul tampilan edit, lalu pilih *shapes* ,pilih gambar bulat dan menggambar *single line diagram* seperti gambar di bawah ini.



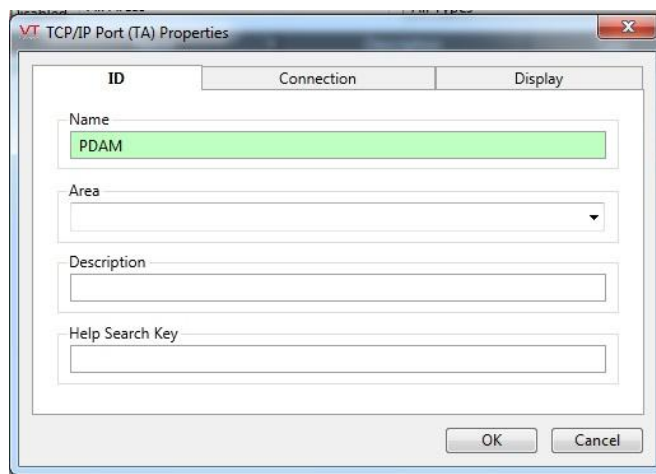
Gambar 4.17 Tampilan Untuk Membuat Single Line Diagram Pada VT Scada

2. Setelah selesai membuat single line diagram sekarang dibuat monitor sensor kekeruhan, sensor *water flow meter*, sensor *ultrasonic*, dengan tampilan digital agar sama persis dengan alat simulator.

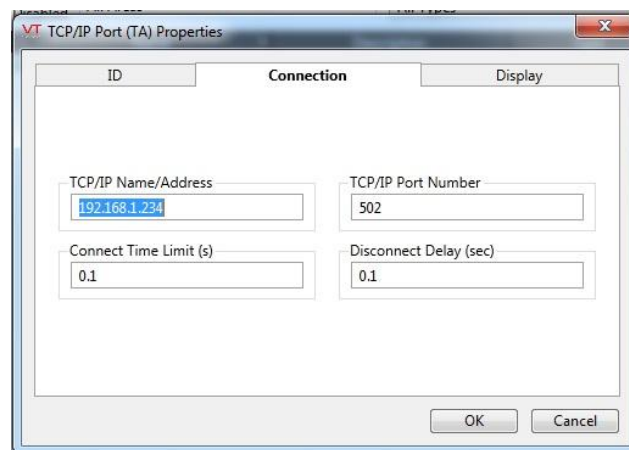


Gambar 4.18 Tampilan Lengkap Single Line Diagram Pada VT Scada

- Langkah selanjutnya adalah membuat New Child pada VTScada, hal ini bertujuan untuk menyambungkan VTScada dengan HMI. Maka hal pertama yang dilakukan adalah memberi nama pada New Child yang akan dibuat. Setelah itu memilih TCP/IP Port memasukkan, alamat TCP/IP yang sudah ditentukan.

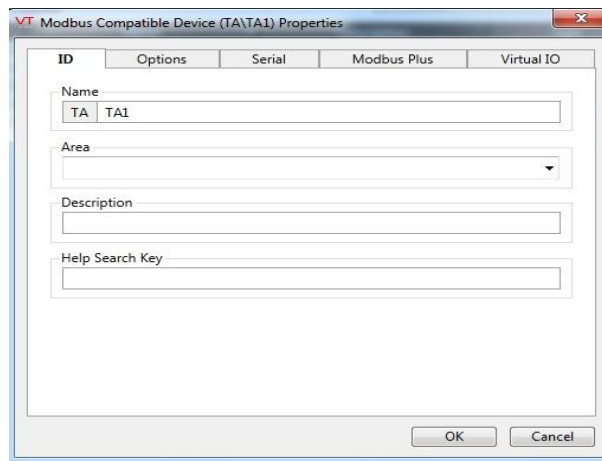


Gambar 4.19 Tampilan ID TCP/IP Port

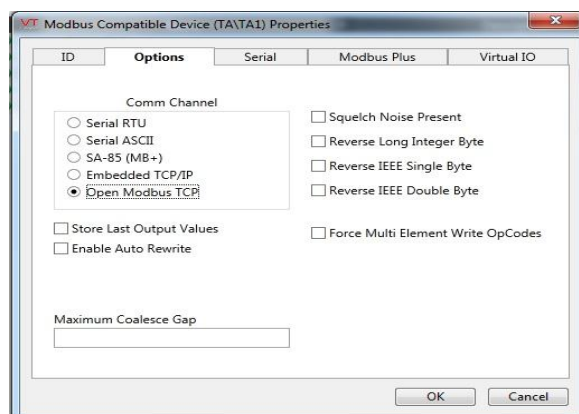


Gambar 4.20 Tampilan Connection TCP/IP Port

- Setelah folder new child td dibuat dengan memasukan koneksi alamat IP yang sesuai dengan alamat ip yang ada di program Arduino, langkah selanjutnya adalah membuat folder baru di dalam new child yang ada tcp ip tadi dengan memilih Modbus compatible device. Setelah memilih Modbus compatible device langkah selanjutnya adalah memasukkan nama sesuai keinginan sebagai nama Modbus dan setelah itu memilih koneksi yang berupa open Modbus TCP.

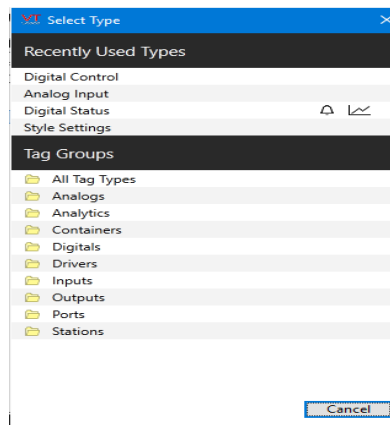


Gambar 4.21 Tampilan ID Modbus Compatible Device

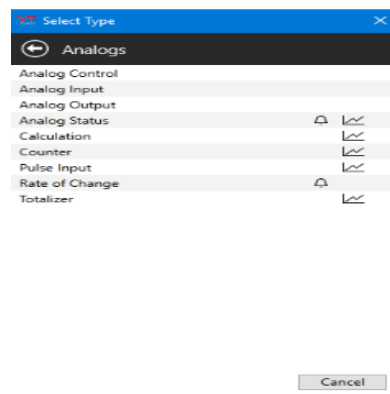


Gambar 4.22 Tampilan Options Modbus Compatible Device

5. Setelah membuat open Modbus TCP, untuk menyambungkan antara pembacaan sensor arus dan agar dapat ditampilkan pada layar HMI maka dibutuhkan new child lagi didalam Modbus compatible device, apabila ingin menampilkan pembacaan sensor arus maka pilih analog input pada menu yang ada pada VTScada.

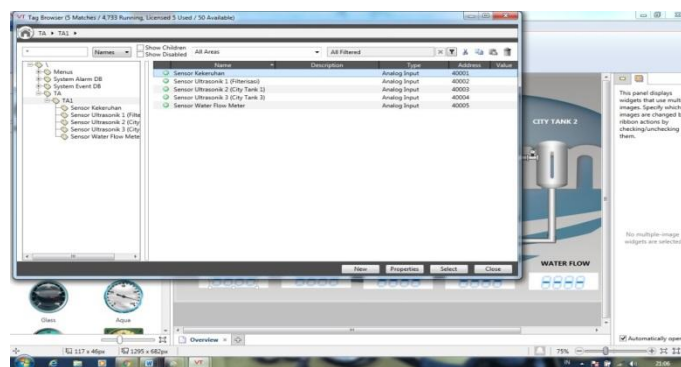


Gambar 4.23 Tampilan Pilihan Tag Untuk Sensor



Gambar 4.24 Tampilan Pilihan Tag Untuk Analog Input

6. Setelah memilih analog input sebagai tampilan pada VTScada untuk menampilkan pembacaan sensor, setelah itu pilih seven segmen untuk memberi tampilan berupa angka agar dapat dilihat pada tampilan layar monitor VTScada.



Gambar 4.25 Tampilan Pilihan Tag Untuk Analog Input

Dibawah ini adalah program dari komunikasi antara VTScada dengan Arduino yang sudah digabung dengan program Arduino.

```
#define PC Serial
```

```
#include <Mudbus.h>
```

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

```
#include <Ethernet.h>
```

```
Mudbus Mb;
```

```
#define FiltTank 0
```

```
#define cityTank_1 1
```

```
#define cityTank_2 2
```

```
#define StorageTank 1
```

```
#define ReFiltTank 2
```

```
#define bayPass 0
```

```
#define turbSens A15
```

```
const uint32_t samplingPeriode = 100;
```

```
const uint32_t maxError = 100; // nilai selisih WaterFlow In &  
Out indikasi kebocoran
```

```
const uint32_t tout_bocor = 3000; // waktu memastikan kebocoran  
(milisecond)
```

```
char buf[100];
```

```
uint32_t t,  
  
    t_bocor,  
  
    t_sample;  
  
  
const int INTch[] = {2, 3, 4};  
  
const int INTpin[] = {21,20,19};  
  
  
uint32_t c_[3],  
  
    t_[3],  
  
    c_copy_N[3],  
  
    t_copy_N[3],  
  
    c_copy_L[3],  
  
    t_copy_L[3],  
  
    c_flow[3],  
  
    t_flow[3],  
  
    rate[3];  
  
int32_t errorFlow;
```

```
float filtError;
```

```
byte penuh[3];
```

```
boolean bocor;
```

```
long jarak[3];
```

```
float fJarak[3];          // FT   CT1   CT2          // kosong = tinggi, penuh =  
rendah
```

```
const float jarakPenuh[3] = {100.0, 100.0, 100.0}; // pengaturan nilai tangki  
kosong
```

```
const float jarakKosong[3] = {300.0, 300.0, 300.0}; // pengaturan nilai tangki  
penuh
```

```
#define batasKeruh 300          // keruh = rendah, jernih = tinggi
```

```
#define batasJernih 400
```

```
float kekeruhan;
```

```
boolean keruh;
```

```
#define buzzer 33

int buzcount;

const int selPin[] = {31, 29, 28};

const int pumpPin[] = {30, 32, 34};

float LPF(float N,float L,float sm) {

    return (N*sm) + (L*(1.0-sm));

}

void EISR_0(void) {c_[0]++; t_[0]=millis();}

void EISR_1(void) {c_[1]++; t_[1]=millis();}

void EISR_2(void) {c_[2]++; t_[2]=millis();}

void (*isrFunc[3])(void) = {EISR_0,EISR_1,EISR_2};

const int echoPin[] = {22, 24, 26};           // deklarasi pin echo masing-masing

SR04
```

```
const int trigPin[] = {23, 25, 27};          // deklarasi pin triger masing-masing  
SR04
```

```
//----- subrutin pengukuran SR04
```

```
long readSR04(int trig_,int echo_){
```

```
long duration;
```

```
digitalWrite(trig_, LOW); // pin triger LOW
```

```
delayMicroseconds(5); // tunda 5 mikrosecond
```

```
digitalWrite(trig_, HIGH); // pin triger HIGH
```

```
delayMicroseconds(10); // tunda 10 mikrosecond
```

```
digitalWrite(trig_, LOW); // pin triger LOW
```

```
duration = pulseIn(echo_, HIGH); // baca panjang pulsa HIGH pin echo
```

```
return duration / 5;
```

```
}
```

```
void initSR04(int trig_,int echo_){ // inialisasi pin SR04
```

```
pinMode(trig_, OUTPUT);    // pin triger sebagai output

pinMode(echo_, INPUT);    // pin eco sebagai input

}
```

```
boolean sistemON;
```

```
#define LED_sys  49
```

```
#define TBL_sys  47
```

```
#define ctrl_com 45
```

```
#define TBL_bocor 43
```

```
#define LED_bocor 41
```

```
int cmd[10];
```

```
void setup() {
```

```
    //SETTING IP ADDRESS MODBUS//
```

```
    uint8_t mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // dirubah sesuai
```

```
    kesepakatan
```



```
uint8_t ip[] = { 192, 168, 1, 234 }; // dirubah sesuai kesepakatan

uint8_t gateway[] = { 192, 168, 1, 1 };

uint8_t subnet[] = { 255, 255, 255, 0 };

Ethernet.begin(mac, ip, gateway, subnet);

//Avoid pins 4,10,11,12,13 when using ethernet shield//

int ch;

PC.begin (9600);

for(ch=0; ch<3;ch++) {

    pinMode(INTpin[ch],INPUT_PULLUP);

    attachInterrupt(INTch[ch], isrFunc[ch], RISING); // set External Interupt

}

for(ch=0; ch<3; ch++) initSR04(trigPin[ch],echoPin[ch]); // inisialisasi SR04

for(ch=0; ch<3;ch++) {
```

```
pinMode(selPin[ch],OUTPUT);           // inisialisasi selenoid

pinMode(pumpPin[ch],OUTPUT);         // inisialisasi pompa
}

pinMode(INTpin[ch],INPUT_PULLUP);

pinMode(buzzer,OUTPUT);              // inisialisasi buzzer

pinMode(TBL_sys,INPUT_PULLUP);       // inisialisasi tombol

pinMode(TBL_bocor,INPUT_PULLUP);

pinMode(ctrl_com,OUTPUT);            // inisialisasi COMON

pinMode(LED_sys,OUTPUT);             // inisialisasi LED

pinMode(LED_bocor,OUTPUT);

digitalWrite(ctrl_com,LOW);

t = millis();

t_sample = t;
```

```
}
```

```
void loop() {
```

```
    int ch;
```

```
    if(!sistemON) {if(digitalRead(TBL_sys)==LOW) sistemON = true;}
```

```
    if(bocor)    {if(digitalRead(TBL_bocor)==LOW) bocor = false;}
```

```
    digitalWrite(LED_sys, sistemON);
```

```
    digitalWrite(LED_bocor, bocor);
```

```
    t = millis();
```

```
    if((t-t_sample)>samplingPeriode) {
```

```
        t_sample = t;
```

```
        for(ch=0; ch<3; ch++) {
```

```
            c_copy_L[ch] = c_copy_N[ch];
```

```
            t_copy_L[ch] = t_copy_N[ch];
```

```

}

for(ch=0; ch<3; ch++) {

    c_copy_N[ch] = c_[ch];

    t_copy_N[ch] = t_[ch];

}

for(ch=0; ch<3; ch++) {

    c_flow[ch] = c_copy_N[ch] - c_copy_L[ch];

    t_flow[ch] = t_copy_N[ch] - t_copy_L[ch];

    if(t_flow[ch]!=0) {

        // flow(mL/mnt) = freq * resolusi * 1 * 1

        // pulse * 1(L) * 60000(ms) * 1000(mL)

        // -----

        // periode(ms) * 5880 * 1(mnt) * 1(L)

        uint32_t Num = c_flow[ch] * 500000L;

        uint32_t Den = t_flow[ch] * 49L;

        rate[ch] = Num / Den;

    }

```

```
    else        rate[ch] = 0;

}

errorFlow = rate[FiltTank] - rate[cityTank_1] - rate[cityTank_2];

filtError = LPF((float)errorFlow,filtError,0.01);

if(filtError<maxError) {

    t_bocor = t;

}

else if((t-t_bocor)>=tout_bocor) {

    bocor = true;

    t_bocor = t;

}

}

for(ch=0; ch<3; ch++) {

    jarak[ch] = readSR04(trigPin[ch],echoPin[ch]);    // baca jarak pada gate
yang sedang dipantau

    fJarak[ch] = LPF((float)jarak[ch],fJarak[ch],0.1);
```

```
if(fJarak[ch]<jarakPenuh[ch])    penuh[ch] = 1;

else if(fJarak[ch]>jarakKosong[ch]) penuh[ch] = 0;

}

kekeruhan = LPF((float)analogRead(turbSens),kekeruhan,0.1);

if(kekeruhan<batasKeruh)    keruh = true;

else if(kekeruhan>batasJernih) keruh = false;

if(sistemON) {

    if(penuh[FiltTank]==1) {

        digitalWrite(pumpPin[StorageTank],LOW);

        Mb.C[4]=LOW;

        if(keruh) {

            digitalWrite(pumpPin[FiltTank],LOW);

            Mb.C[5]=LOW;

            digitalWrite(selPin[bayPass],LOW);
```

```
Mb.C[7]=LOW;

digitalWrite(selPin[cityTank_1],LOW);

Mb.C[8]=LOW;

digitalWrite(selPin[cityTank_2],LOW);

Mb.C[9]=LOW;

}

else {

    digitalWrite(pumpPin[FiltTank],HIGH);

    Mb.C[5]=HIGH;

    if(bocor) {

        digitalWrite(selPin[cityTank_1],LOW);

        Mb.C[8]=LOW;

        digitalWrite(selPin[cityTank_2],LOW);

        Mb.C[9]=LOW;

        if((penuh[cityTank_1]==1)|(penuh[cityTank_2]==1)) {

            digitalWrite(selPin[bayPass],LOW);

            Mb.C[7]=LOW;
```

```
sistemON = false;

}

else digitalWrite(selPin[bayPass],HIGH);

Mb.C[7]=HIGH;

}

else {

digitalWrite(selPin[bayPass],LOW);

Mb.C[5]=LOW;

for(ch=cityTank_1; ch<=cityTank_2; ch++) {

    if(penuh[ch]==1) digitalWrite(selPin[ch],LOW);

    else digitalWrite(selPin[ch],HIGH);

}

if((penuh[cityTank_1]==1)&(penuh[cityTank_2]==1)) sistemON = false;

}

}

}
```



```
else {

    digitalWrite(pumpPin[StorageTank],HIGH);

    Mb.C[4]=HIGH;

    digitalWrite(pumpPin[FiltTank],LOW);

    Mb.C[5]=LOW;

    digitalWrite(selPin[bayPass],LOW);

    Mb.C[7]=LOW;

    digitalWrite(selPin[cityTank_1],LOW);

    Mb.C[8]=LOW;

    digitalWrite(selPin[cityTank_2],LOW);

    Mb.C[9]=LOW;

}

digitalWrite(pumpPin[ReFiltTank],(keruh) ? HIGH : LOW);

Mb.C[6]=HIGH;

if(bocor) {

    if(buzcount==0)    digitalWrite(buzzer,HIGH);
```

```
else if(buzcount==5) digitalWrite(buzzer,LOW);

buzcount++;

if(buzcount>=10) buzcount = 0;

}

else digitalWrite(buzzer,LOW);

}

else {

digitalWrite(pumpPin[StorageTank],LOW);

Mb.C[4]=LOW;

digitalWrite(pumpPin[FiltTank],LOW);

Mb.C[5]=LOW;

digitalWrite(pumpPin[ReFiltTank],LOW);

Mb.C[6]=LOW;

digitalWrite(selPin[bayPass],LOW);

Mb.C[7]=LOW;
```

```
digitalWrite(selPin[cityTank_1],LOW);

Mb.C[8]=LOW;

digitalWrite(selPin[cityTank_2],LOW);

Mb.C[9]=LOW;

digitalWrite(buzzer,LOW);

keruh = false;

bocor = false;

buzcount = 0;

}

// serial monitor

PC.print(filError,0);  PC.write('\t');           // eror debit

for(ch=0; ch<3; ch++) {

    PC.print(fJarak[ch],0);  PC.write('\t');     // jarak sensor - air

}

PC.print(kekeruhan,0);  PC.write('\t');        // kekeruhan

PC.print(sistemON,DEC);  PC.write('\t');       // sistem ON
```

```
PC.print(bocor,DEC);  PC.write('\t');           // kebocoran

PC.write('[');

for(ch=0; ch<3; ch++) {

    PC.print(digitalRead(pumpPin[ch]),DEC);  PC.write(',');  // status pompa

}

PC.write(']');

PC.write('{');

for(ch=0; ch<3; ch++) {

    PC.print(digitalRead(selPin[ch]),DEC);  PC.write(',');  // status selenoid

}

PC.write('}');

PC.write('\n');

//----- kirim mudbus

Mb.R[0]= (int)kekeruhan;

Mb.R[1] = fJarak[FiltTank];           // jarak FiltTank
```

```
Mb.R[2] = fJarak[cityTank_1];           // jarak cityTank_1

Mb.R[3] = fJarak[cityTank_2];         // jarak cityTank_2

Mb.R[4] = filtError;                  // error bocor

Mb.C[2] = bocor;                       // status bocor

Mb.C[3] = sistemON;                   // status sistem

Mb.C[4] = pumpPin[StorageTank];       // status pompa

Mb.C[5] = pumpPin[FiltTank];

Mb.C[6] = pumpPin[ReFiltTank];

Mb.C[7] = selPin[bayPass];            // status selenoid

Mb.C[8] = selPin[cityTank_1];

Mb.C[9] = selPin[cityTank_2];

}

Mb.Run();

if(!sistemON) {if(Mb.C[0]) sistemON = true;}
```

```
if(bocor) {if(Mb.C[1]) bocor = false;}
```

```
}
```