

BUKU AJAR PERANCANGAN BERBASIS KOMPUTER DENGAN INTEGRASI ARDUINO DAN ANDROID

Penulis :

Eko Agus Suprayitno, S.Si, MT

Rohman Dijaya, S.Kom, M.Kom



BUKU AJAR
PERANCANGAN BERBASIS KOMPUTER DENGAN
INTEGRASI ARDUINO DAN ANDROID

Penulis

Eko Agus Suprayitno, S.Si, MT
Rohman Dijaya, S.Kom, M.Kom



Diterbitkan oleh
UMSIDA PRESS
Jl. Mojopahit 666 B Sidoarjo
ISBN: 978-602-5914-05-8
Copyright©2018.
Authors
All rights reserved

**BUKU AJAR
PERANCANGAN BERBASIS KOMPUTER DENGAN
INTEGRASI ARDUINO DAN ANDROID**

Penulis :

Eko Agus Suprayitno, S.Si, MT
Rohman Dijaya, S.Kom, M.Kom

ISBN :

978-602-5914-05-8

Editor :

Septi Budi Sartika, M.Pd
M. Tanzil Multazam , S.H., M.Kn.

Copy Editor :

Fika Megawati, S.Pd., M.Pd.

Design Sampul dan Tata Letak :

Mochamad Nashrullah, S.Pd

Penerbit :

UMSIDA Press

Redaksi :

Universitas Muhammadiyah Sidoarjo
Jl. Mojopahit No 666B
Sidoarjo, Jawa Timur

Cetakan pertama,

© Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dengan suatu
apapun tanpa ijin tertulis dari penerbit.

IDENTITAS BUKU

1. Judul Buku : PERANCANGAN BERBASIS KOMPUTER
DENGAN INTEGRASI ARDUINO DAN ANDROID
2. Penulis : Eko Agus Suprayitno, S,Si, MT
& Rohman Dijaya. Skom., M.Kom
3. Penerbit : UMSIDA PRESS
4. Cetakan : Pertama, Sidoarjo
5. Jumlah Halaman : 103 Halaman
6. Tahun : 2018

Belakangan ini mikrokontroler keluarga Arduino Uno menjadi sangat populer. Tidak mengherankan jika turunan mikrokontroler tersebut pun bermunculan. Kepopulerannya tidak luput dari dukungan berbagai vendor yang menyediakan berbagai komponen yang dapat mendukung Arduino Uno dalam pengaplikasiannya. Disamping itu perkembangan teknologi jika tidak terintegrasi dengan smartphone juga akan kurang lengkap. Android sudah menempati posisi utama di setiap smartphone yang ada di Indonesia. Belajar mengintegrasikan Smartphone android dengan Arduino salah

sau peluang besar dalam meningkatkan daya saing teknologi di rana elektronika. Modul ini mencoba untuk membantu siapa saja yang mempelajari Panduan Praktik Mempelajari Mikrokontroler Menggunakan Arduino Uno dan integrasinya dengan Aplikasi di Smartphone Android. Arduino uno sendiri merupakan salah satu mata kuliah dalam bidang Teknik Elektro, Ilmu Komputer, dan Teknik Informatika yang sangat penting untuk dikuasai oleh mahasiswa sejak tingkat menengah. Modul ini disusun untuk membantu mahasiswa dalam memahami konsep elektronika dan sensor dengan cepat dan mudah, serta memudahkan mahasiswa mengontrol Hardware dengan bantuan smartphone. Pada buku ini, penulis menyusun modul dengan jumlah 10 bab, yakni Mengendalikan LCD sebagai Display pada Arduino, Rangkaian Motor DC, Rangkaian Sensor Ultrasonik, Rangkaian I2C LCD 16x2, Rangkaian Sensor LM 35, Rangkaian Relay, Sensor MAX30100, Sensor Suhu DS18B20, Komunikasi Arduino dengan Android memanfaatkan Bluetooth, Filter pada Instrumentasi Biomedik.

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, yang telah memberikan berkat dan karuniannya sehingga penulis dapat menyelesaikan PERANCANGAN BERBASIS KOMPUTER DENGAN INTEGRASI ARDUINO DAN ANDROID ini dengan baik. Perkembangan ilmu pengetahuan dan teknologi saat ini sangatlah cepat. Terutama Kemajuan teknologi dibidang Teknologi Informasi dan Komunikasi Otomasi. Perkembangan dan Kemajuan Teknologi tersebut tidak lepas dari semangat untuk memudahkan kehidupan dan komunikasi manusia. Sehingga integrasi keilmuan secara ilmiah antara pengembang teknologi lintas bidang keahlian sangat diperlukan.

Belakangan ini mikrokontroler keluarga Arduino Uno menjadi sangat populer. Tidak mengherankan jika turunan mikrokontroler tersebut pun bermunculan. Kepopulerannya tidak luput dari dukungan berbagai vendor yang menyediakan berbagai komponen yang dapat mendukung Arduino Uno dalam pengaplikasiannya. Disamping itu perkembangan teknologi jika tidak terintegrasi dengan smartphone juga akan kurang lengkap. Android sudah menempati posisi utama di setiap smartphone yang ada di Indonesia. Belajar mengintegrasikan Smartphone android dengan Arduino salah satu peluang besar dalam meningkatkan daya saing teknologi di rana elektronika. Modul ini mencoba untuk membantu siapa saja yang mempelajari Panduan Praktik Mempelajari Mikrokontroler Menggunakan Arduino Uno dan integrasinya dengan Aplikasi di Smartphone Android. Arduino uno sendiri

merupakan salah satu mata kuliah dalam bidang Teknik Elektro, Ilmu Komputer, dan Teknik Informatika yang sangat penting untuk dikuasai oleh mahasiswa sejak tingkat menengah. Modul ini disusun untuk membantu mahasiswa dalam memahami konsep elektronika dan sensor dengan cepat dan mudah, serta memudahkan mahasiswa mengontrol Hardware dengan bantuan smartpone. Pada buku ini, penulis menyusun modul dengan jumlah 10 bab, yakni mengendalikan LCD 16X2, Mengendalikan Motor DC, Membaca sensor Ultrasonik, Membaca sensor LM35, mengedalikan Relay, Menggabungkan I2C dengan LCD16x2, Membuat Aplikasi Android dengan MIT App Inventor, Integrasi Bluetooth HC 05 Arduino dengan Aplikasi Android, Menampilkan Nilai sensor Suhu dan jarak Arduino pada Aplikasi Android, Modul Wifi ESP8266-01 dan Nodemcu.

Dengan dibuatnya Buku Instrumentasi PERANCANGAN BERBASIS KOMPUTER DENGAN INTEGRASI ARDUINO DAN ANDROID ini, penulis berharap banyak masyarakat, Mahasiswa, enginer, maupun pendidik mampu memahami, mampu membuat dan mengembangkannya menjadi Instrumentasi yang tepat guna dan akurat.

Sidoarjo, 20 Januari 2018

Penulis

DAFTAR ISI

1. Mengendalikan LCD sebagai Display pada Arduino
 - a. Tujuan
 - b. Teori LCD 16X2 dan Pemrograman Arduino
 - c. Tugas Keterampilan Pemahaman & Pemrograman
2. Rangkaian Motor DC
 - a. Tujuan
 - b. Teori Motor DC dan Pemrograman Arduino
 - c. Tugas Keterampilan Pemahaman & Pemrograman
3. Rangkaian sensor Ultrasonik
 - a. Tujuan
 - b. Teori Sensor Ultrasonik dan Pemrograman Arduino
 - c. Tugas Keterampilan Pemahaman & Pemrograman
4. Rangkaian I2C LCD 16x2
 - a. Tujuan
 - b. Teori I2C LCD 16x2 dan Pemrograman Arduino
 - c. Tugas Keterampilan Pemahaman & Pemrograman
5. Rangkaian Sensor LM 35
 - a. Tujuan
 - b. Teori Sensor Sensor LM 35 dan Pemrograman Arduino
 - c. Tugas Keterampilan Pemahaman & Pemrograman
6. Rangkaian Relay
 - a. Tujuan
 - b. Teori Relay dan Pemrograman Arduino
 - c. Tugas Keterampilan Pemahaman & Pemrograman
7. Sensor MAX30100
 - a. Tujuan

- b. Teori Sensor MAX30100 dan Pemrograman Arduino
 - c. Tugas Keterampilan Pemahaman & Pemrograman
- 8. Sensor Suhu DS18B20
 - a. Tujuan
 - b. Teori Sensor Suhu DS18B20 dan Pemrograman Arduino
 - c. Tugas Keterampilan Pemahaman & Pemrograman
- 9. Komunikasi Arduino dengan Android memanfaatkan Bluetooth
 - a. Tujuan
 - b. Teori *Bluetooth*, Software Android MIT Inventor dan Pemrograman Arduino Software Android MIT Inventor
 - c. Tugas Keterampilan Pemahaman & Pemrograman
- 10. Filter pada Instrumentasi Biomedik
 - a. Instrumentasi Low Pass Filter
 - b. Instrumentasi High Pass Filter
 - c. Instrumentasi Notch Filter

BAB

11. BAB I : Mengendalikan LCD sebagai Display pada Arduino
12. BAB II : Rangkaian Motor DC
13. Bab III : Rangkaian Sensor Ultrasonik
14. Bab IV : Rangkaian I2C LCD 16x2
15. Bab V : Rangkaian Sensor LM 35
16. Bab VI : Rangkaian Relay
17. BAB VII : Sensor MAX30100
18. BAB VIII: Sensor Suhu DS18B20
19. BAB IX :Komunikasi Arduino dengan Android memanfaatkan Bluetooth
20. BAB X : Filter pada Instrumentasi Biomedik

CAPAIAN PEMBELAJARAN

1. Mahasiswa dapat Mengetahui dan memahami prinsip kerja Sensor Tepat Guna di bidang Elektronika dan Otomasi secara Umum
2. Mahasiswa Mampu melakukan pemrogram otomasi sensor dan integrasinya dengan sensor-sennsor lainnya untuk aplikasinya pada teknologi Tepat Guna
3. Mahasiswa dapat mampu membuat Aplikasi Android pada Smartphone
4. Mahasiswa Mampu memahami komunikasi Nirkabel antara Arduino dengan Aplikasi Android di Smartphone
5. Mahasiswa mampu membuat Teknologi Tepat Guna Otomatis yang bermanfaat bagi masyarakat dengan memanfaatkan sensor yang ada
6. Mahasiswa mampu mengendalikan Hardware Teknologi Tepat Guna menggunakan Aplikasi Android di Smartphone.
7. Mahasiswa mampu Filter-filter Instrumentasi Medis yang terintegrasi dengan mikrokontroller

BAB I

Mengendalikan LCD sebagai Display pada Arduino

1. Instrumentasi LCD (*Liquid Crystal Display*) 16x2

A. TUJUAN

Mengenal Komponen LCD (*Liquid Crystal Display*), Membuat Tulisan di LCD, dan Memprogram Karakter Tulisan di LCD 16x2.

B. TEORI

LCD (*Liquid Crystal Display*) merupakan suatu display tampilan karakter yang dibuat dari bahan cairan Kristal yang pengoperasiannya menggunakan system dot matriks. LCD (*Liquid Crystal Display*) 16x2 dapat menampilkan sebanyak 32 karakter yang terdiri dari 2 baris dan tiap baris dapat menampilkan 16 karakter. (Andrianto H, Darmawan A. 2016).

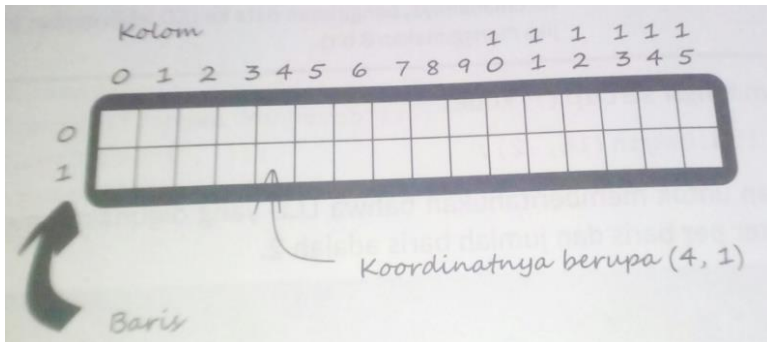


Gambar 1 LCD16x2 (Andrianto H, Darmawan A. 2016).

Tabel 1: Fungsi Pin LCD *Character* 16x2 (Wardhana, Lingga 2006).

PIN	Name	Function
1	V _{SS}	Ground voltage
2	V _{CC}	+5V
3	V _{EE}	Contrast voltage
4	RS	Register Select 0 = Instruction Register 1 = Data Register
5	R/W	Read/ Write, to choose write or read mode 0 = write mode 1 = read mode
6	E	Enable 0 = start to lacht data to LCD character 1= disable
7	DB0	LSB
8	DB1	-
9	DB2	-
10	DB3	-
11	DB4	-
12	DB5	-
13	DB6	-
14	DB7	MSB
15	BPL	Back Plane Light
16	GND	Ground voltage

Pada LCD yang berukuran 16x2 (16 kolom, 2 baris) terdapat 32 koordinat yang berarti dapat ditempati 32 karakter. Maksud dari ukuran 16x2 (16 kolom, 2 baris) yakni bertujuan untuk memudahkan dalam meletakkan kursor di kolom 0-15 (kolom pertama) pada baris ke 0 maupun baris ke 1. Untuk baris utama pada LCD disebut baris ke 0 yang biasanya ada di baris atas sendiri, dan baris ke 1 biasanya posisinya setelah baris ke 0. Untuk memudahkan ilustrasinya maka lihat gambar system koordinat yang berlaku di LCD terkait penempatan kolom dan baris sebagai berikut :

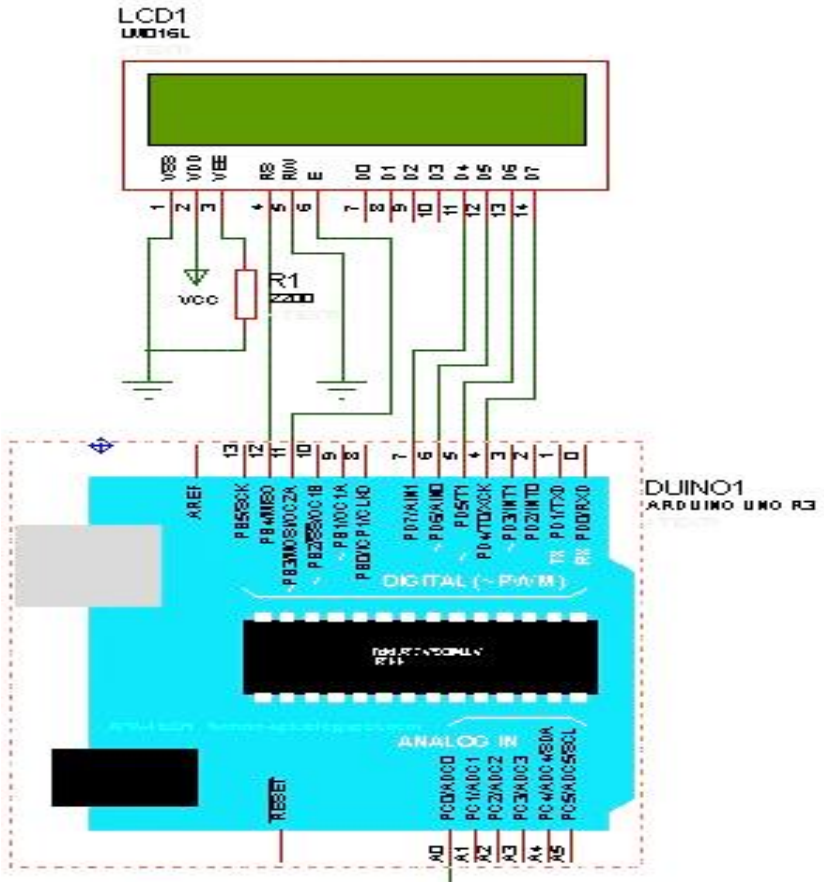


Gambar 2: system koordinat di LCD (Kadir A, 2013)

Koordinat pojok kiri-atas dinyatakan $(0,0)$, sedangkan koordinat pojok kanan-atas berupa $(0,15)$, Jadi secara umum, koordinat dinyatakan dengan (**kolom, Baris**).

- Sebagai contoh Koordinat dari kolom 0-15, baris 0 yang mengikuti acuan (**kolom, Baris**) seperti berikut ini :
 Koordinat $(0,0)$, koordinat $(1,0)$, koordinat $(2,0)$,
 korrdinat $(3,0)$, koordinat $(4,0)$, koordinat $(5,0)$,
 koordinat $(6,0)$, koordinat $(7,0)$, koordinat $(8,0)$,
 koordinat $(9,0)$, koordinat $(10,0)$, koordinat $(11,0)$,
 koordinat $(12,0)$, koordinat $(13,0)$, koordinat $(14,0)$,
 koordinat $(15,0)$.
- Sebagai contoh koordinat kolom 0-15, baris 1 yang mengikuti acuan (**kolom, Baris**) seperti berikut ini :
 Koordinat $(0,1)$, koordinat $(1,1)$, koordinat $(2,1)$,
 korrdinat $(3,1)$, koordinat $(4,1)$, koordinat $(5,1)$,
 koordinat $(6,1)$, koordinat $(7,1)$, koordinat $(8,1)$,
 koordinat $(9,1)$, koordinat $(10,1)$, koordinat $(11,1)$,
 koordinat $(12,1)$, koordinat $(13,1)$, koordinat $(14,1)$,
 koordinat $(15,1)$. (Kadir A, 2013).

Material yang digunakan dalam Latihan pengoprasian LCD (*Liquid Crystal Display*) 16x2 dengan Arduino dapat dilakukan dengan cara berikut.






Gambar 3. Skematik Rangkain LCD (*Liquid Crystal Display*) 16x2

❖ **Persiapan secara Mekanik pada Hardware**

1. Perhatikan Gambar 3 Skematik Rangkain *LCD* diatas sebagai ilustrasi dalam uji cobanya. Berikutnya ikuti langkah seperti panduan point nomor 2 hingga point nomor 11. Untuk memudahkan dalam praktek maka siapkan Modul Hardware Arduino UNO, Kabel USB Arduino ke PC, Kabel Jumper, LCD, dan Install Software Arduino IDE pada PC anda.
2. Masukkan kabel jumper ke pin digital Arduino no.4 kemudian sambungkan kabel jumper tersebut ke kaki LCD No.14/D7.
3. Masukkan kabel jumper ke pin digital Arduino no.5 kemudian sambungkan kabel jumper tersebut ke kaki LCD No.13/D6.
4. Masukkan kabel jumper ke pin digital Arduino no.6 kemudian sambungkan kabel jumper tersebut ke kaki LCD No.12/D5.
5. Masukkan kabel jumper ke pin digital Arduino no.7 kemudian sambungkan kabel jumper tersebut ke kaki LCD No.11/D4.
6. Masukkan kabel jumper ke pin digital Arduino no.11 kemudian sambungkan kabel jumper tersebut ke kaki LCD No.6/E.
7. Masukkan kabel jumper ke LCD No.5/ R/W lalu untuk output LCD No.5/ R/W pasanglah ke Ground.
8. Masukkan kabel jumper ke pin digital Arduino no.12 kemudian sambungkan kabel jumper tersebut ke kaki LCD No.4/RS.
9. Masukkan kabel jumper ke LCD No.3/VEE lalu untuk output dari LCD No.3/VEE pasanglah ke Input Resistor dengan

- ukuran Hambatannya 2200 ohm. Setelah kabel terpasang ke Resistor untuk outputnya pasang menuju ke Ground.
10. Masukkan kabel jumper ke LCD No.2/VCC lalu output dari LCD No.2/VCC pasang menuju VCC.
 11. Masukkan kabel jumper ke LCD No.1/VSS lalu output dari LCD No.1/VSS pasang ke Ground.

❖ **Persiapan secara Software *Arduino IDE*:**

1. Nyalakan komputer dan Buka Program Arduino IDE 
2. Ketik Algoritma sebagaimana contoh pada Gambar 4 atau Algoritma Menampilkan karakter di LCD 16x2 .
3. Setelah memasukkan algoritma pada Arduino IDE maka Verifikasi Algoritma yang sudah anda buat agar bisa dideteksi kesalahan dalam penulisan programnya 
4. Compile Algoritma pada modul Arduino agar bisa dilihat hasilnya pada LCD .
5. Lihat hasilnya pada tampilan LCD.
6. Untuk memudahkan pemrograman lihat Algoritma pada Gambar 4.

```
Syukuri_apa_yang_ada $  
  
#include <LiquidCrystal.h>  
const int PIN_RS = 12;  
const int PIN_E = 11;  
const int PIN_DB4 = 7;  
const int PIN_DB5 = 6;  
const int PIN_DB6 = 5;  
const int PIN_DB7 = 4;  
  
// Buat Objek  
  
LiquidCrystal lcd(PIN_RS, PIN_E, PIN_DB4, PIN_DB5, PIN_DB6, PIN_DB7);  
void setup() {  
  lcd.begin(16,2);  
}  
void loop() {  
  lcd.clear();  
  lcd.print("Syukuri Apa");  
  lcd.setCursor(0,1);  
  lcd.print("Yang Ada");  
  delay(5000);  
  
  lcd.clear();  
  lcd.print("Hidup Adalah");  
  lcd.setCursor(0,1);  
  lcd.print("Anugrah");  
  delay(5000);  
  
  lcd.clear();  
  lcd.setCursor(4,1);  
  lcd.print("Tetap jalani");  
  delay(5000);  
}
```

14 Arduino/Genuino Uno on COM1

Gambar 4. Algoritma pada IDE Arduino

Algoritma yang harus di tuliskan :

```

#include <Wire.h>
#include <LiquidCrystal.h>
const int PIN_RS = 12;
const int PIN_E = 11;
const int PIN_DB4 = 7;
const int PIN_DB5 = 6;
const int PIN_DB6 = 5;
const int PIN_DB7 = 4;
// Buat Objek
LiquidCrystal lcd(PIN_RS, PIN_E, PIN_DB4, PIN_DB5,
PIN_DB6, PIN_DB7);
void setup() {
  lcd.begin(16,2);
}
void loop() {
  lcd.clear();
  lcd.print("Syukuri Apa");
  lcd.setCursor(0,1);
  lcd.print("Yang Ada");
  delay(5000);
  lcd.clear();
  lcd.print("Hidup Adalah");
  lcd.setCursor(0,1);
  lcd.print("Anugrah");
  delay(5000);
  lcd.clear();
  lcd.setCursor(4,1);
  lcd.print("Tetap jalani");
  delay(5000); }

```

7. Tulis karakter berdasarkan koordinat (kolom, Baris) untuk keterangan seperti sebagaimana tabel berikut ini, dan Foto Hasilnya untuk dimasukkan pada Laporan Praktikum.

No	Koordinat (Kolom, Baris)	Karakter	Hasil Foto
1	(3,0)	Nama	
2	(2,1)	NIM	
3	(0,1)	Jurusan	
4	(4,0)	Institusi	
5	(5,1)	Hobi	

C. TUGAS

1. Jelaskan macam-macam tipe ukuran LCD selain LCD 16x2 dan tentukan macam-macam koordinatnya.
2. Jelaskan Fungsi ***void setup()*** dan ***void loop()*** pada Arduino IDE.

BAB 2

RANGKAIAN MOTOR DC

1. INSTRUMENTASI MOTOR DC

A. TUJUAN

Mengenal Motor DC, Merakit agar Motor DC bisa berputar, dan Memprogram Arduino IDE untuk menggerakkan Motor DC.

B. TEORI

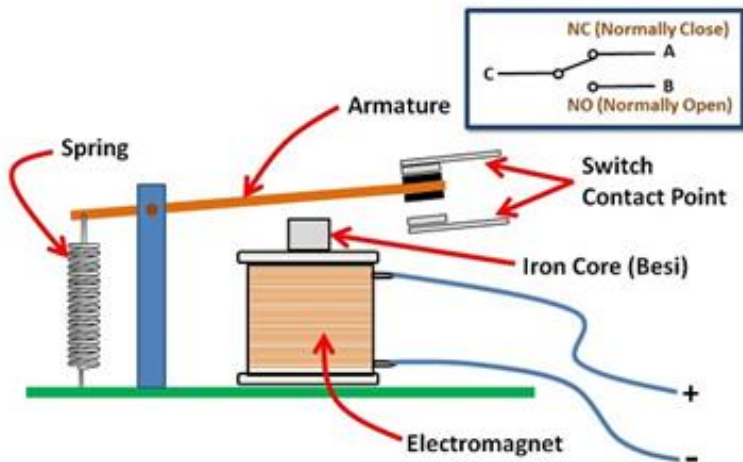
1. Motor DC

Motor DC ialah motor yang menggunakan sumber tegangan DC dan digunakan untuk mengubah tenaga listrik menjadi tenaga mekanis. Komponen ini bekerja dengan prinsip electromagnet. Apabila Motor DC diberi tegangan, maka medan magnet yang diam atau disebut stator akan terbentuk. Medan magnet ini membuat rotor atau bagian yang bergerak atau berputar akan dapat dimanfaatkan untuk memutar benda yang sulit untuk manusia kerjakan. Kecepatan Motor DC sendiri tergantung dengan besarnya tegangan. Semakin besar tegangan akan semakin cepat Motor DC berputar dan begitu juga sebaliknya apabila Motor DC mendapatkan tegangan kecil maka kecepatan juga akan lamban. Namun tegangan yang terdapat di Motor DC memiliki batasan dalam bekerja atau berputar. Apabila kemampuan Motor DC di hiraukan maka Motor akan terbakar. Contoh Gambar 1. adalah motor spesifikasi tegangan 1,5V-5V (Kadir A, 2013). Motor DC memiliki kutub positif dan kutub negatif, kutub-kutub tersebut mempengaruhi jenis arah putaran Motor DC.



Gambar 1. Bentuk Motor DC (Kadir A 2013)

2. Relay



Gambar 2. Struktur sederhana Relay

Relay merupakan suatu Saklar (*Switch*) elektromagnetik yang apabila kumparan pada relay di aliri arus listrik dapat membuat Inti Besi (*Iron Core*) menjadi magnet sehingga dapat menarik kontak (*Armatur*) jika gaya

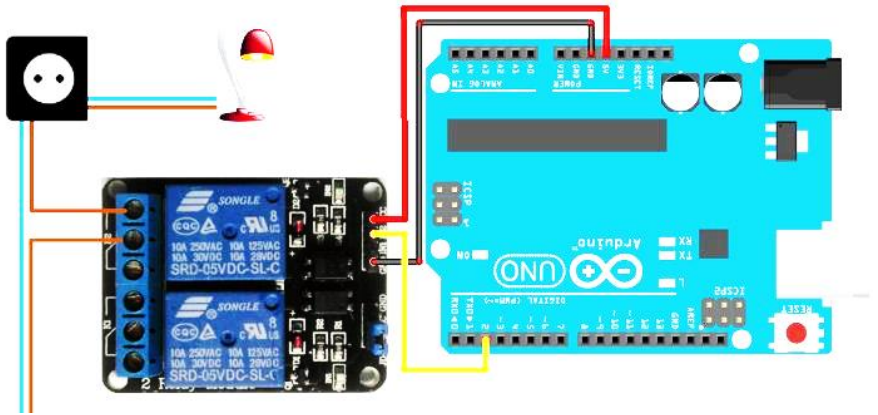
magnet yang dihasilkan menjadi lebih besar dari pada gaya pegas (*spring*) yang ada sebagaimana ilustrasi pada gambar (aziz, 2009). Relay yang menggunakan Elektromagnet 5V dan 50 mA mampu menggerakkan *Armature Relay* (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A.

Berdasarkan pada Gambar diatas Kontak Poin Relay terdiri dari 2 jenis yaitu :

- *Normally Close* (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi CLOSE (tertutup)
- *Normally Open* (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi OPEN (terbuka)

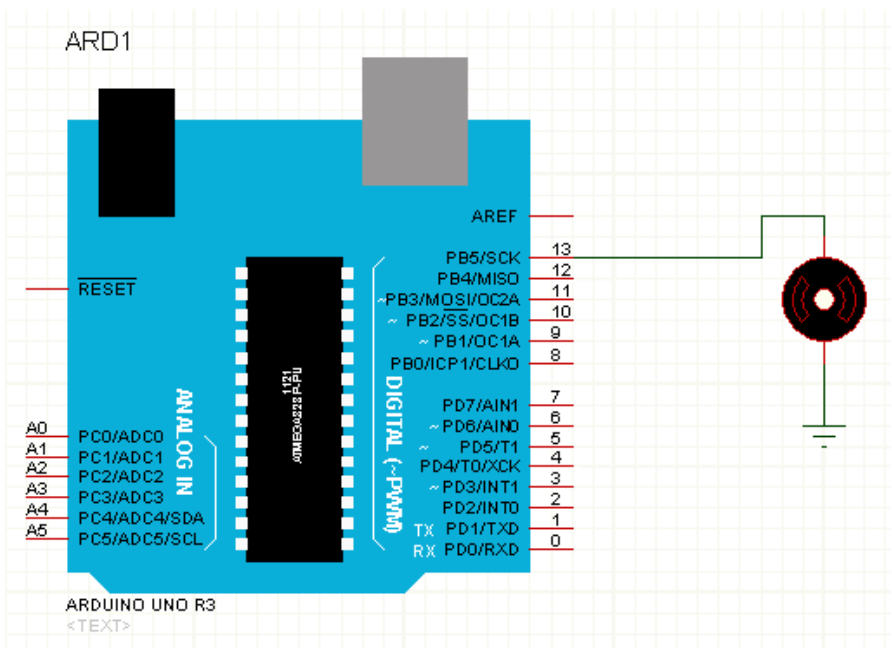
Pada gambar diatas apabila Kumparan Coil dilewati arus listrik, maka akan timbul gaya Elektromagnet yang akan menarik *Armature* untuk berpindah dari Posisi sebelumnya (NC) ke posisi baru (NO), sehingga Saklar dapat menghantarkan arus listrik di posisi (NO). Sehingga Posisi awal *Armature* sebelumnya (NC) menjadi tidak terhubung. Begitu sebaliknya ketika Kumparan Coil tidak dialiri arus listrik maka *Armature* akan kembali lagi ke posisi *Normally Close* (NC). Arus listrik yang dibutuhkan dalam memindahkan kontak dari NC ke NO relatif kecil. Perlu diketahui bahwa Relay ada yang aktif Low dan aktif High. Maksudnya aktif Low adalah relay akan aktif jika diberi masukan LOW atau nol, sedangkan aktif High adalah relay akan aktif jika diberi masukan HIGH atau satu.

Berikut ini Contoh gambar pemasangan Relay 2 Chanel pada Arduino UNO untuk menyalakan dan mematikan Lampu.



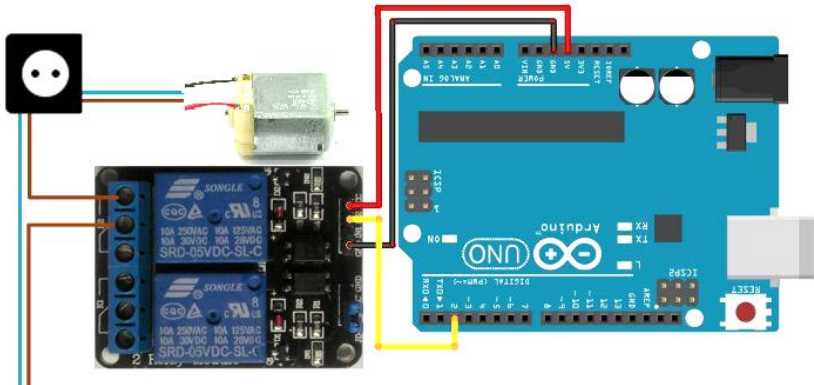
Gambar 3. Pemasangan Relay 2 Chanel pada Arduino UNO

Material yang digunakan dalam latihan penggunaan motor DC dapat dilakukan dengan cara berikut.



Gambar 4. Rangkaian Motor DC



1. Siapkan Arduino Uno sebelum merakit Motor DC.
2. Gunakan Motor DC sesuai dengan tegangan yang sudah ditentukan.
3. Motor DC pada praktikum ini akan digerakkan dengan bantuan Supplay Tegangan 12 Volt sehingga butuh relay dalam pengoperasiannya. Relay yang digunakan pada praktikum ini adalah **tipe relay Aktif LOW**. Sehingga Masukkan kabel jumper warna merah pada Normally Close (NC) dan Normlly Open (NO) supplay tegangan 12 Volt untuk menggerakkan Motor seperti gambar dibawah ini. Taruh kabel hitam Motor DC pada ground.



Gambar 5. pemasangan Relay 2 Chanel pada Arduino UNO

4. Masukkan kabel jumper pada Pin 12 Arduino (digital) dan hubungkan pin tersebut ke pin masukan (Input) Relay 4 Chanel. Masukkan Ground Relay 4 Chanel ke ground Arduino Uno. Serta Masukkan supplay (vcc) Relay 4 Chanel ke vcc Arduino Uno.
5. Program Arduino IDE seperti coding dibawah ini :

<pre> Motor_sederhana \$ const int PIN_12 =12; void setup() { pinMode(PIN_12, OUTPUT); } void loop() { digitalWrite(PIN_12, LOW); delay(1000); digitalWrite(PIN_12, HIGH); delay(1000); } </pre>	<pre> const int PIN_12 =12; void setup() { pinMode(PIN_12, OUTPUT); } void loop() { digitalWrite(PIN_12, LOW); delay(1000); digitalWrite(PIN_12, HIGH); delay(1000); } </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6. Setelah memasukkan algoritma pada Arduino IDE maka Verifikasi Algoritma yang sudah anda buat agar bisa dideteksi kesalahan dalam penulisan programnya 
7. Compile Algoritma pada modul Arduino agar bisa dilihat hasilnya pada Lputaran motor .
8. Isi tabel berikut ini untuk menggerakkan Motor jika ketentuannya sebagai berikut :

No	Posisi PIN Arduino	Delay	Kecepatan Putaran Motor (rpm)	Foto Hasil
1	Pin 13	2000		
2	Pin 9	1000		
3	Pin 6	1500		
4	Pin 2	2500		
5	Pin 4	1000		

C. TUGAS

1. Apakah Motor DC dapat Berputar dengan baik jika kabel jumper antara relay dengan arduino dimasukkan di Pin Analog Arduino (A0) ?
2. Apa alat ukur yang digunakan untuk mengukur putaran Motor ? dan apa satuannya ?

BAB III

RANGKAIAN SENSOR ULTRASONIK

1. INSTRUMENTASI SENSOR ULTRASONIK

A. TUJUAN

Mengoprasikan, memanfaatkan, dan memprogram Sensor HC-SR04 (Ultrasonik) untuk mengukur jarak atau kedalaman.

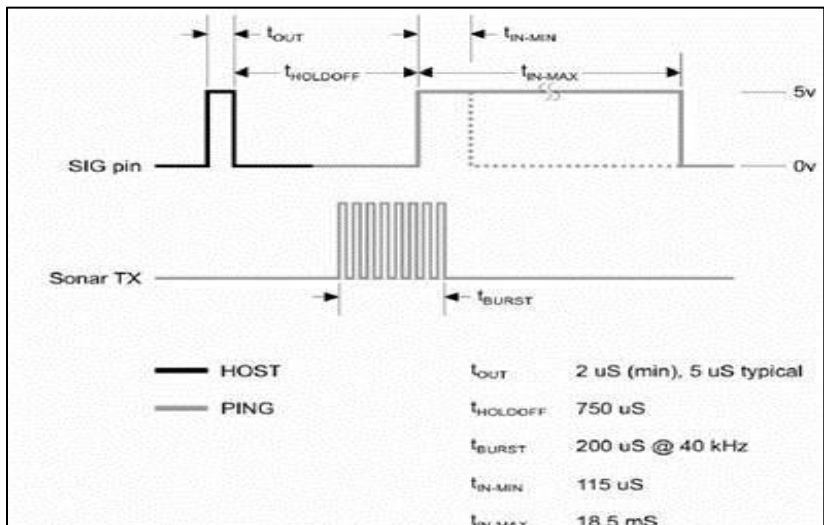
B. TEORI

Sensor ultrasonik bekerja dengan cara memancarkan suatu gelombang dan kemudian menghitung waktu pantulan gelombang tersebut. Gelombang ultrasonik bekerja pada frekuensi mulai dari 20 KHz sampai dengan 20 MHz. frekuensi kerja yang digunakan dalam gelombang ultrasonik bervariasi tergantung pada medium yang dilalui, mulai dari kerapatan pada fasa gas, cair, hingga padat. Sensor ultrasonik terdiri dari sebuah *chip* pembangkit sinyal 40 KHz, sebuah speaker ultrasonik, dan sebuah *microphone* ultrasonik. Speaker ultrasonik (*transmitter*) mengubah sinyal 40 KHz menjadi suara sementara *microphone* ultrasonik (*receiver*) berfungsi untuk mendeteksi pantulan suaranya. Sensor ultrasonik akan mengirimkan suara ultrasonik ketika ada pulsa *trigger* dari mikrokontroler. Suara ultrasonik dengan frekuensi sebesar 40 KHz akan dipancarkan selama 200 μ s. Suara ini akan merambat di udara dengan kecepatan 340 m/s atau 29.412 μ s setiap 1 cm, mengenai objek dan akan terpantul kembali ke sensor ultrasonik. Selama menunggu pantulan, sensor ultrasonik akan menghasilkan sebuah pulsa. Pulsa ini akan berlogika *low* ketika suara pantulan terdeteksi oleh sensor ultrasonik. Maka dari itu, lebar pulsa dapat

merepresentasikan jarak antara sensor ultrasonik dengan objek. Contoh Gambar sensor ultrasonic dapat dilihat seperti gambar berikut ini. (Andirianto H, Darmawan A 2016)

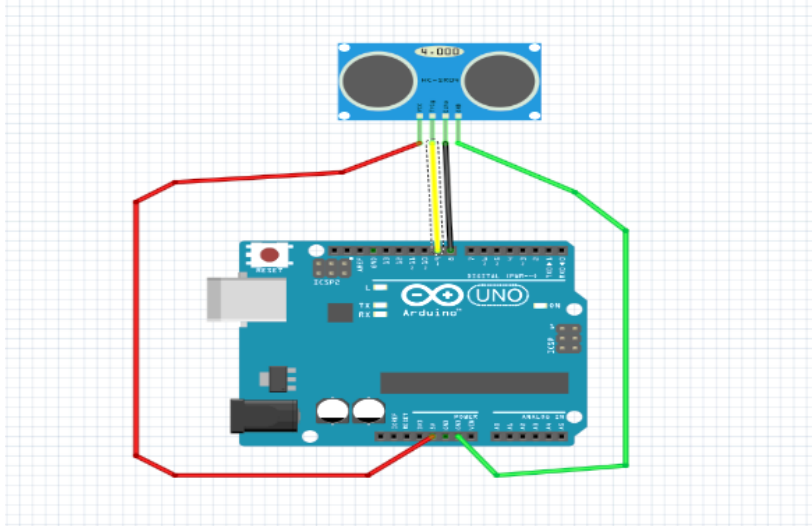


Gambar 1. Sensor Ultrasonik. (Andirianto H, Darmawan A 2016)



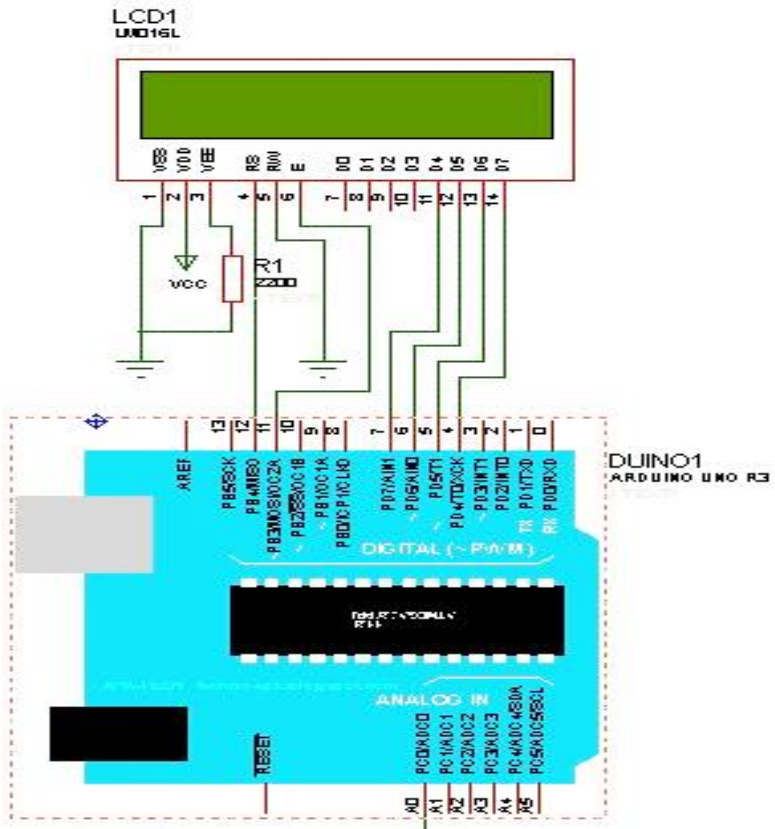
Gambar 2. Diagram waktu sensor ultrasonik (Andirianto H, Darmawan A 2016)

Sensor Ultrasonik memiliki 4 pin yang diantaranya pin Ground, pin vcc, pin Transmitter dan Receiver. Pin Transmitter dan Receiver akan terhubung ke pin Digital Arduino. Material yang digunakan dalam latihan penggunaan *sensor ultrasonik* dapat dilakukan dengan cara berikut.



Gambar 3. Rangkaian Sensor Ultrasonik

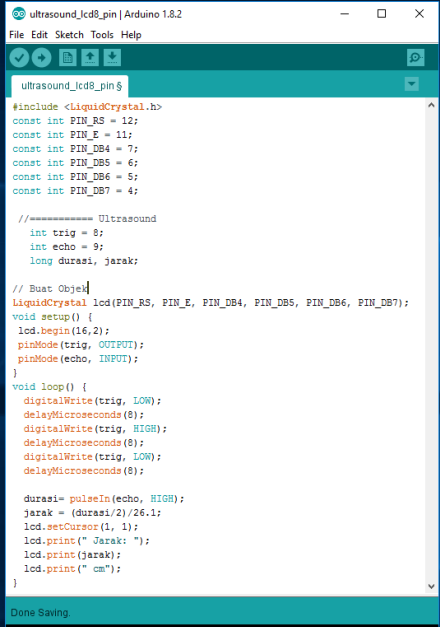
1. Siapkan dulu Arduino Uno sebelum kita merakit Sensor Ultrasonik sebagaimana gambar 3 diatas dengan rincian komponennya adalah sebagai berikut : Arduino UNO, Kabel jumper , sensor ultrasonic, dan Penggaris
2. Siapkan Sensor Ultrasonik.
3. Pasang LCD 16x2 Seperti Percobaan Modul 1 awal Praktikum ini, dengan memperhatikan pin-pin antara Arduino dengan LCD 16x2. LCD disini digunakan untuk menampilkan jarak yang terbaca oleh sensor Ultrasonik.





Gambar 4. Skematik Rangkain LCD (*Liquid Crystal Display*) 16x2

4. Masukkan kabel jumper ke vcc (Sensor Ultrasonik), kemudian hubungkan kabel jumper tersebut menuju Tegangan vcc Arduino sebagai sumber.
5. Masukkan kabel jumper ke ground (Sensor Ultrasonik), kemudian hubungkan kabel jumper tersebut menuju Tegangan ground Arduino sebagai sumber.

6. Masukkan kabel jumper ke Trigger pada sensor ultrasonik kemudian masukkan kembali kabel jumper menuju Pin Digital No.9 pada Arduino Uno.
7. Masukkan kabel jumper ke Echo pada sensor ultrasonik kemudian masukkan kembali kabel jumper menuju Pin Digital No.8 pada Arduino Uno.
8. Program Arduino IDE seperti gambar dan keterangan koding berikut ini :

 <pre> ultrasound_lcd8_pin Arduino 1.8.2 File Edit Sketch Tools Help ultrasound_lcd8_pin \$ #include <LiquidCrystal.h> const int PIN_RS = 12; const int PIN_E = 11; const int PIN_DB4 = 7; const int PIN_DB5 = 6; const int PIN_DB6 = 5; const int PIN_DB7 = 4; //===== Ultrasonid int trig = 8; int echo = 9; long durasi, jarak; // Buat Objek LiquidCrystal lcd(PIN_RS, PIN_E, PIN_DB4, PIN_DB5, PIN_DB6, PIN_DB7); void setup() { lcd.begin(16,2); pinMode(trig, OUTPUT); pinMode(echo, INPUT); } void loop() { digitalWrite(trig, LOW); delayMicroseconds(8); digitalWrite(trig, HIGH); delayMicroseconds(8); digitalWrite(trig, LOW); delayMicroseconds(8); durasi= pulseIn(echo, HIGH); jarak = (durasi/2)/26.1; lcd.setCursor(1, 1); lcd.print(" Jarak: "); lcd.print(jarak); lcd.print(" cm"); } Done Saving </pre>	<pre> #include <LiquidCrystal.h> const int PIN_RS = 12; const int PIN_E = 11; const int PIN_DB4 = 7; const int PIN_DB5 = 6; const int PIN_DB6 = 5; const int PIN_DB7 = 4; //===== Ultrasonid int trig = 8; int echo = 9; long durasi, jarak; // Buat Objek LiquidCrystal lcd(PIN_RS, PIN_E, PIN_DB4, PIN_DB5, PIN_DB6, PIN_DB7); void setup() { lcd.begin(16,2); pinMode(trig, OUTPUT); pinMode(echo, INPUT); } void loop() { </pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<pre> digitalWrite(trig, LOW); delayMicroseconds(8); digitalWrite(trig, HIGH); delayMicroseconds(8); digitalWrite(trig, LOW); delayMicroseconds(8); durasi= pulseIn(echo, HIGH); jarak = (durasi/2)/26.1; lcd.setCursor(1, 1); lcd.print(" Jarak: "); lcd.print(jarak); lcd.print(" cm"); } </pre>
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

9. Setelah memasukkan algoritma pada Arduino IDE maka Verifikasi Algoritma yang sudah anda buat agar bisa dideteksi kesalahan dalam penulisan programnya 
10. Compile Algoritma pada modul Arduino agar bisa dilihat hasilnya pada LCD .
11. Lihat jarak yang dibaca ultrasonik pada tampilan LCD dengan memberikan halangan buku/kertas pada sensor ultrasonik.
12. Bandingkan jarak yang terbaca oleh sensor ultrasonik dengan penggaris (alat standar).
13. Hitung selisih yang terbaca.
14. Isi tabel pengambilan data sebagai berikut :

No	Jarak kertas dengan ultrasound (penggaris)	Nilai jarak yang terbaca di LCD	Selisi	Ketepatan (%)
1	20 cm			
2	30 cm			
3	40 cm			
4	50 cm			
5	60 cm			

15. Gunakan Persamaan pengukuran ketepatan sebagai berikut :

$$\text{Ketepatan} = p = \left| 1 - \left| \frac{Y_n - X_n}{X_n} \right| \right| \times 100\%$$

Dimana : Y_n = Hasil pengukuran dengan penggaris

X_n = Nilai yang terbaca di LCD

C. TUGAS

1. Cobalah uji sensor ultrasonic dengan halangan benda padat, apakah sensor dapat mengirimkan informasi jarak pada LCD.
2. Cobalah uji sensor ultrasonic dengan halangan benda cairn (air dalam gelas) yang bening, apakah sensor dapat mengirimkan informasi jarak pada LCD.

3. Hitung secara manual dengan ketepatan sensor ultrasound dalam membaca jarak halangan pada tabel langkah 4 diatas

BAB IV

RANGKAIAN I2C LCD 16x2

INSTRUMENTASI I2C LCD 16x2

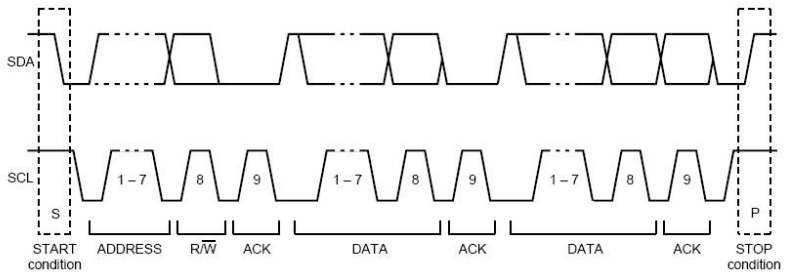
a. TUJUAN

Meminimalisir penggunaan pin Arduino untuk LCD 16x2 dengan memanfaatkan I2C dalam menampilkan karakter huruf maupun angka.

b. TEORI

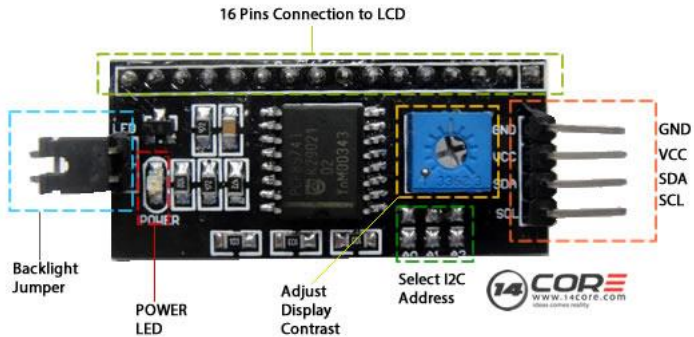
Konsep I2C *Serial Bus*

Untuk menghubungkan antar processor atau controller dibutuhkan suatu system bus tertentu. Bus ini harus dapat melewati data atau parameter control antar processor yang berfungsi sebagai jaringan atau networking. Alasan penggunaan *serial bus* adalah implementasinya yang lebih mudah dalam sisi pengkabelan dan lebih mudah dari pada ada *parallel bus*. I2C *serial bus* dikembangkan oleh Philips sejak tahun 1980 ditunjukkan untuk peggaplikasian control dengan mikrokontroler. Pada dokumen resmi yang dikeluarkan oleh Philips semiconductor memiliki 3 buah kecepatan pada I2C serial bus, yakni 100 KHz (Standard mode), 400 KHz (Fast-mode) dan 3.4 MHz (Highspeed mode). Komunikasi pada I2C serial bus merupakan komunikasi serial sinkron, artinya data pada bus dikirim *bit per bit* (serial) dan *clock* ikut dikirimkan (sinkron). Sebagai saluran *clock berupa SCL line*. Dan satu buah saluran *common ground* sebagai referensi. (Mubarak T, Sumaryo, Abdurohman, 2008).



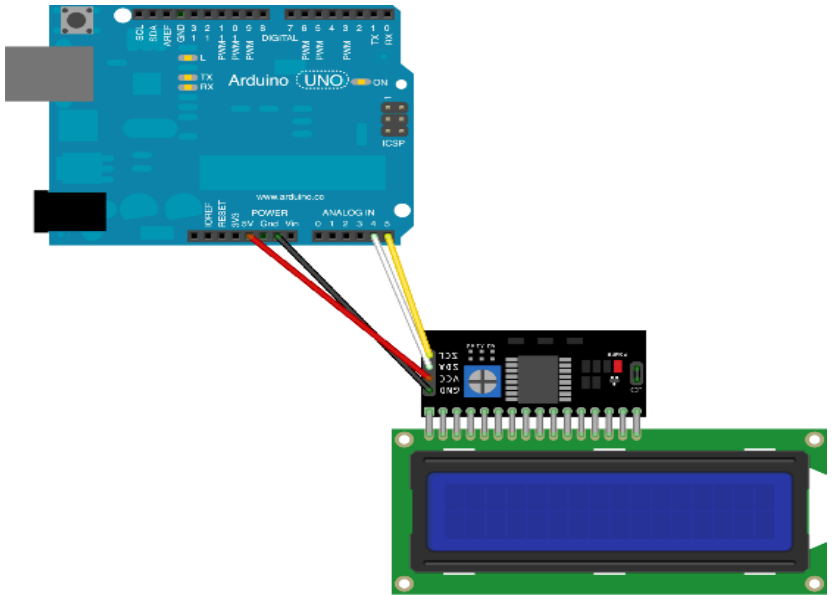
Gambar 1. *Data transfer* pada *I2C serial bus*. (Mubarak T, Sumaryo, Abdurohman, 2008)

Modul LCD I2C backpack interface ini meminimalisir jumlah pin mikrokontroler yang digunakan untuk mengendalikan LCD. Awalnya pakai 8/4 pin data, skarang cukup 2 yakni SCL dan SDA.



Gambar 2. Tampilan I2C

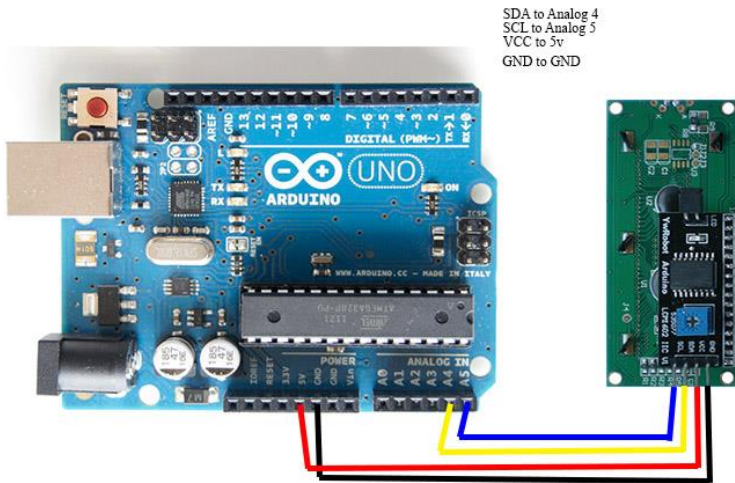
Material yang digunakan dalam latihan penggunaan *I2C* dapat dilakukan dengan cara berikut.



Gambar 3. Rangkaian I2C

1. Siapkan Arduino Uno sebelum merakit rangkaian I2C.
2. Siapkan LCD 16x2 dan I2C.
3. Siapkan kabel jumper secukupnya untuk pemasangan I2C ke Arduino Uno.
4. Pasanglah I2C ke dalam pin-pin LCD yang berjumlah 16 kaki.
5. Solder kaki I2C pada pin-pin LCD yang berjumlah 16 kaki.
6. Masukkan kabel jumper ke Ground pada I2C, kemudian masukkan kembali kabel jumper menuju Ground pada Arduino Uno.
7. Masukkan kabel jumper ke VCC pada I2C, kemudian masukkan kembali kabel jumper menuju VCC atau tegangan pada Arduino Uno.

- Masukkan kabel jumper ke **SDA** pada I2C, kemudian masukkan kembali menuju **Pin Analog** no. 4 pada Arduino Uno.
- Masukkan kabel jumper ke **SCL** pada I2C, kemudian masukkan kembali menuju **Pin Analog** no. 5 pada Arduino Uno sebagaimana gambar berikut ini.



Gambar 4. Penempatan SDA dan SCL I2C ke Pin Analog Arduino

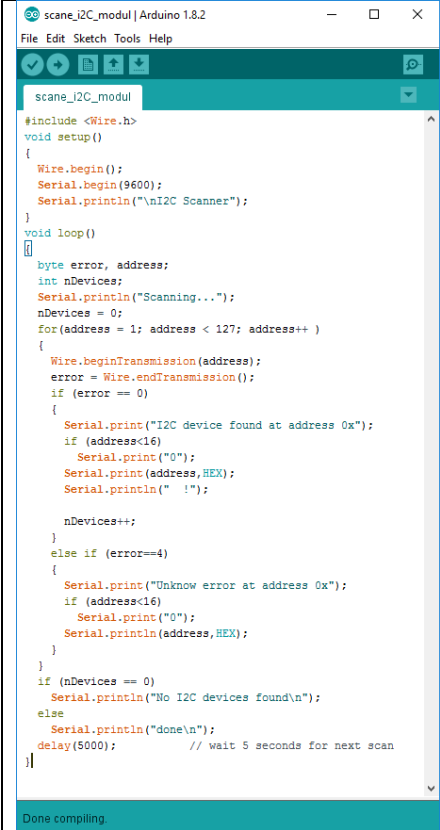
❖ **Persiapan secara Software *Arduino IDE*:**

- Nyalakan komputer dan Buka Program Arduino IDE



- Scane dan isikan program Arduino dengan algoritma seperti Gambar 5 dibawah ini agar Arduino bisa dijalankan dengan I2C. Tanpa mengisikan program ini kedalam arduino, kemungkinan besar saat ngompail

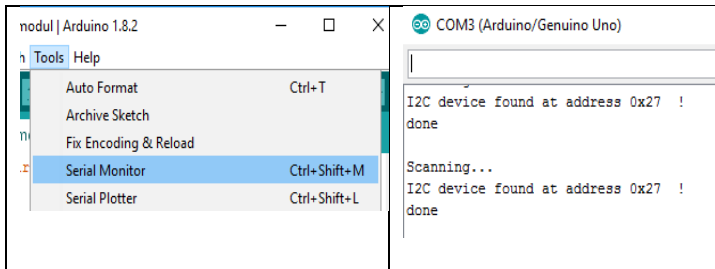
program yang diinginkan ke I2C akan gagal. Adapun programnya adalah sebagai berikut. Secara file.ino akan disediakan oleh asisten lab prodi teknik elektro (di file My Document dengan nama folder "Scane I2C").

 <pre> scane_i2C_modul Arduino 1.8.2 File Edit Sketch Tools Help scane_i2C_modul #include <Wire.h> void setup() { Wire.begin(); Serial.begin(9600); Serial.println("\nI2C Scanner"); } void loop() { byte error, address; int nDevices; Serial.println("Scanning..."); nDevices = 0; for(address = 1; address < 127; address++) { Wire.beginTransmission(address); error = Wire.endTransmission(); if (error == 0) { Serial.print("I2C device found at address 0x"); if (address<16) Serial.print("0"); Serial.print(address,HEX); Serial.println(" !"); nDevices++; } else if (error==4) { Serial.print("Unknow error at address 0x"); if (address<16) Serial.print("0"); Serial.println(address,HEX); } } if (nDevices == 0) Serial.println("No I2C devices found\n"); else Serial.println("done\n"); delay(5000); // wait 5 seconds for next scan } Done compiling </pre>	<pre> #include <Wire.h> void setup() { Wire.begin(); Serial.begin(9600); Serial.println("\nI2C Scanner"); } void loop() { byte error, address; int nDevices; Serial.println("Scanning..."); nDevices = 0; for(address = 1; address < 127; address++) { Wire.beginTransmission(address s); error = Wire.endTransmission(); if (error == 0) { Serial.print("I2C device found at address 0x"); if (address<16) </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



	<pre> Serial.print("0"); Serial.print(address,HEX); Serial.println(" !"); nDevices++; } else if (error==4) { Serial.print("Unknow error at address 0x"); if (address<16) Serial.print("0"); Serial.println(address,HEX); } } if (nDevices == 0) Serial.println("No I2C devices found\n"); else Serial.println("done\n"); delay(5000); // wait 5 seconds for next scan } </pre>
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

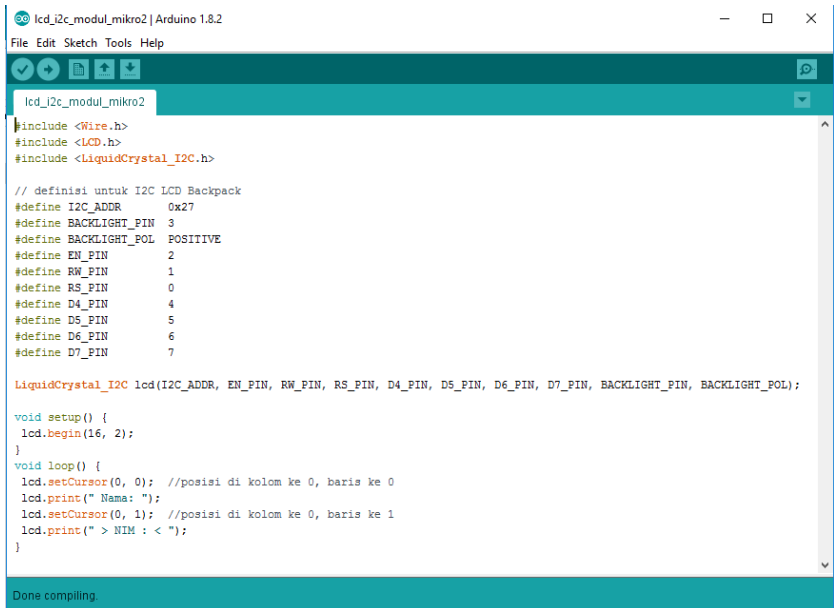
Gambar 5. Scane Arduino sebelum membuat karakter huruf atau angka pada LCD 16x2 menggunakan i2C

3. Setelah Algoritma sebagaimana gambar 5 diatas di Compail maka klik pada menu Tools → pilih Serial Monitor. Sehingga akan muncul gambar seperti berikut ini :



Gambar 6. Serial Monitor

4. Ketik Algoritma sebagaimana contoh pada Gambar 7a atau 7b yang merupakan Algoritma dalam Menampilkan karakter di LCD 16x2 dengan i2C.
5. Setelah memasukkan algoritma pada Arduino IDE maka Verifikasi Algoritma yang sudah anda buat agar bisa dideteksi kesalahan dalam penulisan programnya 
6. Compile Algoritma pada modul Arduino agar bisa dilihat hasilnya pada LCD i2C 16x2 .
7. Lihat hasilnya pada tampilan LCD.



```
lcd_i2c_modul_mikro2 | Arduino 1.8.2
File Edit Sketch Tools Help

lcd_i2c_modul_mikro2

#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>

// definisi untuk I2C LCD Backpack
#define I2C_ADDR 0x27
#define BACKLIGHT_PIN 3
#define BACKLIGHT_POL POSITIVE
#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

LiquidCrystal_I2C lcd(I2C_ADDR, EN_PIN, RW_PIN, RS_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN, BACKLIGHT_PIN, BACKLIGHT_POL);

void setup() {
  lcd.begin(16, 2);
}

void loop() {
  lcd.setCursor(0, 0); //posisi di kolom ke 0, baris ke 0
  lcd.print(" Nama: ");
  lcd.setCursor(0, 1); //posisi di kolom ke 0, baris ke 1
  lcd.print(" > NIM : < ");
}

Done compiling.
```

Gambar 7a. Algoritma pada IDE Arduino

```

#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
// definisi untuk I2C LCD Backpack
#define I2C_ADDR    0x27
#define BACKLIGHT_PIN 3
#define BACKLIGHT_POL POSITIVE
#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
LiquidCrystal_I2C  lcd(I2C_ADDR,  EN_PIN,
RW_PIN, RS_PIN, D4_PIN, D5_PIN, D6_PIN,
D7_PIN, BACKLIGHT_PIN, BACKLIGHT_POL);
void setup() {
  lcd.begin(16, 2);
}
void loop() {
  lcd.setCursor(0, 0); //posisi di kolom ke 0, baris
ke 0
  lcd.print(" Nama: ");
  lcd.setCursor(0, 1); //posisi di kolom ke 0, baris
ke 1
  lcd.print("> NIM : <");
}

```

Gambar 7b. Algoritma pada IDE Arduino

8. Tulis karakter berdasarkan koordinat (kolom, Baris) untuk keterangan seperti sebagaimana tabel berikut ini, dan Foto Hasilnya untuk dimasukkan pada Laporan Praktikum.

No	Koordinat (Kolom, Baris)	Karakter	Hasil Foto
1	(2,0)	Nama	
2	(1,1)	NIM	
3	(2,1)	Jurusan	
4	(0,0)	Institusi	
5	(5,1)	Hobi	

C. TUGAS

1. Buat Algoritma berjalan dari kiri ke kanan untuk Nama dan NIM pada LCD menggunakan **Algoritma Utama** dibawah ini :

<pre> Jalan_kiri_kanan // Gerakan Ke kanan for (int pos=0; pos < 16; pos++) { lcd.scrollDisplayRight(); delay(500); } lcd.clear(); lcd.print("Selamat Bekerja"); lcd.setCursor(0,1); lcd.print("Tetap Semangat"); delay(2000); for (int jos=0; jos < 16; jos++) { lcd.scrollDisplayLeft(); delay(500); } </pre>	<pre> // Gerakan Ke kanan for (int pos=0; pos < 16; pos++) { lcd.scrollDisplayRig ht(); delay(500); } lcd.clear(); lcd.print("Selamat Bekerja"); lcd.setCursor(0,1); lcd.print("Tetap Semangat"); delay(2000); for (int jos=0; jos < 16; jos++) { lcd.scrollDisplayLef t(); delay(500); } </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	}
--	---

NB : Anda harus tetap mendeklarasikan sesuai panduan awal, karena Algoritma Utama di sini hanya program ini, bukan program keseluruhan.

BAB V

RANGAKAIAN SENSOR LM 35

SENSOR LM 35

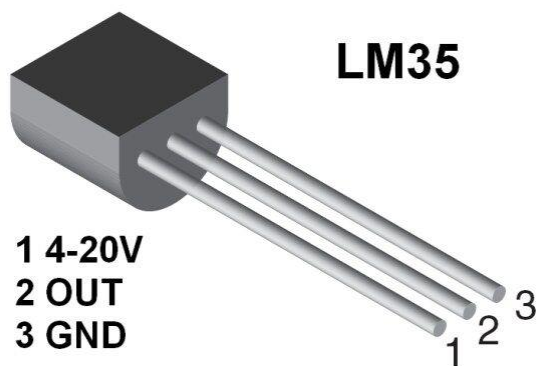
A. TUJUAN

Mendeteksi Suhu Ruang dengan sensor LM 35.

B. TEORI

LM35 adalah komponen sensor suhu berukuran kecil seperti transistor (TO-92), komponen ini mampu mengukur suhu hingga 100°C. Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan.

LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus seperti arduino, serta tidak memerlukan penyetelan lanjutan. Dengan tegangan keluaran yang terskala linear dengan suhu terukur,



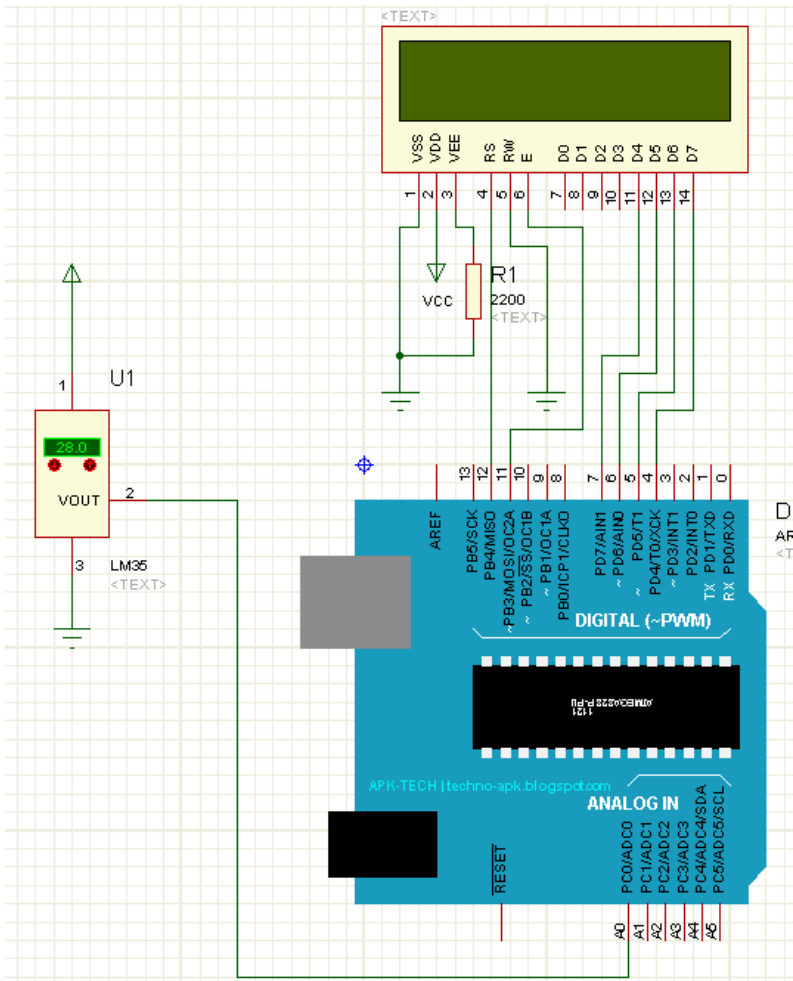
yakni 10 milivolt per 1 derajat celcius.

Gambar 1. Sensor suhu LM 35 (Innovative, 2017)

Dari gambar diatas dapat diketahui bahwa sensor suhu IC LM35 pada dasarnya memiliki 3 pin diantaranya yaitu, pin 1 berfungsi sebagai sumber tegangan kerja dari LM35, pin 2 atau tengah digunakan sebagai tegangan keluaran atau V dengan jangkauan kerja dari 0 Volt sampai dengan 1,5 Volt dengan out tegangan operasi sensor LM35 yang dapat digunakan antar 4 Volt sampai 30 Volt. Keluaran sensor ini akan naik sebesar 10 mV setiap derajat celcius. Tegangan keluaran sensor yang terskala linear terhadap suhu terukur, yakni 10 milivolt per 1 derajat celcius. Jadi jika $V_{out} = 530\text{mV}$, maka suhu terukur adalah 53 derajat Celcius. Dan jika $V_{out} = 320\text{mV}$, maka suhu terukur adalah 32 derajat Celcius.

Simulasi Menampilkan Suhu di layar LCD dengan memakai LM 35 dapat dilakukan dengan cara sebagai berikut.

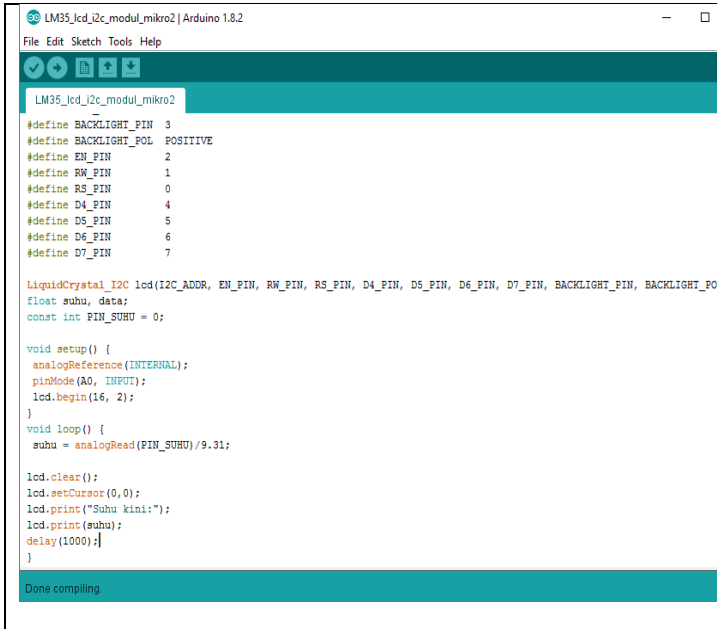
1. Siapkan Kit Arduino dan LCD 16x2; kabel jumper penghubung, dan sensor LM35. Gunakan gambar 2 Skematik Rangkain *LCD* sebagai ilustrasi dalam prakteknya



Gambar 2. Skematik Sensor Suhu LM35

2. Sensor suhu LM35 vout nya masuk ke pin analog yaitu pin A0 pada arduino.

3. Pemasangan jumper LCD sebagaimana Modul 4 (IV) pada petunjuk praktikum ini. Gunakan I2C untuk menampilkan nilai suhu di LCD 16x2.
4. Program Arduino IDE seperti gambar dan keterangan koding berikut ini :



```
LM35_lcd_i2c_modul_mikro2 | Arduino 1.8.2
File Edit Sketch Tools Help

LM35_lcd_i2c_modul_mikro2
#define BACKLIGHT_PIN 3
#define BACKLIGHT_POL POSITIVE
#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

LiquidCrystal_I2C lcd(I2C_ADDR, EN_PIN, RW_PIN, RS_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN, BACKLIGHT_PIN, BACKLIGHT_POL)
float suhu, data;
const int PIN_SUHU = 0;

void setup() {
  analogReference(EXTERNAL);
  pinMode(A0, INPUT);
  lcd.begin(16, 2);
}

void loop() {
  suhu = analogRead(PIN_SUHU)/9.31;

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Suhu kini:");
  lcd.print(suhu);
  delay(1000);}
}

Done compiling
```

Gambar 3. Algoritma membaca suhu dengan LM35



```
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
// definisi untuk I2C LCD Backpack
#define I2C_ADDR 0x27
```

```

#define BACKLIGHT_PIN 3
#define BACKLIGHT_POL POSITIVE
#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
LiquidCrystal_I2C lcd(I2C_ADDR, EN_PIN,
RW_PIN, RS_PIN, D4_PIN, D5_PIN, D6_PIN,
D7_PIN, BACKLIGHT_PIN, BACKLIGHT_POL);
float suhu, data;
const int PIN_SUHU = 0;
void setup() {
analogReference(INTERNAL);
pinMode(A0, INPUT);
lcd.begin(16, 2);
}
void loop() {
suhu = analogRead(PIN_SUHU)/9.31;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Suhu kini:");
lcd.print(suhu);
delay(1000);
}

```

Gambar 4. Algoritma membaca suhu dengan LM35

5. Setelah memasukkan algoritma pada Arduino IDE maka Verifikasi Algoritma yang sudah anda buat agar bisa dideteksi kesalahan dalam penulisan programnya 
6. Compile Algoritma pada modul Arduino agar bisa dilihat hasilnya pada LCD .
7. Lihat suhu yang dibaca LM35 pada tampilan LCD dengan mendekati pada kondisi panas (korek api/lilin) sebagai variabel.
8. Bandingkan suhu yang terbaca oleh sensor ultrasonik dengan Termometer Infrared (alat standar).
9. Hitung selisi yang terbaca.
10. Isi tabel pengambilan data sebagai berikut :

No	Suhu LM35 (°C)	Nilai Suhu pada alat Standar (°C)	Selisi	Ketepatan (%)
1	10 detik pertama			
2	5 detik kedua			
3	5 detik ketiga			
4	5 detik keempat			
5	5 detik kelima			

11. Gunakan Persamaan pengukuran ketepatan sebagai berikut :

$$\text{Ketetapan} = p = \left| 1 - \left| \frac{Y_n - X_n}{X_n} \right| \right| \times 100\%$$

Dimana: Y_n = Hasil pengukuran suhu dengan Alat Standar ($^{\circ}\text{C}$)

X_n = Nilai suhu yang terbaca di LCD ($^{\circ}\text{C}$)

TUGAS

1. Cobalah uji sensor LM35 dengan suhu tubuh anda dengan cara memegang LM35, apakah sensor dapat mengirimkan informasi suhu pada LCD.
2. Hitung secara manual pada laporan praktikum anda ketepatan sensor LM35 dalam membaca suhu pada tabel langkah 10 diatas.

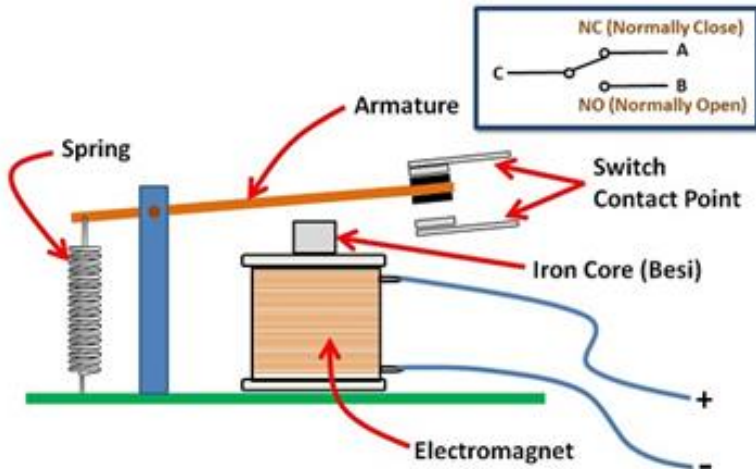
BAB VI RANGKAIAN RELAY

RELAY

B. TUJUAN

Membuat, mengontrol aplikasi dan menguji pembuatan instrument arduino dengan menerapkan komponen relay.

C. TEORI



Gambar 1. Struktur sederhana Relay

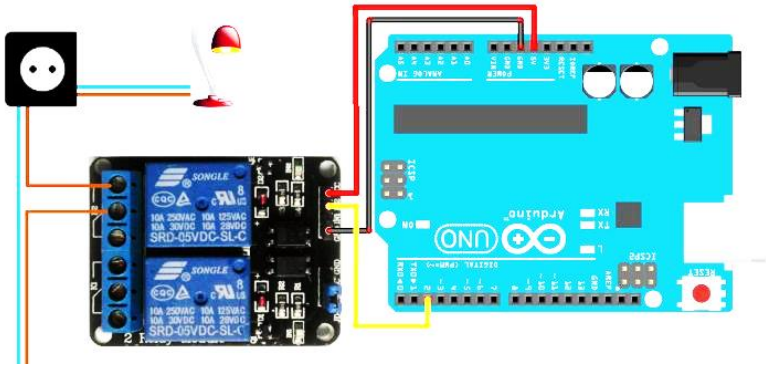
Relay merupakan suatu Saklar (*Switch*) elektromagnetik yang apabila kumparan pada relay di aliri arus listrik dapat membuat Inti Besi (*Iron Core*) menjadi magnet sehingga dapat menarik kontak (*Armatu*) jika gaya magnet yang dihasilkan menjadi lebih besar dari pada gaya pegas (*spring*) yang ada sebagaimana ilustrasi pada gambar (aziz, 2009). Relay yang menggunakan Elektromagnet 5V dan 50

mA mampu menggerakkan *Armature Relay* (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A. Berdasarkan pada Gambar diatas Kontak Poin Relay terdiri dari 2 jenis yaitu :

- *Normally Close* (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi CLOSE (tertutup)
- *Normally Open* (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi OPEN (terbuka)

Pada gambar diatas apabila Kumparan Coil dilewati arus listrik, maka akan timbul gaya Elektromagnet yang akan menarik *Armature* untuk berpindah dari Posisi sebelumnya (NC) ke posisi baru (NO), sehingga Saklar dapat menghantarkan arus listrik di posisi (NO). Sehingga Posisi awal *Armature* sebelumnya (NC) menjadi tidak terhubung. Begitu sebaliknya ketika Kumparan Coil tidak dialiri arus listrik maka *Armature* akan kembali lagi ke posisi *Normally Close* (NC). Arus listrik yang dibutuhkan dalam memindahkan kontak dari NC ke NO relatif kecil. Perlu diketahui bahwa Relay ada yang aktif Low dan aktif High. Maksudnya aktif Low adalah relay akan aktif jika diberi masukan LOW atau nol, sedangkan aktif High adalah relay akan aktif jika diberi masukan HIGH atau satu.

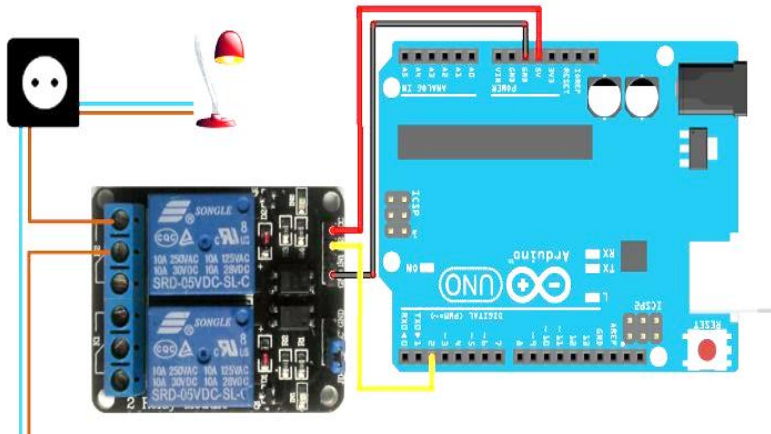
Berikut ini Contoh gambar pemasangan Relay 2 Chanel pada Arduino UNO untuk menyalakan dan mematikan Lampu.



Gambar 2. Pemasangan Relay 2 Chanel pada Arduino UNO

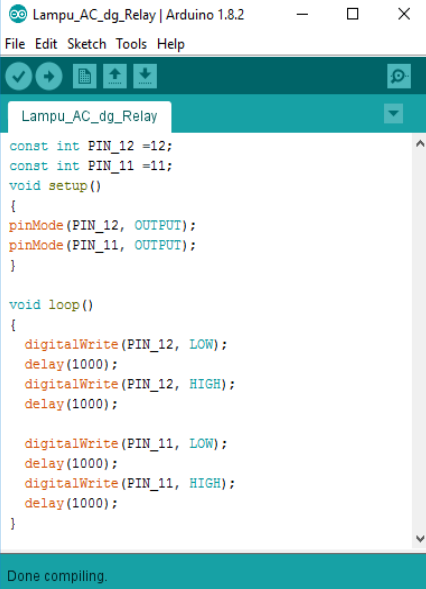
Simulasi Menghidupkan Relay dengan Arduino dapat dilakukan dengan cara sebagai berikut.

1. Siapkan Arduino Uno sebelum merakit Motor DC.
2. Gunakan Lampu AC sesuai dengan tegangan yang sudah ditentukan.
3. Lampu AC pada praktikum ini akan digerakkan dengan bantuan Supplay Tegangan ± 12 Volt sehingga butuh relay dalam pengoperasiannya. Relay yang digunakan pada praktikum ini adalah **tipe relay Aktif LOW**. Sehingga Masukkan kabel jumper warna merah pada Normally Close (NC) dan Normally Open (NO) supplay tegangan ± 12 Volt untuk menyalakan lampu seperti gambar dibawah ini. Taruh kabel hitam Lampu AC pada ground.



Gambar 3. Contoh Pemasangan Relay 2 Chanel pada Arduino UNO

4. Karena Lampunya ada 2 maka ulangi langkah 3 untuk menyalakan lampu lainnya.
5. Masukkan kabel jumper pada Pin 12 Arduino (digital) dan hubungkan pin tersebut ke pin masukan (Input) Relay 4 Chanel.
6. Masukkan kabel jumper pada Pin 11 Arduino (digital) dan hubungkan pin tersebut ke pin masukan (Input) Relay 4 Chanel.
7. Masukkan Ground Relay 4 Chanel ke ground Arduino Uno. Serta Masukkan supplay (vcc) Relay 4 Chanel ke vcc Arduino Uno.
8. Program Arduino IDE seperti coding dibawah ini :


 <pre> const int PIN_12 =12; const int PIN_11 =11; void setup() { pinMode(PIN_12, OUTPUT); pinMode(PIN_11, OUTPUT); } void loop() { digitalWrite(PIN_12, LOW); delay(1000); digitalWrite(PIN_12, HIGH); delay(1000); digitalWrite(PIN_11, LOW); delay(1000); digitalWrite(PIN_11, HIGH); delay(1000); } </pre> <p>Done compiling.</p>	<pre> const int PIN_12 =12; const int PIN_11 =11; void setup() { pinMode(PIN_12, OUTPUT); pinMode(PIN_11, OUTPUT); } void loop() { digitalWrite(PIN_12, LOW); delay(1000); digitalWrite(PIN_12, HIGH); delay(1000); digitalWrite(PIN_11, LOW); delay(1000); digitalWrite(PIN_11, HIGH); delay(1000); } </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Gambar 4. Algoritma Nyalakan lampu AC dengan Relay

9. Setelah memasukkan algoritma pada Arduino IDE maka Verifikasi Algoritma yang sudah anda buat agar bisa

dideteksi kesalahan dalam penulisan programnya 

10. Compile Algoritma pada modul Arduino agar bisa dilihat

hasilnya pada putaran motor .

11. Isi tabel berikut ini untuk menggerakkan Motor jika ketentuannya sebagai berikut :

No	Posisi PIN Arduino	Delay	Kondisi lampu (nyala/Tidak)	Foto Hasil
1	Pin 13	2000		
2	Pin 9	1000		
3	Pin 6	1500		
4	Pin 2	2500		
5	Pin 4	1000		

D. TUGAS

1. Apakah Lampu AC dapat menyala dengan baik jika kabel jumper antara relay dengan arduino dimasukkan di Pin Analog Arduino (A0) ?
2. Apa alat ukur yang digunakan untuk mengukur kekuatan cahaya lampu ? dan apa satuannya ?

BAB VII

Sensor MAX30100

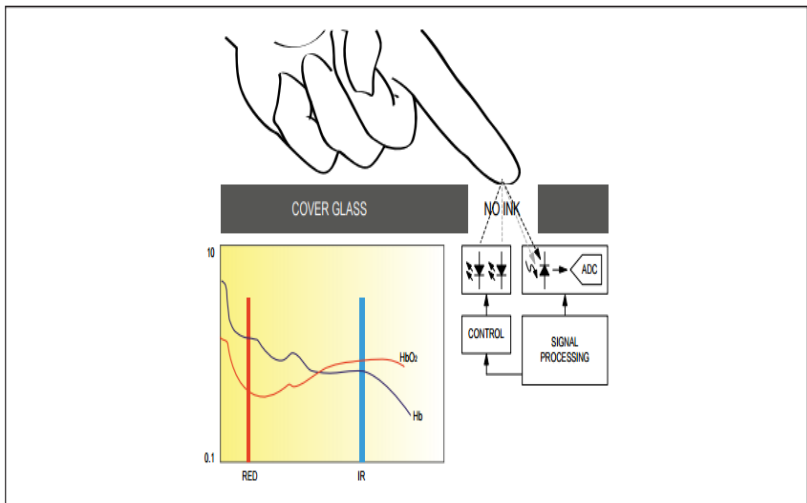
Sensor MAX30100

A. TUJUAN

Memahami Prinsip Kerja Sensor MAX30100, dan mampu memprogram dan mengoperasikan sensor MAX30100 pada Instrumentasi Medis.

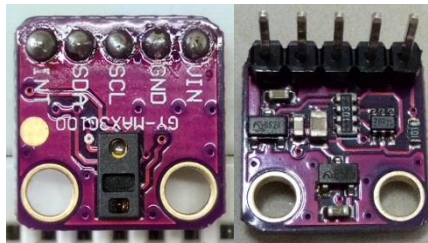
B. TEORI

Sensor MAX30100 adalah sensor kesehatan memanfaatkan sensor optik yang terintegrasi antara IR dan LED merah yang dikombinasikan dengan *photodetector*, serta pemrosesan sinyal analog rendah untuk mendeteksi dan mengukur oksimetri denyut nadi dan sinyal detak jantung melalui salah satu ujung jari tangan.



Gambar 1. Sistem Blok Diagram MAX30100

Cara kerja sensor ini adalah dari pancaran dua LED yang berwarna merah yang menghasilkan dua panjang gelombang cahaya, kemudian inframerah mengukur absorpsi darah berdenyut melalui *fotodetektor* yang menghasilkan pembacaannya. Pembacaan sensor akan tidak maksimal apabila banyak melakukan pergerakan dan terlalu banyak tekanan pada jari tangan maupun sensor mengakibatkan menyempitnya aliran darah kapiler oleh karena itu dapat mengurangi kehandalan pembacaan data pada sensor.



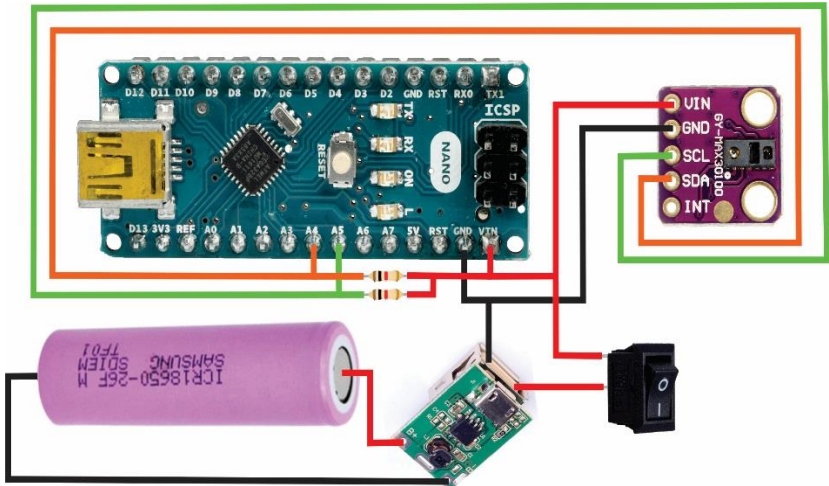
Gambar 2. Sensor MAX30100 (Rizal, 2018)

Tabel 1. Spesifikasi Sensor MAX30100

No.	Fitur	Keterangan
1.	Tegangan Operasi	5 Volt
2.	Arus Operasi	3 mA
3.	Protokol Komunikasi	I2C (SDA dan SCL)
4.	Kerja Sensor	<ul style="list-style-type: none"> • Saturasi Oksigen dalam Darah (SpO²) • Heart Rate (bpm) • Suhu Object (°C)
5.	Dimensi	14 mm x 14 mm

Pada rangkaian sensor MAX30100 ini juga menggunakan sumber tegangan pada modul powerbank 1 slot. Sensor ini

dilengkapi juga dengan rangkaian ADC sehingga data yang dikeluarkan berupa data digital, oleh karena itu data yang dikirimkan ke arduino nano memanfaatkan pin SDA dan SCL juga.



Gambar 3. Rangkaian Sensor MAX30100 (Rizal, 2018)

Pada Gambar 3. rangkaian sensor MAX30100 dengan mikrokontroler arduino nano diatas dengan port yang terhubung adalah pin SDA sensor MAX30100 dihubungkan pada pin A4 arduino nano, sedangkan pin SCL sensor MAX30100 dihubungkan ke pin A5 arduino nano. Untuk tabel dari rangkaian sensor MAX30100 dapat dilihat pada Tabel 2.

Tabel 2 Pengalamatan Pin Rangkaian Keseluruhan

No .	Alamat Port Arduino	Alamat Port Alat	Hardware Sensor
1	SDA	SDA	

2	SCL	SCL	Sensor MAX30100
3	VCC 5v Eksternal	VCC	
4	GND	GND	

Tabel Pemrograman

```

MAX_30100_Android | Arduino 1.8.5
File Edit Sketch Tools Help
MAX_30100_Android$
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"
#define REPORTING_PERIOD_MS 1000
PulseOximeter pox;
uint32_t tsLastReport = 0;
void onBeatDetected()
{Serial.println();}
void setup()
{Serial.begin(9600);
//Serial.println("Initializing MAX30100");
pox.begin();
pox.setOnBeatDetectedCallback(onBeatDetected);}
void loop()
{
pox.update();
if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
Serial.print("SpO2 : ");
Serial.print(pox.getSpO2());
Serial.print(" % ");
Serial.print("Heart : ");
Serial.print(pox.getHeartRate());
Serial.println(" bpm");
Serial.print("Temp : ");
Serial.print(pox.getTemperature());
Serial.println(" *C");
tsLastReport = millis();
}
}

```

```

#include <Wire.h>
#include
    "MAX30100_PulseOximeter.h"
#define    REPORTING_PERIOD_MS
    1000
PulseOximeter pox;
uint32_t tsLastReport = 0;
void onBeatDetected()
{
    Serial.println();
}
void setup()
{
    Serial.begin(9600);
    //Serial.println("Initializing
        MAX30100");
    pox.begin();
    pox.setOnBeatDetectedCallback
        (onBeatDetected);
}

```

```
void loop()
{
  pox.update();
  if (millis() - tsLastReport >
    REPORTING_PERIOD_MS) {
    Serial.print("SpO2 : ");
    Serial.print(pox.getSpO2());
    Serial.print(" % ");
    Serial.print("Heart : ");
    Serial.print(pox.getHeartRate());
    Serial.println(" bpm");
    Serial.print("Temp : ");
    Serial.print(pox.getTemperature());
    Serial.println(" *C");
    tsLastReport = millis();
  }
}
```

C. TUGAS

1. Jalankan pemrograman Sensor MAX30100 menggunakan LCD 16x2
2. Tampilkan nilai Sensor MAX30100 di Smartphone Android menggunakan Komunikasi Bluetooth.

BAB VIII

Sensor Suhu DS18B20

A. TUJUAN

Memahami Prinsip Kerja Sensor **DS18B20**, dan mampu memprogram dan mengoperasikan sensor **DS18B20** pada Instrumentasi Medis.

B. TEORI

Sensor DS18B20 adalah sensor untuk mengukur suhu badan manusia yang memiliki akurasi nilai dan kestabilan lebih baik dari sensor Lm35. DS18B20 adalah sensor suhu digital atau untuk pembacaan suhu, sensor ini tahan air dikarenakan sensor terlindungi oleh casing waterproof serta memiliki 3 pin yang terdiri dari vcc, Ground dan Data Input/Output. Berikut konfigurasi pin Ds18b20.

Tabel 1. Konfigurasi Pin Sensor DS18B20

No	Nama Pin	Fungsi Pin
1	Merah	Sumber tegangan 5 volt dc
2	Hitam	0 volt dc / Ground
3	Kuning	Input Sensor

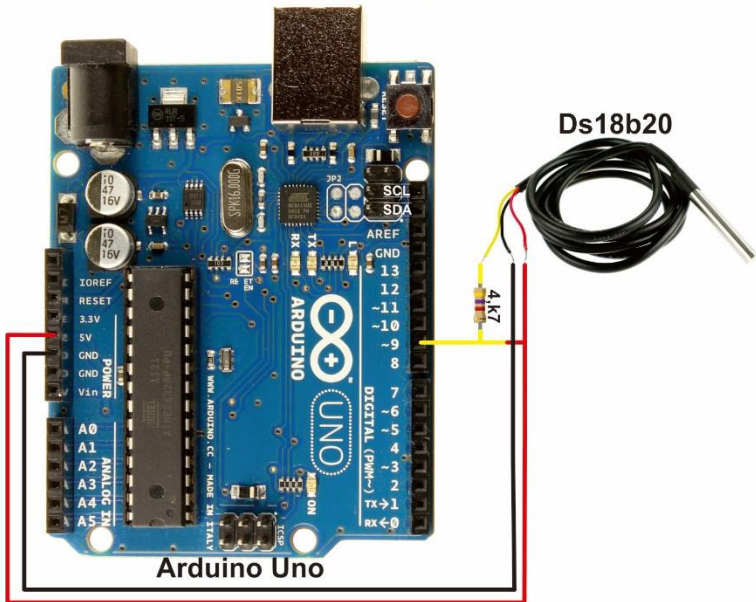


Gambar 2. Sensor DS18B20

Tabel 3. Spesifikasi Sensor DS18B20

No	Fitur	Keterangan
1	Tegangan	5 VDC
2	Arus	1 mA
3	Jumlah Pin	3 Pin
4	Deteksi Temperatur	-1.5 °C sampai 125 °C
5	Min. Deteksi Temperatur	-55 °C
6	Maks. Deteksi Temperatur	125 °C
7	Akurasi	0.5 °C

Tabel 3 di atas menerangkan tentang spesifikasi sensor DS18B20 untuk mempermudah mengetahui dalam menggunakan sensor DS18B20 ini. Sehingga tidak sampai merusak sensor jika salah penggunaannya. Dikarenakan sudah terdapat batasan-batasan tegangan sensor. Sensor Ds18b20 adalah sebuah sensor yang dapat digunakan untuk pengukuran suhu. DS18B20 merupakan sebuah sensor suhu dimana akurasi nilai suhu dan kecepatan pengukuran memiliki kestabilan yang jauh lebih baik dari sensor LM35DZ. Pada rangkaian sensor DS18B20 ini menggunakan sumber tegangan 5 Volt yang didapat dari power supply pada pin Vin.



Gambar 3. Rangkaian Arduino Uno dengan Sensor Ds18b20

Dari Gambar 3. Rangkaian sensor ds18b20 dengan mikrokontroler arduino, dengan rangkaian skematik diatas dapat diketahui port yang terhubung adalah Port digital atau PD9 arduino untuk dihubungkan ke input sensor ds18b20.

Tabel 4. Pengalamanan Port Microcontroller Arduino dengan Rangkaian Sensor Ds18b20

No.	Alamat Port Arduino	Alamat Port Alat
1.	PD 9	INPUT SENSOR
2.	VCC 5 Volt	VCC
3.	GND	GND

```

sketch_aug02a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_aug02a $
#include <OneWire.h>
#include <DallasTemperature.h>
// sensor diletakkan di pin 2
#define ONE_WIRE_BUS 2
// setup sensor
OneWire oneWire(ONE_WIRE_BUS);
// berikan nama variabel, masukkan ke pustaka Dallas
DallasTemperature sensorSuhu(&oneWire);
float suhuSekarang;
void setup(void)
{
  Serial.begin(9600);
  sensorSuhu.begin();
}
void loop(void)
{
  suhuSekarang = ambilSuhu();
  Serial.println(suhuSekarang);
  delay(2000);
}

float ambilSuhu()
{
  sensorSuhu.requestTemperatures();
  float suhu = sensorSuhu.getTempCByIndex(0);
  return suhu;
}
  
```

Gambar 4. Screen Shoot Pemrograman Arduino IDE Sensor DS18B20

```
#include <OneWire.h>
#include <DallasTemperature.h>
// sensor diletakkan di pin 2
#define ONE_WIRE_BUS 2
// setup sensor
OneWire oneWire(ONE_WIRE_BUS);
// berikan nama variabel, masukkan ke pustaka Dallas
DallasTemperature sensorSuhu(&oneWire);
float suhuSekarang;
void setup(void)
{
  Serial.begin(9600);
  sensorSuhu.begin();
}
void loop(void)
{
  suhuSekarang = ambilSuhu();
  Serial.println(suhuSekarang);
  delay(2000);
}
float ambilSuhu()
{
  sensorSuhu.requestTemperatures();
  float suhu = sensorSuhu.getTempCByIndex(0);
  return suhu;
}
```

Gambar 5. Screen Shoot Pemrograman Arduino IDE Sensor DS18B20

C. TUGAS

1. Tampilkan Nilai Sensor DS18B20 pada tampilan LCD 16x2, dan bandingkan hasilnya dengan sensor LM35
2. Hitung Akurasi ketepatan Sensor DS18B20 dengan alat standar. Cari nilai deviasi, ketepatan, akurasi, dan standar deviasinya terhadap alat standar.

BAB IX

Komunikasi Arduino dengan Android memanfaatkan Bluetooth

A. TUJUAN

Mengintegrasikan Antara Hardware dengan Android MIT App Inventor di Smartphone

B. TEORI

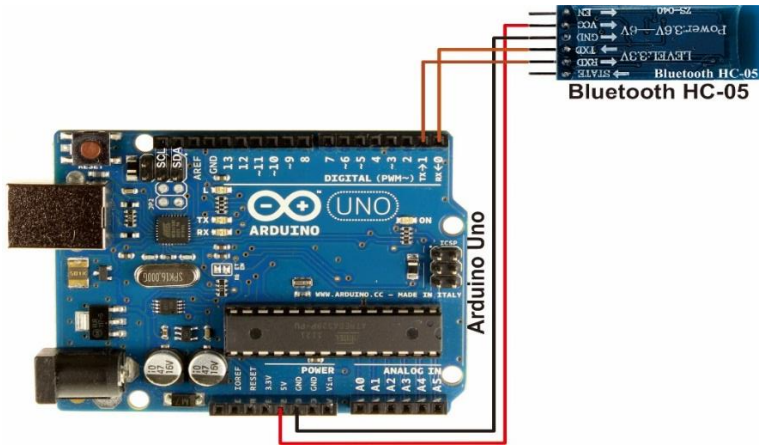
1. Bluetooth

Bluetooth adalah protokol komunikasi *wireless* yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain. Salah satu hasil contoh modul *Bluetooth* yang paling banyak digunakan adalah tipe HC-05. modul *Bluetooth* HC-05 merupakan salah satu modul *Bluetooth* yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul *Bluetooth* HC-05 terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda - beda. Untuk gambar *module bluetooth* dapat dilihat pada gambar 1 berikut.



Gambar 1. Modul *Bluetooth* HC 05

Bluetooth dalam rangkaian digunakan sebagai pengirim data atau menerima data dari nirkabel yang rx dan tx nya sama – sama bertegangan 3.3 vdc, sedangkan tegangan 5 volt digunakan untuk VCC nya. *Module Bluetooth* HC-05 merupakan *module Bluetooth* yang bisa menjadi *slave* ataupun *master* hal ini dibuktikan dengan bisa memberikan notifikasi untuk melakukan *pairing* keperangkat lain, maupun perangkat lain tersebut yang melakukan *pairing* ke *module Bluetooth HC-05* (Pratama, 2014).



Gambar 2. Rangkaian Arduino Uno dengan Bluetooth HC-05

Tabel 1. Pengalamatan Port Mikrokontroller Arduino dengan Bluetooth hc-05

No.	Alamat Port Arduino	Alamat Port Alat
1.	PD 0	TX Bluetooth
2.	PD 1	RX Bluetooth
3.	VCC 5 Volt	VCC
4.	GND	GND

2. Software Android MIT Inventor

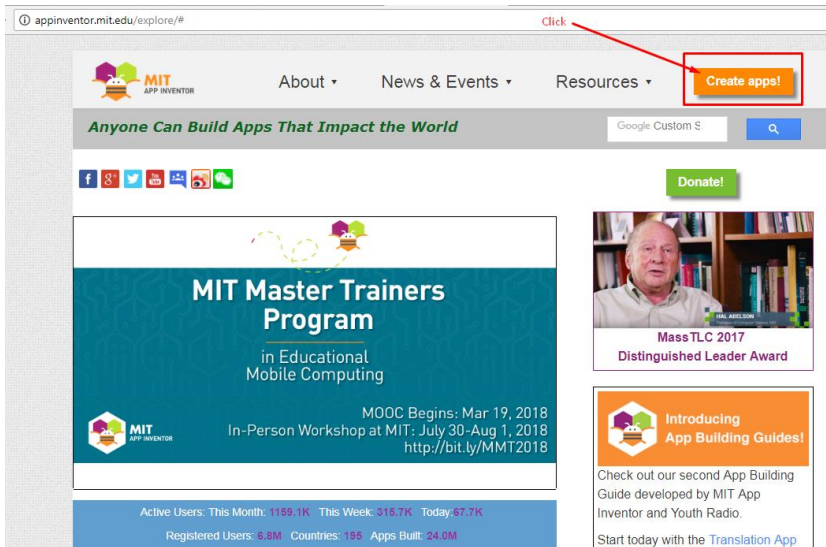
Android merupakan sistem operasi untuk perangkat *mobile* berbasis linux, yang dimodifikasi sedemikian rupa, sehingga dapat dijalankan dalam perangkat seperti komputer tablet dan *smartphone*. Dikembangkan pertama kali oleh Android.Inc yang kemudian namanya digunakan sebagai nama proyek sistem operasi android.

Untuk aplikasi android yang berjalan dalam *smartphone*, dibuat dengan menggunakan Android MIT Inventor, Pembuatan aplikasi dan coding untuk android

membutuhkan IDE (*Intergration Development Evironment*). Sehingga dengan aplikasi ini maka pengguna bisa membuat Aplikasi android sesuai dengan disain yang disukai dan di butuhkan.



Gambar 3. Tampilan Software MIT Inventor



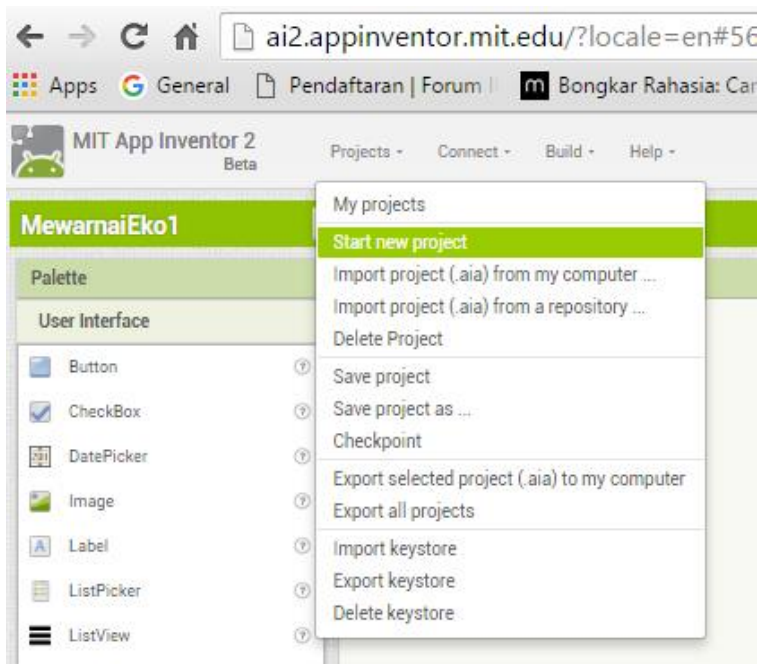
Gambar 4. Tampilan Link MIT App Inventor
(<http://appinventor.mit.edu/explore/#>)

Untuk MIT App Inventor dapat diakses di Link : <http://appinventor.mit.edu/explore/#> untuk akun masuknya menggunakan akun Gmail. Jadi pengguna harus memiliki email Gmail agar bias mengakses ke MIT App Inventor.

Berikut ini Langkah untuk membuat Aplikasi Android dengan MIT App Inventor dan mengintegrasikan Arduino menggunakan komunikasi Bluetooth.

1. Masuk Akun MIT App Inventor dengan akun Gmail anda.
2. Buat Project baru dengan Klik **Project**, dan pilih **Start new project** seperti Gambar 1, dan beri nama Project baru

anda. Usahakan dalam memberi nama tidak ada spasi, jika memungkinkan memberi nama dengan spasi maka gunakan underscors (_).

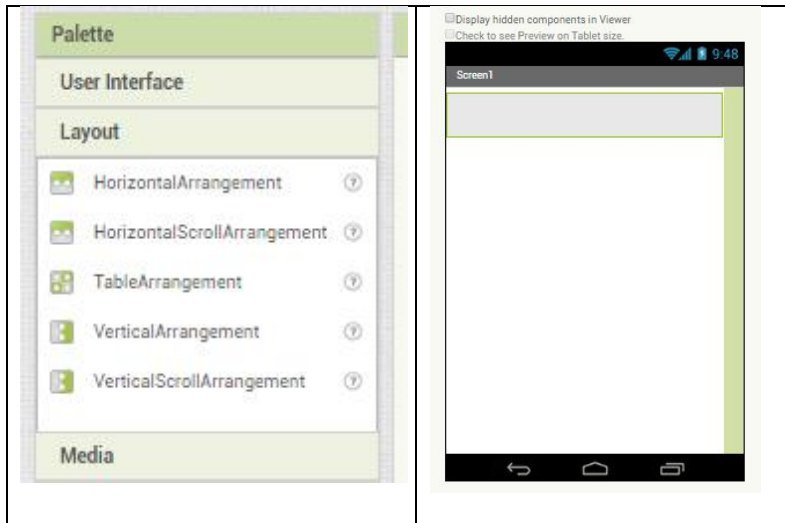


Gambar 5 Tampilan Awal Project Baru MIT App Inventor

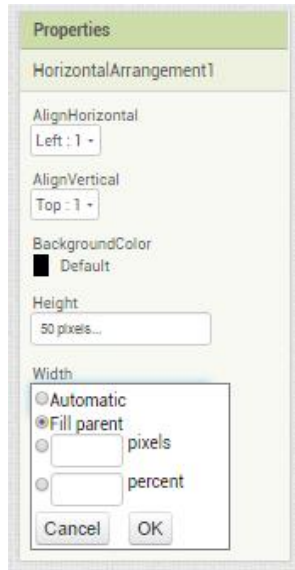


Gambar 6. Pemberian Nama Project Baru

3. Manfaatkan Pallet yang ada pada Aplikasi dengan menambahkan *HorizontalArrangement* pada Screen1 seperti gambar berikut ini. *HorizontalArrangement* disini digunakan sebagai alas untuk menata posisi menu Pallet lainnya seperti button, text, dan sebagainya agar posisi menu Pallet bisa di tempatkan di tengah, samping kiri, samping kanan, bawah, maupun di posisi atas.

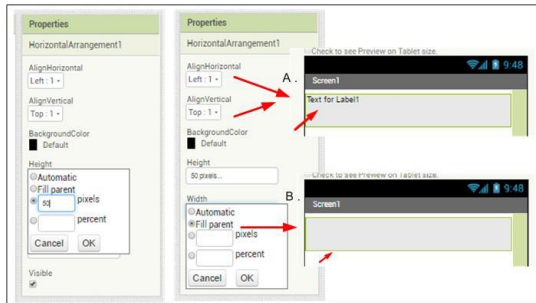


Gambar 7. Tampilan menu HorizontalArrangement

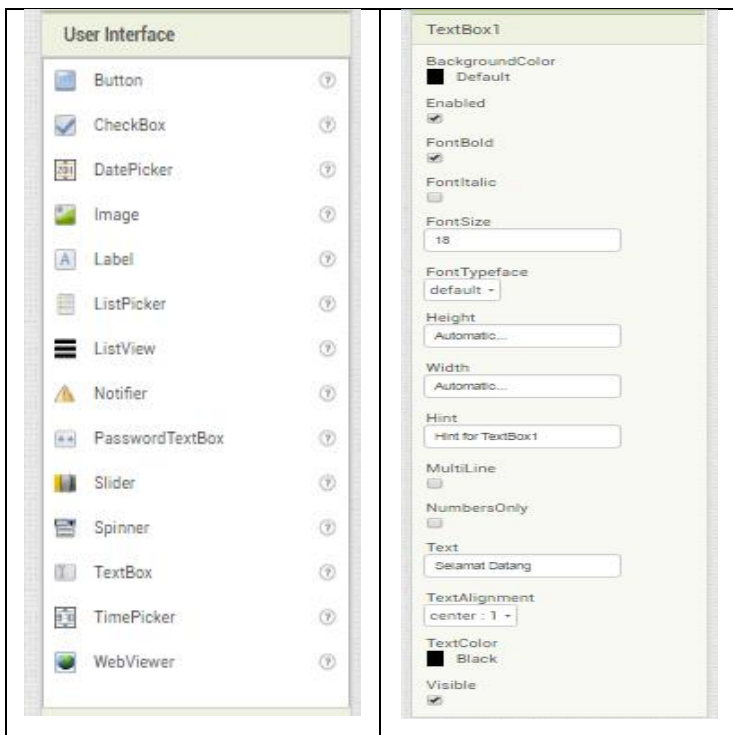


Gambar 8. Menu Pallet HorizontalArrangement

Pada menu ini anda bisa menempatkan tombol **Button** di posisi tengah, samping kanan, maupun samping kiri dengan memilih posisi **AllignHorizontal** maupun **AllignVertical**. Untuk ukuran Aplikasi agar bisa full kiri kanan sesuaidengan ukuran layer, anda bisa pilih menu **Fil parent**. Menu **pixels** digunakan untuk menentukan seberapa panjang ukuran aplikasi yang anda perlukan. Setelah semua di isi maka silahkan tekan tombol oke.



Gambar 9. Contoh setting pada HorizontalArrangement

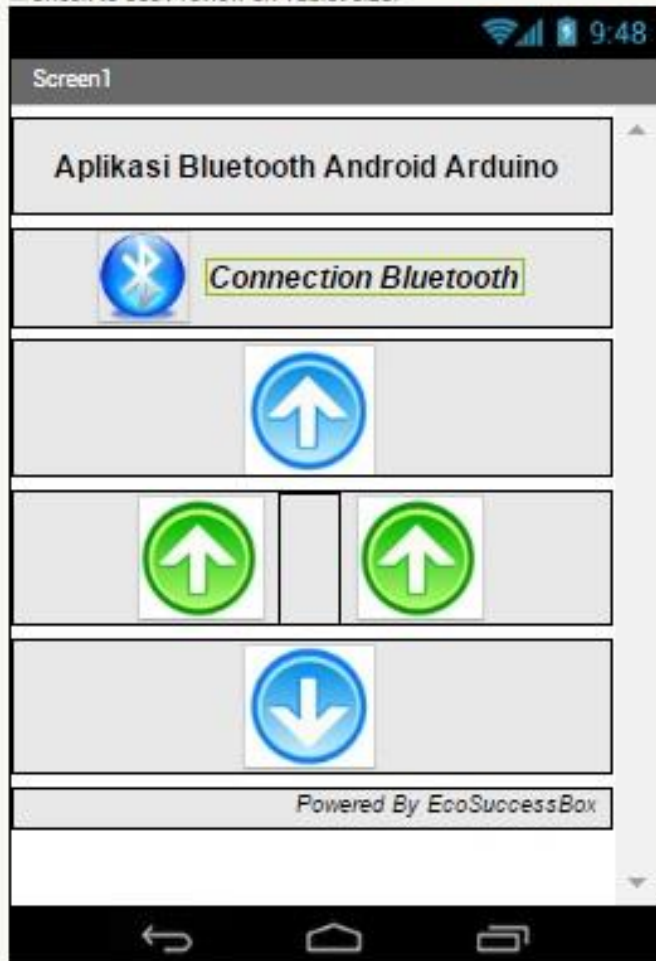


Gambar 10. Komposisi User interface

Dimenu User Interface terdapat banyak menu diantaranya Button, Label, Listpicker, image dan lainnya. Untuk membuat aplikasi Android dengan Hardware memanfaatkan Bluetooth maka gunakan tombol Button, listpicker, bluetoothclient, clock. Untuk menambahkan image seperti tanda panah anda bisa input image dengan mengganti image di menu brose pada button atau listpicke yang sudah anda pilih. Untuk tampilan dasar komunikasi Bluetooth dengan hardware disajikan dengan contoh berikut ini. Setelah itu cobalah anda membuat menu tampilan sebagaimana gambar berikut.

Display hidden components in Viewer

Check to see Preview on Tablet size.



Non-visible components



BluetoothClient1




Clock1

Gambar 11. Tampilan Aplikasi Android

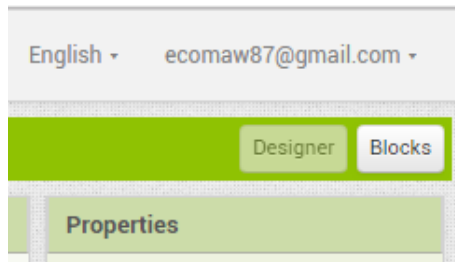
Untuk memudahkan dalam pembuatan aplikasi diatas maka perhatikan table Pembuatan menu aplisi di bawah ini.

Tabel 2. Menu Pembuatan Aplikasi

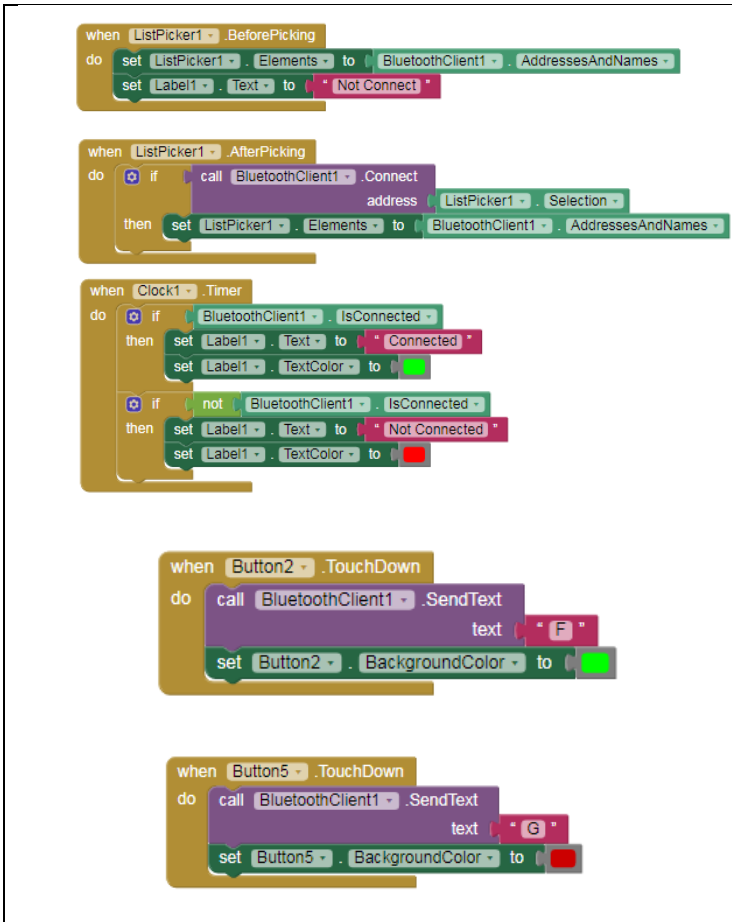
Gambar	Nama Menu	Gambar & Keterangan	Keterangan
 Label	Label1	Connection Bluetooth 	Untuk Informasi Koneksi dg Bluetooth
 ListPicker	Listpicker1		Untuk tombol koneksi bluetooth
 Button	Button2		Tombol on/off
 Button	Button5		Tombol on/off
 Connectivity  ActivityStarter  BluetoothClient	BluetoothClient	 BluetoothClient1	Koneksi bluetooth

<p>Sensors</p> <ul style="list-style-type: none"> AccelerometerSensor BarcodeScanner Clock 	<p>Clock1</p>		<p>Timer parsing data</p>
------------------------------------------------------------------------------------------------------------------------------------	---------------	-----------------------------------------------------------------------------------	-----------------------------------

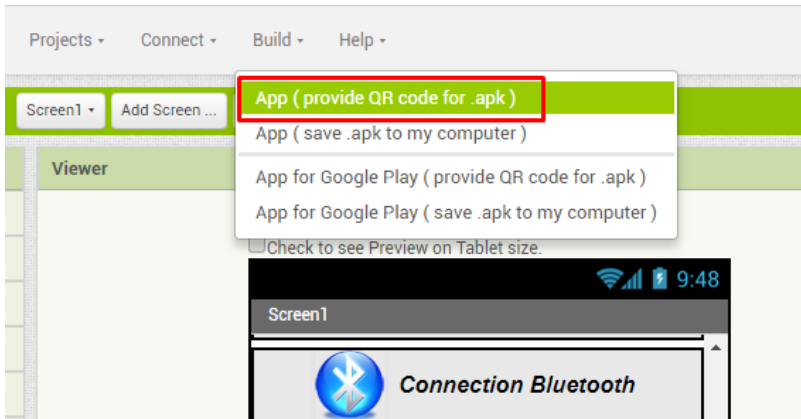
Jika semua komponen sudah terpenuhi, maka masuk ke menu pemrograman, adapun menu pemrograman dapat dilakukan dengan menekan menu blok dipojok kanan atas sebagaimana gambar berikut ini.



Gambar 12. Menu Blocks untuk masuk ke menu pemrograman

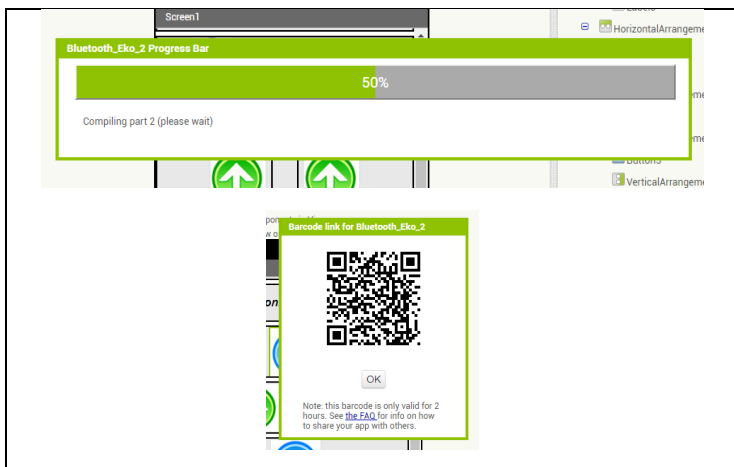


Gambar 13. Pemrograman MIT App Inventor untuk Bluetooth



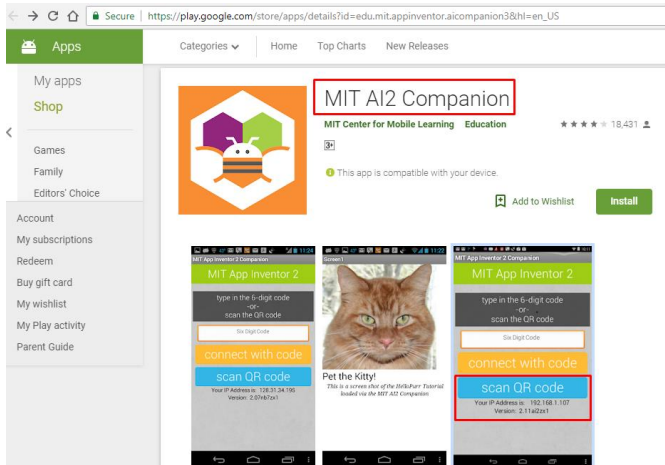
Gambar 14. Pemindahan Aplikasi Android dari desktop/PC ke Smartphone android.

Untuk memindah Aplikasi Android dari desktop/PC ke Smartphone android dilakukan dengan cara klik **Build** dan memilih App (Provide QR code for .apk) agar muncul barcode QR Code seperti gambar berikut.



Gambar 15. Barcode QR Code untuk memindah Aplikasi android dari PC ke Smartphone.

Untuk pembacaan Barcode QR Code dapat anda lakukan dengan dengan menginstall MIT AI2 Companion yang ada di google Playstore.

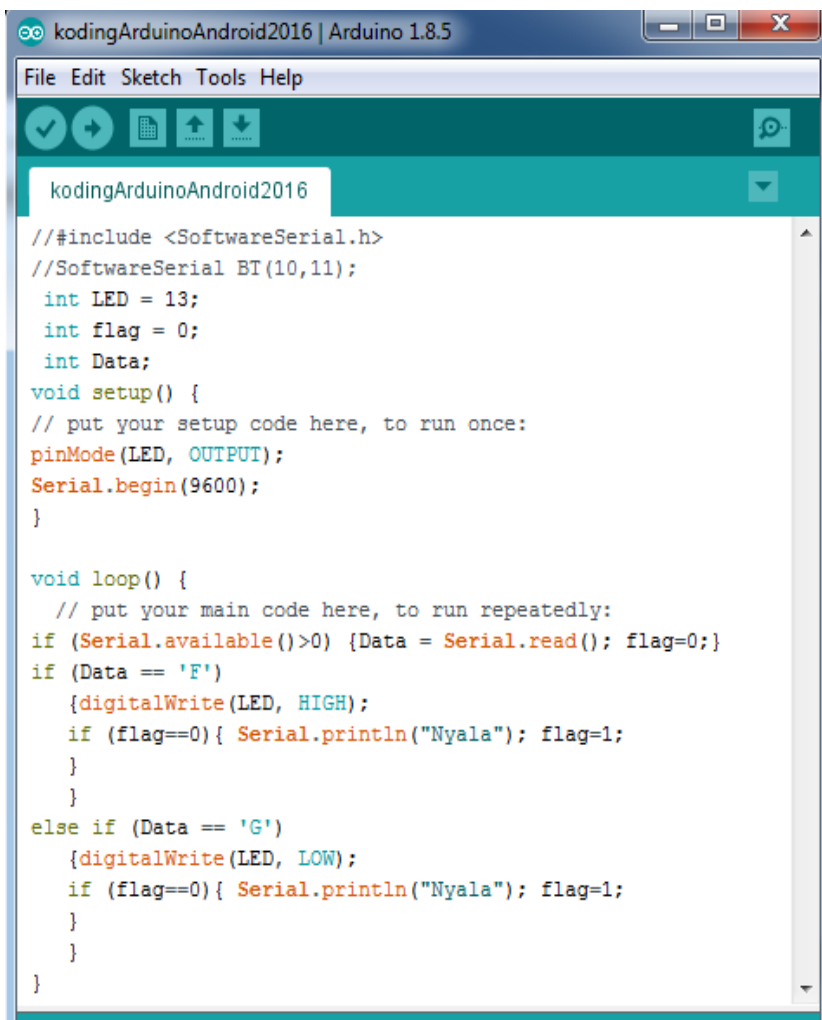


Gambar 16. MIT AI2 Companion



Gambar 17. Cara Pembacaan QR Code dengan MIT AI2 Companion

Untuk berikutnya melakukan Pemrograman di IDE Arduino. Pemrogramannya dibuat seperti gambar berikut ini.



Gambar 18. Screenshoot Pemrograman IDE Arduino untuk Bluetooth

```

//#include <SoftwareSerial.h>
//SoftwareSerial BT(10,11);
int LED = 13;
int flag = 0;
int Data;
void setup() {
// put your setup code here, to run once:
pinMode(LED, OUTPUT);
Serial.begin(9600);
}
void loop() {
// put your main code here, to run repeatedly:
if (Serial.available(>0) {Data = Serial.read(); flag=0;}
if (Data == 'F')
{digitalWrite(LED, HIGH);
if (flag==0){ Serial.println("Nyala"); flag=1;
}}
else if (Data == 'G')
{digitalWrite(LED, LOW);
if (flag==0){ Serial.println("Nyala"); flag=1;
}
}
}
}

```

Gambar 19. Pemrograman IDE Arduino untuk Bluetooth

Untuk mengintegrasikan Arduino dengan smartphone dilakukan dengan cara sebagai berikut ini :

1. Aktifkan *Bluetooth* pada *Smartphone Android*.



Gambar 20. Aktifkan *Bluetooth* pada *Smartphone Android*.

2. Buka aplikasi kontrol tekan icon icon gambar *Bluetooth* pada aplikasi *Smartphone Android*.



Gambar 21. Aplikasi di *Smartphone Android*.

3. pilih *Modul Bluetooth HC-05* pada daftar pencarian di *smartphone* anda.



Gambar 22 Pencarian koneksi *bluetooth*.

4. Pilih komunikasi HC 05 sebagaimana gambar 22. Jika Koneksi tersambung maka aplikasi siap digunakan untuk menggerakkan hardware atau mematikan lampu di hardware.

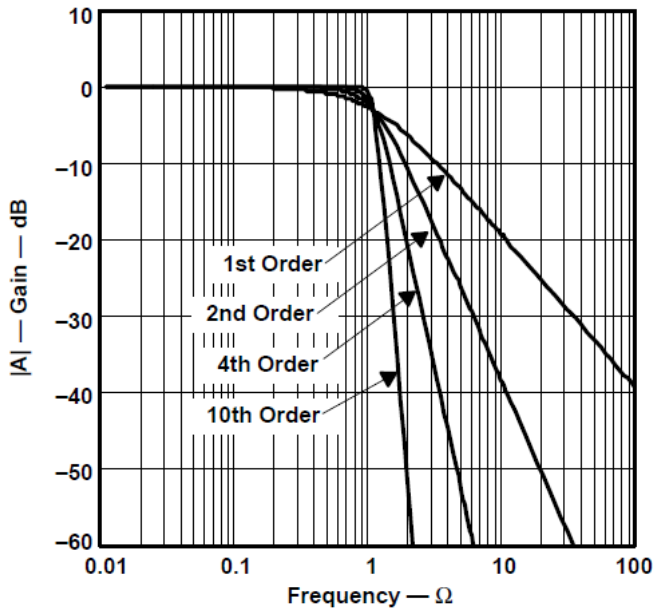
BAB X FILTER PADA INSTRUMENTASI BIOMEDIK

A. Instrumentasi *Low Pass Filter*

Instrumentasi *Low Pass Filter* merupakan suatu instrumentasi yang berfungsi untuk memfilter suatu sinyal frekuensi rendah. Grafik hasil uji Instrumentasi *low pass filter* dapat dilihat seperti gambar 4.2 dibawah ini. Grafik Instrumentasi *Low Pass Filter* orde 1 berbeda dengan grafik Instrumentasi *Low Pass Filter* orde 10, makin tinggi orde berdampak pada makin curamnya grafik dalam mendekati frekuensi *cut off* dan makin tinggi kepresisiannya. Salah satu contohnya Instrumentasi *Low Pass Filter* frekuensi *Cut Off* 450 Hz, ini maksudnya suatu instrumentasi filter sinyal yang meloloskan sinyal frekuensi dibawah 450 Hz sedangkan sinyal diatas frekuensi 450 Hz di reduksi maupun di hilangkan (Gambar 4.1). Ini sangat berbeda dengan *High Pass Filter* dengan frekuensi *cut off* 20 Hz yang memiliki makna bahwa suatu instrumentasi filter sinyal yang meloloskan sinyal frekuensi di atas 20 Hz sedangkan sinyal di bawah frekuensi 20 Hz di reduksi maupun di hilangkan.

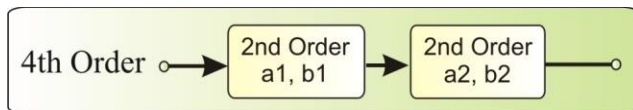


Gambar 4.1. Ilustrasi *Low pass Filter*. Sinyal Frekuensi dibawah 450 Hz di loloskan, sedangkan diatas 450 Hz dihilangkan maupun di reduksi.

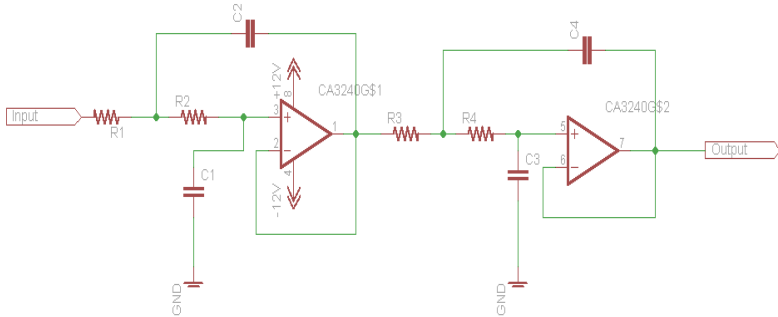


Gambar 4.2. Pola sinyal *Butterworth Low-Pass Filter* (Kugelstadt, 2008)

Rangkaian *Low Pass Filter* orde 4 untuk frekuensi *cutoff* (f_c) 500 Hz merupakan suatu rangkaian yang dibentuk dari 2 buah rangkaian *Low Pass Filter* orde 2 *Sallen-Key Topology* sebagaimana Gambar 4.3 dan Gambar 4.4 (Skematik).



Gambar 4.3. Susunan tahapan *Low Pass Filter* orde 4



Gambar 4.4. Rangkaian *Low Pass Filter* orde 4 frekuensi *cut off* 500 Hz

Sehingga diperlukan 2 buah rangkaian *Low Pass Filter* orde 2 sebanyak 2 buah untuk membuatnya. Selain itu diperlukan Nilai koefisien $a_1 = 1.8478$; $b_1 = 1.0000$; $a_2 = 0.7654$ dan $b_2 = 1.0000$, nilai-nilai tersebut merupakan koefisien *Butterworth* untuk orde 4 (lihat Tabel 4.1).

Tabel 4.1. *Koefisien Butterworth* Rangkaian *Low Pass Filter* orde 4 frekuensi *cut off* 500 Hz (Kugelstadt, 2008)

n	i	a_i	b_i	$k_i = \frac{f_{Ci}}{f_c}$	Q_i
1	1	1.0000	0.0000	1.000	—
2	1	1.4142	1.0000	1.000	0.71
3	1	1.0000	0.0000	1.000	—
	2	1.0000	1.0000	1.272	1.00
4	1	1.8478	1.0000	0.719	0.54
	2	0.7654	1.0000	1.390	1.31

Dengan menggunakan nilai kapasitor $C_1 = 46$ nF, koefisien $a_1 = 1.8478$ dan koefisien $b_1 = 1.0000$, serta nilai resistor R_1 , R_2 , dan kapasitor C_2 *Low Pass Filter* orde 2 pertama dapat dihitung

dengan Persamaan (4.1) dan (4.2). Sehingga didapat nilai resistor $R_1 = R_2 = 6260.333379 \Omega$ dan kapasitor $C_2 = 5.506148546211 \times 10^{-8} \text{ F}$. *Low Pass Filter* banyak digunakan dalam pembuatan Instrumentasi Medis, diantaranya Instrumentasi Medis *Phonocardiography*, Instrumentasi Medis *Electrocardiography*, Instrumentasi Medis *Electromyograph*.

Perlu diketahui bahwa dalam menghitung nilai R_1, R_2, R_3 , dan R_4 maka nilai koefisien *Butterworth* a_1 dan b_1 adalah untuk rangkaian *Low Pass Filter* orde 2 yg pertama, sedangkan a_2 dan b_2 adalah untuk rangkaian *Low Pass Filter* orde 2 yang ke dua. Koefisien a_2 dan b_2 digunakan untuk membantu perhitungan pada rangkaian kedua, dengan mengganti nilai R_1 menjadi R_3 dan R_2 menjadi R_4 . Sedangkan nilai C_1 dan C_3 semuanya sama sesuai persamaan (4.2). Rangkaian *Low Pass Filter* banyak digunakan dalam pembuatan instrumentasi *Electrocardiography*, Instrumentasi medis *Elektromyograph*, dan Instrumentasi Medis *Carotid Pulse*.

$$R_{1,2} = \frac{a_1 C_2 \mp \sqrt{a_1^2 C_2^2 - 4b_1 C_1 C_2}}{4\pi f_c C_1 C_2} \quad (4.1)$$

$$C_2 \geq C_1 \frac{4b_1}{a_1^2} \quad (4.2)$$

Dengan :

$a_1, b_1, a_2, b_2,$ = koefisien *Butterworth*

C_1, C_2 = Kapasitor (F)

R_1, R_2, R_3, R_4 = Resistor (Ohm)

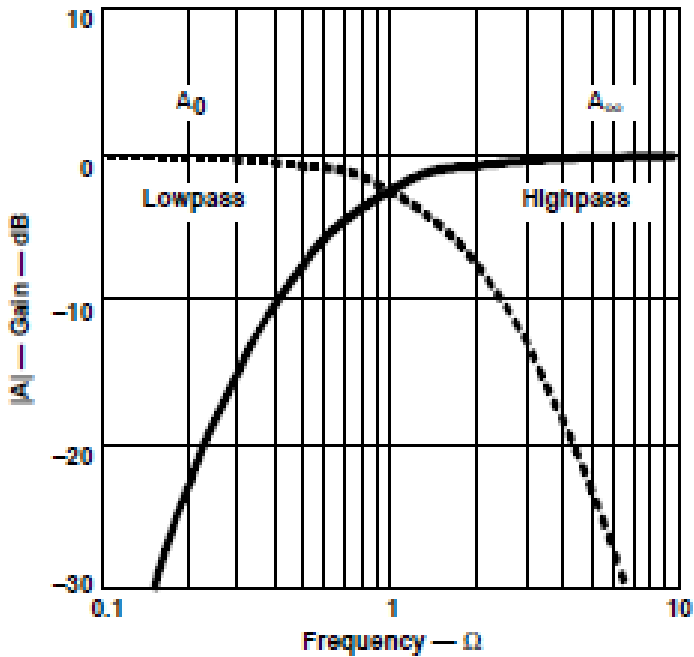
f_c = Frekuensi Cut Off (Hz)

π = koefisien (3,14)

Pada Rangkaian *Low Pass Filter* orde 4, perlu diketahui pula bahwa *Low Pass Filter* orde 2 yang kedua dihitung menggunakan Persamaan (4.1) dan (4.2) dengan mengganti nilai R_1 menjadi R_3 ; Nilai R_2 menjadi R_4 ; Kapasitor C_1 menjadi C_3 ; Kapasitor C_2 menjadi C_4 ; koefisien a_1 menjadi a_2 ; koefisien b_1 menjadi b_2 sehingga didapatkan nilai $C_3 = 46 \text{ nF}$; $R_3 = R_4 = 2593.169806 \text{ } \Omega$ dan $C_4 = 3.20908287893516 \times 10^{-7} \text{ F}$.

B. INSTRUMENTASI HIGH PASS FILTER

Instrumentasi High Pass Filter merupakan suatu instrumentasi yang berfungsi untuk memfilter suatu sinyal frekuensi tinggi. Grafik hasil uji Instrumentasi *High pass filter* dapat dilihat seperti gambar 4.5 dibawah ini.



Gambar 4.5. Pola sinyal *Butterworth High-Pass Filter* dan *Low Pass Filter*

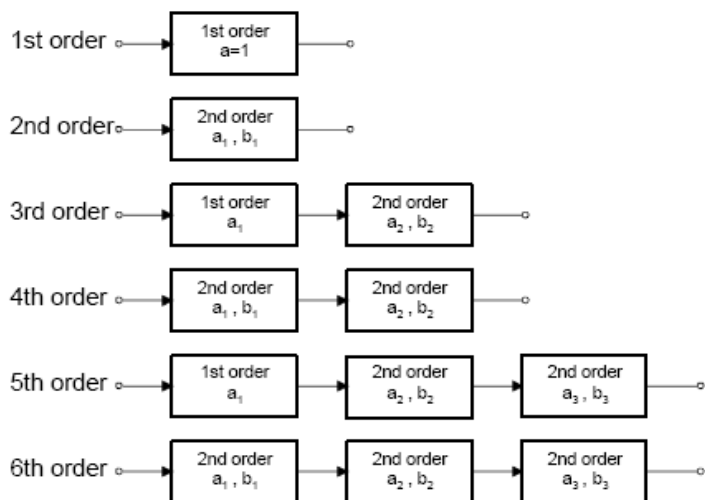
Grafik Instrumentasi *High Pass Filter* orde 1 berbeda dengan grafik Grafik Instrumentasi *High Pass Filter* orde 10, makin tinggi orde berdampak pada makin curamnya grafik dalam mendekati frekuensi *cut off* dan makin tinggi kepresisiannya. Salah satu contohnya Instrumentasi *High Pass Filter* frekuensi *Cut Off* 20 Hz, ini maksudnya suatu instrumentasi filter sinyal yang meloloskan sinyal frekuensi diatas 20 Hz sedangkan sinyal dibawah frekuensi 20 Hz di reduksi maupun di hilangkan Gambar 4.6. Ini sangat berbeda dengan *Low Pass Filter* dengan frekuensi cut off 450 Hz yang

memiliki makna bahwa suatu instrumentasi filter sinyal yang meloloskan sinyal frekuensi di bawah 450 Hz sedangkan sinyal di diatas frekuensi 450 Hz di reduksi maupun di hilangkan (gambar 4.1).

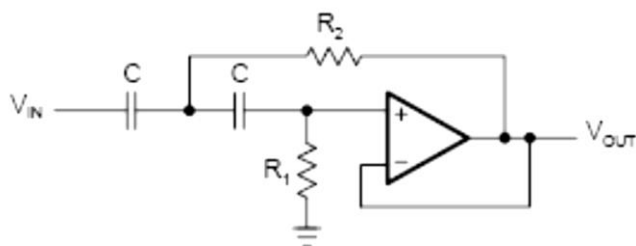


Gambar 4.6. Ilustrasi *High Pass Filter*. Sinyal Frekuensi diatas 20 Hz di loloskan, sedangkan dibawah 20 Hz dihilangkan maupun di reduksi.

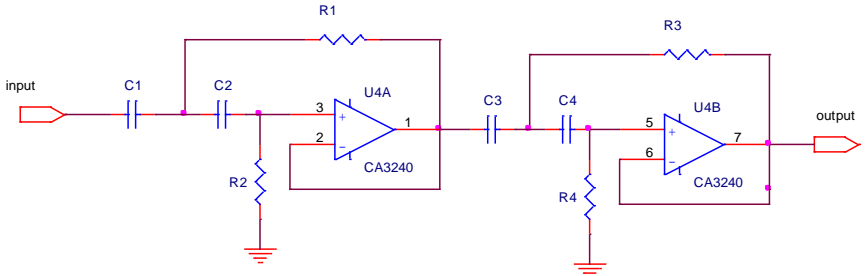
Rangkaian *High Pass Filter* orde 4 untuk frekuensi *cutoff* (f_c) 20 Hz merupakan suatu rangkaian yang dibentuk dari 2 buah rangkaian *High Pass Filter* orde 2 *Sallen-Key Topology* sebagaimana Gambar 4.7, Gambar 4.8, Gambar 4.9 (Skematik). Sehingga diperlukan 2 buah rangkaian *High Pass Filter* orde 2 sebanyak 2 buah untuk membuatnya. Selain itu diperlukan Nilai koefisien $a_1 = 1.8478$; $b_1 = 1.0000$; $a_2 = 0.7654$ dan $b_2 = 1.0000$, nilai-nilai tersebut merupakan koefisien *Butterworth* untuk orde 4 (lihat Tabel 1).



Gambar 4.7. Susunan tahapan *High Pass Filter* orde 1 sampai 6



Gambar 4.8. Rangkaian *High Pass Filter* orde 2



Gambar 4.9. Rangkaian *High Pass Filter* orde 4

Dengan menggunakan nilai kapasitor $C_1 = C_2 = 1 \mu\text{F}$, koefisien $a_1 = 1.8478$ dan koefisien $b_1 = 1.0000$, serta nilai resistor R_1 dan R_2 *High Pass Filter* orde 2 pertama dapat dihitung dengan Persamaan (4.3) dan (4.4). *High Pass Filter* banyak digunakan dalam pembuatan Instrumentasi Medis, diantaranya Instrumentasi Medis *Phonocardiography*, *Electromyograph* dan Instrumentasi Medis lainnya.

Perlu diketahui bahwa dalam menghitung nilai R_1 , R_2 , R_3 , dan R_4 maka nilai koefisien *Butterworth* a_1 dan b_1 adalah untuk rangkaian *High Pass Filter* orde 2 yg pertama, sedangkan a_2 dan b_2 adalah untuk rangkaian *High Pass Filter* orde 2 yang ke dua. Koefisien a_2 dan b_2 digunakan untuk membantu perhitungan pada rangkaian kedua, dengan mengganti nilai R_1 menjadi R_3 dan R_2 menjadi R_4 . Sedangkan nilai C_1 dan C_3 semuanya sama menggunakan kapasitor yang sama.

$$R_1 = \frac{1}{\pi f_c C a_1} \quad (4.3)$$

$$R_2 = \frac{a_1}{4\pi f_c C b_1} \quad (4.4)$$

Dengan :

a_1, b_2, a_2, b_2 = koefisien *Butterworth*

C_1, C_2 = Kapasitor (Farad)

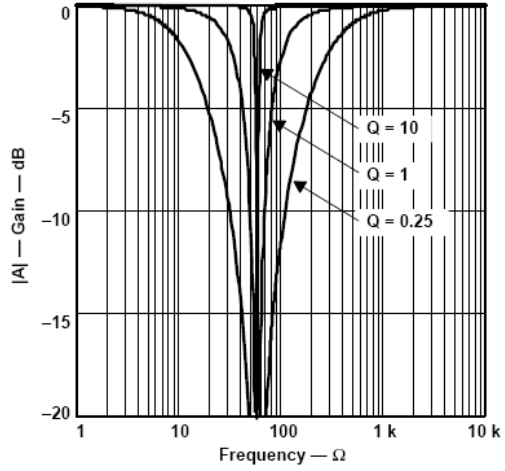
R_1, R_2, R_3, R_4 = Resistor (Ohm)

f_c = Frekuensi Cut Off (Hz)

π = koefisien (3,14)

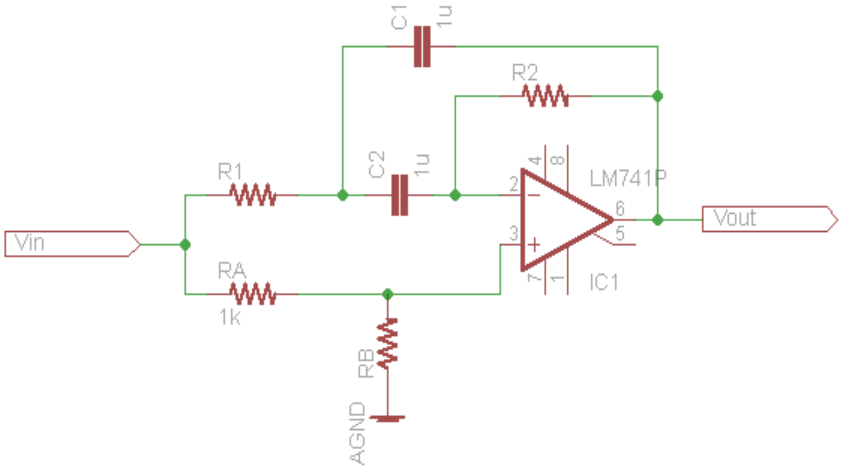
C. INSTRUMENTASI NOTCH FILTER

Rangkaian *Notch Filter* adalah suatu rangkaian yang berfungsi untuk memfilter sinyal di frekuensi tertentu, yang diakibatkan suplay tegangan berasal dari listrik PLN 220 volt. Penggunaan *Notch Filter* di Indonesia digunakan untuk memfilter sinyal frekuensi jala-jala PLN yang nilainya sebesar 50 Hz. Untuk Instrumentasi medis, sinyal frekuensi 50 Hz jala-jala PLN sangat mempengaruhi bentuk sinyal sebuah Instrumentasi medis, dan hasil penelitian yang sudah dilakukan tahun 2014 (eko, 2014) sinyal 50 Hz dapat merusak bentuk sinyal asli sebuah Instrumentasi Medis. Sehingga akan menyulitkan dokter dalam menganalisa bentuk sinyal tubuh, dan dapat menurunkan akurasi Analisa penyakit pasien karena noise sinyal 50 Hz yang merusak bentuk sinyal asli tubuh pasien. Oleh sebab itu sinyal frekuensi 50 Hz harus dihilangkan dengan bantuan Instrumentasi *Notch Filter* frekuensi *Cut off* 50 Hz.



Gambar 4.10. Pola sinyal *Notch Filter* pada Frekuensi *Cut Off* tertentu.

Rangkaian *Notch Filter* di tunjukkan seperti Gambar 4.11.



Gambar 4.11. Rangkaian *Notch Filter*

Skematik rangkaian *Notch Filter* di tunjukkan pada gambar 4.11 diatas. Pada Rangkaian ini nilai kapasitor ($C1$) yang digunakan sebesar $1 \mu\text{F}$ dan resistor (R_A) = $1 \text{ k}\Omega$ dengan langkah-langkah pembuatan disain *Notch Filter* adalah sebagai berikut:

1. menentukan *center frequency* (ω_r) dengan *frequency cut off* (f_o) = 50 Hz dan $\pi = 314$ menggunakan Persamaan (4.5)
2. menentukan Bandwidth (B) sebesar 50
3. nilai *quality factor* Q sebesar 6.28 dihitung dengan Persamaan (4.6).

4. Nilai Resistor $R_2 = 40 \text{ k}\Omega$, Nilai Resistor $R_1 = 253.55998 \text{ }\Omega$, dan nilai Resistor $R_B = 78.8768 \text{ k}\Omega$ dihitung dengan Persamaan (4.7), (4.8) dan (4.9).

$$\omega_r = 2\pi f_o \quad (4.5)$$

$$Q = \frac{\omega_r}{B} \quad (4.6)$$

$$R2 = \frac{2}{BC} \quad (4.7)$$

$$R1 = \frac{R2}{4Q^2} \quad (4.8)$$

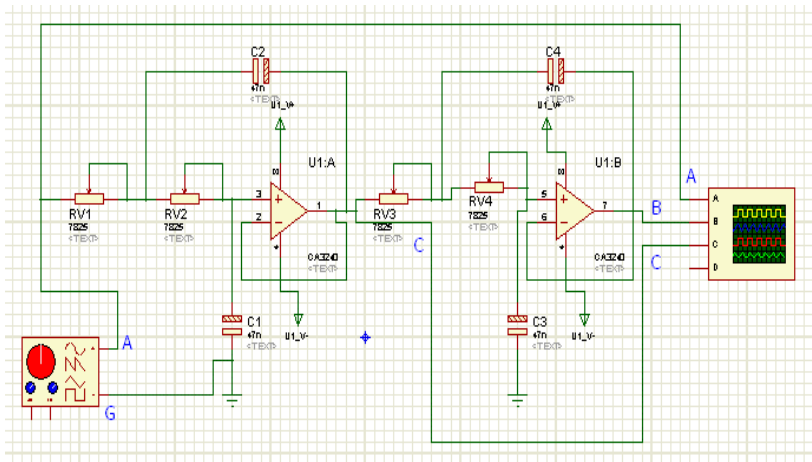
$$RB = 2Q^2 R_a \quad (4.9)$$

D. Simulasi Instrumentasi *Low Pass Filter* , *High Pass Filter* , dan *Notch Filter* di Proteus.

Simulasi Instrumentasi Filter Sinyal di Proteus akan membantu memberikan pemahaman yang lebih komprehensif akan pengertian dan pemanfaatan Instrumentasi Filter Sinyal pada Instrumentasi medis. Program yang diperlukan adalah program Proteus dengan memanfaatkan Function Generator dan Osiloskop. Berikut cara mensimulasikan instrumentasi Filter di Proteus.

D.1. *Low Pass Filter* di Proteus

Untuk mencoba simulasi, aktifkan proteus dan buat skematik *Low Pass Filter* di Proteus, pasang Osiloskop dan function generator sebagaimana gambar.

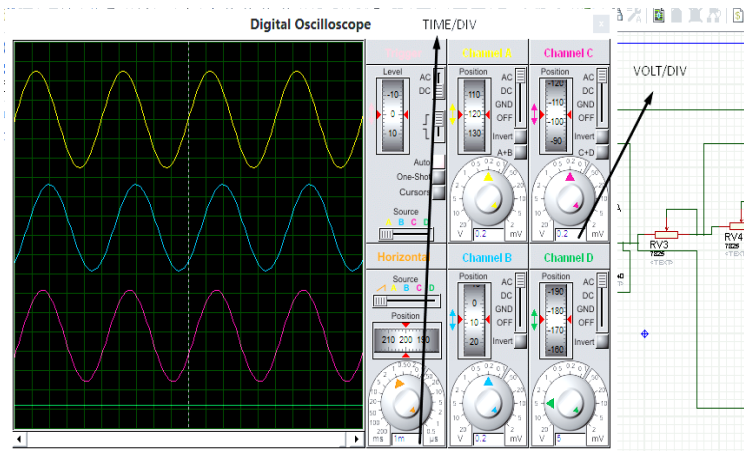


Gambar 4.12. Rangkaian *Low Pass Filter* orde 4 frekuensi *cutoff* 500 Hz di Proteus

- b) Buat skematik rangkaian *Low Pass Filter* di Proteus seperti gambar 4.12. Gunakan Op Amp tipe IC CA3240 serta resistor (R) dan kapasitor (C) *tantalum* pada rangkaian tersebut.
- c) Perhatikan Gambar 4.12. Pasang *Function Generator* untuk titik (+) pada input rangkaian (RV1) sekaligus hubungkan dengan Osiloskop pada Sinyal A, untuk titik (-) hubungkan ke *Ground* (\equiv). Hubungkan Output Instrumentasi *Low Pass Filter* orde 4 di titik B dengan Osiloskop pada sinyal B. Hubungkan Output Instrumentasi

Low Pass Filter orde 2 di titik C dengan Osiloskop pada sinyal C.

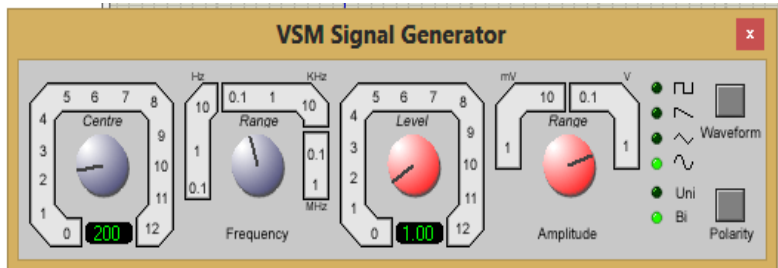
- d) Masukkan Nilai Resistor R_1 dan R_2 sebesar $R_1 = R_2 = 6260.333379 \Omega$ dan kapasitor $C_2 = 5.506148546211 \times 10^{-8} \text{ F}$ serta $C_1 = 46 \text{ nF}$.
- e) Run/Play simulasi pada Proteus, dan amati tampilan sinyalnya di osiloskop. Setting volt/div Chanel 1 dan Chanel 2 osiloskop pada posisi Volt/div sebesar 0.2 volt, dan Time/Div pada posisi 1 ms, Lihat Gambar 4.13.



Gambar 4.13. Tampilan Osiloskop

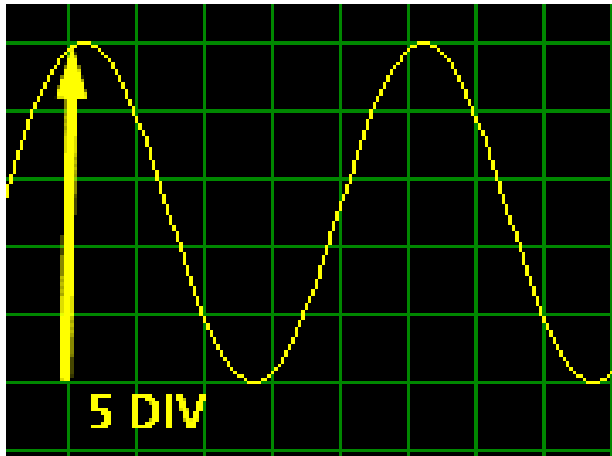
- f) Naikkan nilai frekuensi pada Function generator dari frekuensi 100 Hz, 200 Hz, 300 hz, 400 hz, 500 hz, 600 Hz, 700 Hz, 800 Hz, 1000 Hz. Print Screen hasil gambarnya dan

simpan dalam bentuk JPEG tiap-tiap frekuensi, Lihat Gambar 4.14.



Gambar 4.14. Function Generator

- g) Buat Tabel analisa tinggi Amplitud tiap frekuensi 100 Hz, 200 Hz, 300 hz, 400 hz, 500 hz, 600 Hz, 700 Hz, 800 Hz, 1000 Hz dengan cara menghitung kotak DIV nya. Seperti contoh gambar dibawah 4.15, jumlah DIV nya 5. Cari nilai tegangannya dengan cara mengalikan (Volt/Div) yang terbaca pada Osiloskop dengan jumlah kotaknya (DIV). Contohnya Volt = 0,2 (Volt/Div) \times 5 Div = 1 volt, ini sesuai dengan setting tegangan masukkan Function Generator sebesar 1 volt (Gambar 4.14).



Gambar 4.15. Jumlah DIV pada Osiloskop

Untuk memudahkan menganalisa buat table Analisa sebagaimana Tabel 4.2. berikut ini.

Tabel 4.2. Tabel pengujian Instrumentasi *Low Pass Filter*

No	Vin	DIV	Frekuensi	Volt	Gambar
	Funtion	Osiloskop			
1	1 volt		100		
2	1 volt		200		
3	1 volt		300		
4	1 volt		400		
5	1 volt		500		
6	1 volt		600		
7	1 volt		700		
8	1 volt		800		
9	1 volt		900		
10	1 volt		1000		

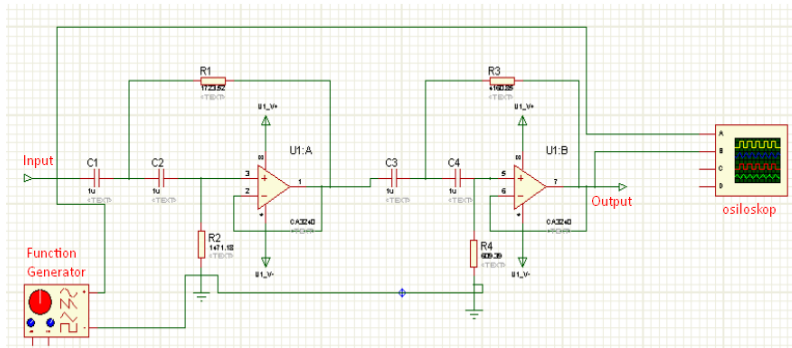
- h) Analisa pengaruh frekuensi terhadap nilai Amplitudo (DIV) nya, perhatikan perbedaannya pada tampilan Gambar di Osiloskop ketika frekuensi di setting 200 hz, **500 Hz** dan 800 Hz. Hasil Analisa harus menunjukkan bahwa pada frekuensi ≥ 500 Hz akan terjadi penurunan amplitude sinyal yag signifikan.

Latihan Soal :

1. Hitung R_1 dan R_2 serta R_3 dan R_4 untuk frekuensi *Cut Off* 100 Hz. Jika nilai C_1 yang digunakan sebesar 47 nF. (Rangkain *Low Pass Filter* orde 4).
2. Hitung nilai tegangan Output yang ditampilkan Osiloskop di software Proteus pada saat frekuensi sinyal 50 Hz, 100 Hz, dan 150 Hz soal Nomor 1, dengan cara mengalikan (Volt/Div) yang terbaca pada Osiloskop dengan jumlah kotaknya (DIV).
3. Buat Grafik Fungsi antara nilai frekuensi (Sumbu X) dengan nilai tegangan (sumbu Y) berdasarkan data yang sudah didapatkan pada Tabel 2.

D.2. High Pass Filter di Proteus

Untuk mencoba simulasi, aktifkan proteus dan buat skematik *High Pass Filter* di Proteus, pasang Osiloskop dan function generator sebagaimana gambar.



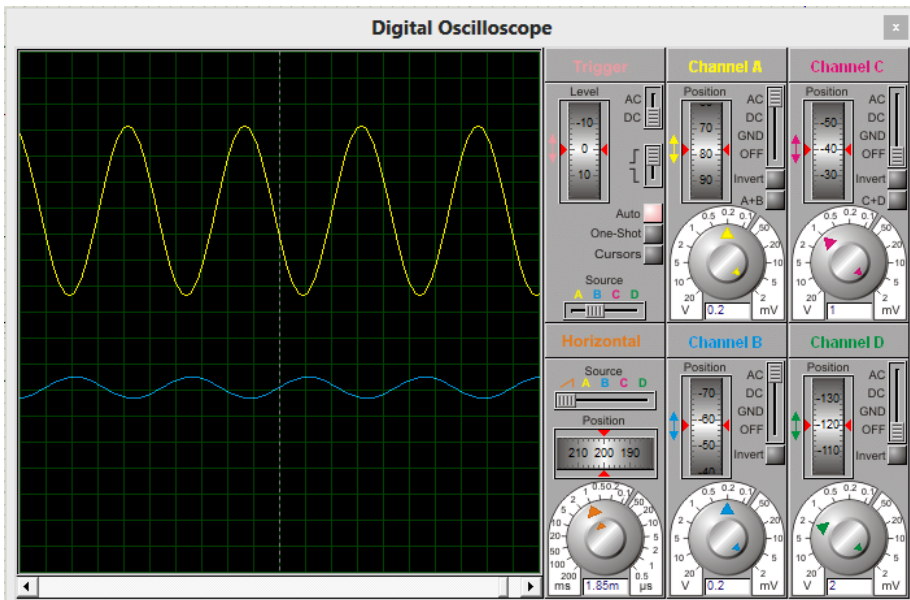
Gambar 4.16. Rangkain *High Pass Filter* orde 4 frekuensi *cutoff* 20 Hz di Proteus

- Buat skematik rangkaian *High Pass Filter* di Proteus seperti gambar 4.16. Gunakan IC CA3240 serta resistor dan kapasitor Tantalum pada rangkaian tersebut.
- Pasang Function Generator pada **input** rangkaian untuk titik (+) dan untuk titik (-) hubungkan ke Ground.
- Untuk membantu melihat tampilan sinyal Input dan output maka Hubungkan titik A Osiloskop pada input rangkaian (C1) dan titik B Osiloskop pada Output rangkaian (OP AMP di titik 7) (Lihat Gambar 4.16).

- d) Masukkan Nilai Resistor R_1 , R_2 , R_3 , R_4 sebesar $R_1 = 1723.52 \Omega$, $R_2 = 1471.18 \Omega$, $R_3 = 4160.85 \Omega$, $R_4 = 609.39 \Omega$ dan kapasitor $C_2 = 1 \mu\text{F}$ serta $C_2 = 1 \mu\text{F}$.
- e) Run/Play simulasi Proteus, dan amati tampilan sinyalnya di osiloskop. Setting volt/div Chanel 1 dan Chanel 2 pada osiloskop dengan disetting volt/div 0.2 volt sebagaimana

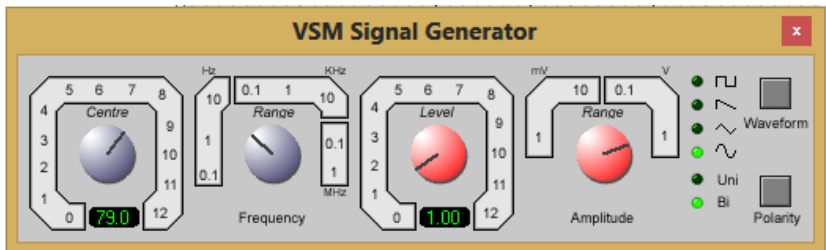


icon ini . Setting tegangan masukkan Function Generator sebesar 1 volt (Gambar 4.18).



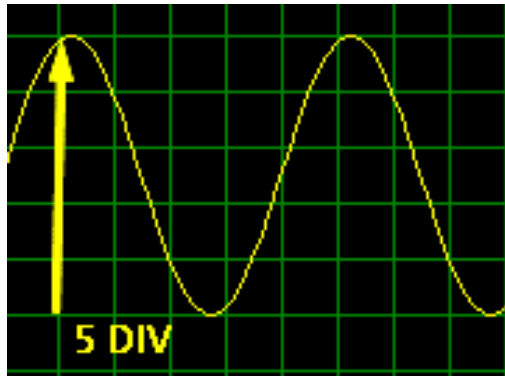
Gambar 4.17. Tampilan Osiloskop Proteus

- f) Naikkan nilai frekuensi pada Function generator (Lihat Gambar 4.18) dari frekuensi 10 Hz, 20 Hz, 30 Hz, 40 Hz, 50 Hz, 60 Hz, 70 Hz, 80 Hz, 90 Hz, 100 Hz dan 1000 Hz. Print Screen hasil gambarnya dan simpan dalam bentuk JPEG tiap-tiap frekuensi.



Gambar 4.18. Function Generator

- g) Buat Tabel analisa tinggi Amplitud tiap frekuensi 10 Hz, 20 Hz, 30 Hz, 40 Hz, 50 Hz, 60 Hz, 70 Hz, 80 Hz, 90 Hz, 100 Hz dan 1000 Hz dengan cara menghitung kotak DIV nya. Seperti contoh gambar dibawah ini, jumlah DIV nya 5. Cari nilai tegangannya dengan cara mengalikan (Volt/Div) yang terbaca pada Osiloskop dengan jumlah kotaknya (DIV). Contohnya $\text{Volt} = 0,2 (\text{Volt/Div}) \times 5 \text{ Div} = 1 \text{ volt}$, ini sesuai dengan setting tegangan masukkan Function Generator sebesar 1 volt (Gambar 4.18).



Gambar 4.19. Jumlah DIV pada Osiloskop

Tabel 4.3 Analisa data

No	Vin	DIV	Frekuensi	Volt	Gambar
	Funtion	Osiloskop			
1	1 volt		10		
2	1 volt		20		
3	1 volt		30		
4	1 volt		40		
5	1 volt		50		
6	1 volt		60		
7	1 volt		70		
8	1 volt		80		
9	1 volt		90		
10	1 volt		100		
11	1 volt		1000		

- h) Analisa pengaruh frekuensi terhadap nilai Amplitudo (DIV) nya, perhatikan perbedaannya pada tampilan Gambar di

Osiloskop ketika frekuensi di setting 10 Hz, 20 Hz dan 50 Hz.

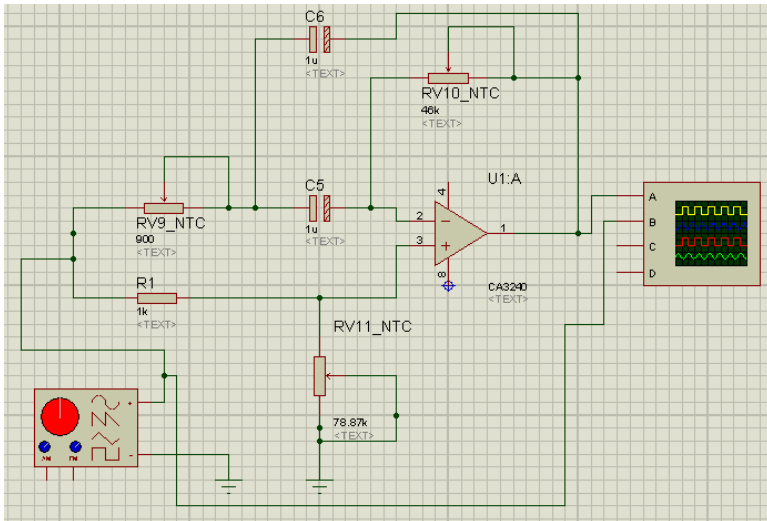
A. Tugas

1. Hitung R1, R2, R3, dan R4 Rangkaian High Pass Filter untuk frekuensi Cutt Off 20 Hz. Jika nilai C1 yang digunakan sebesar 47 nF. (Rangkain High Pass Filter orde 4).
2. Hitung nilai tegangan Output yang ditampilkan Osiloskop dengan cara mengalikan (Volt/Div) yang terbaca pada Osiloskop dengan tinggi amplitude maksimal jumlah kotaknya (DIV) pada sebagaimana panduan point (g) diatas.
3. Buat Grafik Fungsi antara nilai frekuensi (Sumbu X) dengan nilai tegangan (sumbu Y) berdasarkan data yang sudah didapatkan pada Tabel 4.3.

D.3. Notch Filter di Proteus

Untuk mencoba simulasi, aktifkan proteus dan buat skematik *Notch Filter* di Proteus, pasang Osiloskop dan function generator sebagaimana gambar (4.20).

Cara Kerja :

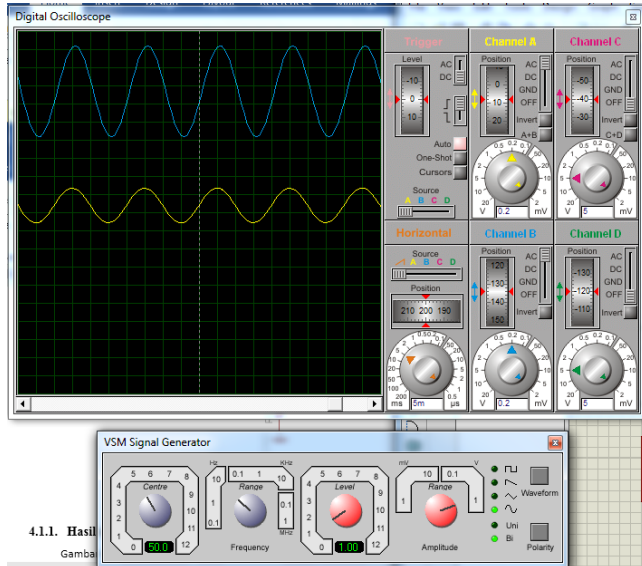


Gambar 4.20. Rangkaian *Notch Filter* frekuensi *cutoff* 50 Hz di Proteus

- A. Buat skematik rangkaian *Notch Filter* di Proteus seperti gambar 4.20. Gunakan IC CA3240 serta resistor dan kapasitor Tantalum pada rangkaian tersebut.
- B. Pasang Function Generator pada **input** rangkaian untuk titik (+) dan untuk titik (-) hubungkan ke Ground.

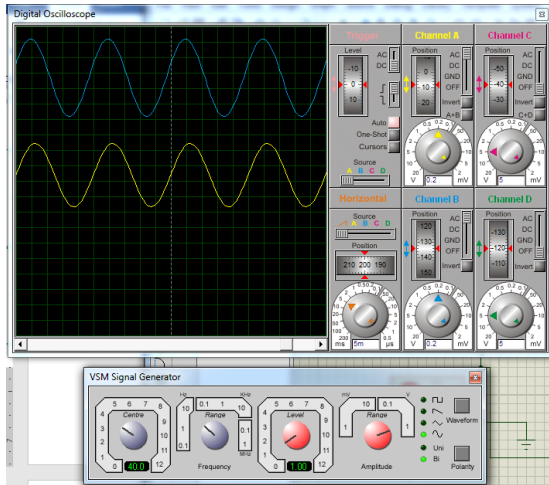
- C. Untuk membantu melihat tampilan sinyal Input dan output maka Hubungkan titik A Osiloskop pada input rangkaian (C1) dan titik B Osiloskop pada Output rangkaian (Lihat Gambar 4.20).
- D. Masukkan Nilai resistor pada rangkaian untuk $R_{V9} = 900\Omega$, $R_{V10} = 46k\Omega$, $R_{V11} = 78.87k\Omega$, $R_1 = 1k\Omega$, $C_5 = 1\mu\Omega$, $C_6 = 1\mu\Omega$.
- E. Setting nilai frekuensi pada Function generator dari 10 Hz, 20 Hz, 30 Hz, 40 Hz, 50 Hz, 60 Hz, 70 Hz, 80 Hz, 90 Hz, 100 Hz. Analisa perbedaan Amplitude sinyal Input (sinyal warna Biru) dengan sinyal output (sinyal warna kuning) ketika berada di frekuensi tersebut. Amati lebih dalam ketika sinyal berada di frekuensi 50 Hz, dan diatas maupun di bawah frekuensi 50 Hz. Akan dihasilkan amplitude yang kecil jika berada frekuensi 50 Hz, dan akan dihasilkan Frekuensi yang besar ketika berada di frekuensi kurang dari atau lebih dari 50 Hz. (lihat Gambar 4.21).
- F. Gambar 4.21 menunjukkan Input Function generator berapa pada frekuensi 50 Hz yang di tunjukkan oleh sinyal warna biru (Amplitudi 5 div), dan output sinyal function generator warna kuning (2.5 div). Hal tersebut berbeda tinggi Amplitudo sinyal Outputnya ketika Input Function

generator berapa pada frekuensi 40 Hz (Gambar 4.22),
 Input Function generator berapa pada frekuensi 60 Hz
 (Gambar 4.23).

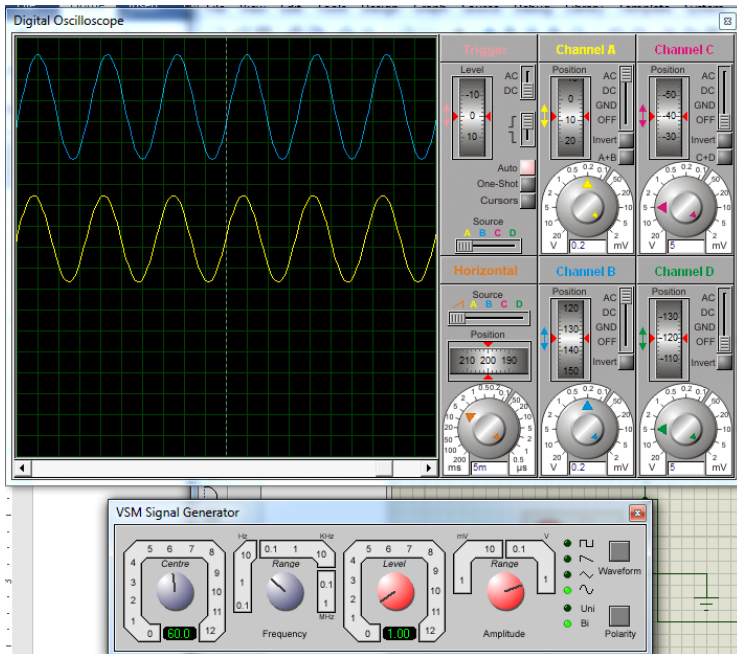


4.1.1. Hasil
 Gamba

Gambar 4.21. Tampilan Osiloskop dan Function generator
Notch Filter frekuensi *Cut Off* 50 Hz di Proteus



Gambar 4.22. Tampilan Osiloskop dan Function generator
Notch Filter frekuensi *Cut Off* 40 Hz di Proteus



Gambar 4.23. Tampilan Osiloskop dan Function generator
Notch Filter frekuensi *Cut Off* 50 Hz di Proteus

A. Tugas

1. Hitung persamaan komponen Instrumentasi *Notch Filter* frekuensi *cut off* (f_c) 50 Hz dengan *Bandwidth* B sebesar 30. Pada Instrumentasi ini, nilai C1 yang digunakan sebesar 1 μF dan $R_A = 1\text{k}\Omega$.

Carilah nilai :

(a). *Quality factor* Q ?

(b) Nilai R_2 ?

(c). R_1 ? dan,

(d). R_B ?

2. Hitung persamaan komponen Instrumentasi *Notch Filter* frekuensi *cut off* (f_c) 50 Hz dengan *Bandwidth* B sebesar 52. Pada Instrumentasi ini, nilai C_1 yang digunakan sebesar $1 \mu\text{F}$ dan $R_A = 1\text{k}\Omega$.

Carilah nilai :

(a). *Quality factor* Q ?

(b) Nilai R_2 ?

(c). R_1 ? dan,

(d). R_B ?

3. Buat Skematik Rangkaian *Notch Filter* soal nomor 2 di Proteus. Buat Layout yang siap dicetak di PCB, dan Sertakan tampilan 3D dengan Proteus.

Daftar Pustaka

- [1]. Dinata, Yuwono Marta. 2015. "Arduino Itu Mudah". Penderbit PT Elex Media Komputindo.
- [2]. Andrianto, Heri. 2016. "Arduino Belajar Cepat dan Pemrograman". Penerbit Informatika Bandung.
- [3]. Darmawan, Aan. 2016. "Arduino Belajar Cepat dan Pemrograman". Penerbit Informatika Bandung.
- [4]. Wardhana, Lingga. 2006. "Belajar Sendiri Mikrokontroler AVR Seri AT Mega32 Simulasi, Hardware, Aplikasi". Penerbit Andi Offset Yogyakarta.
- [5]. Kadir, Abdul. 2013. " Panduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya Menggunakan Arduino". Penerbit Andi Yogyakarta.
- [6]. Mubarak T, Sumaryo S. Abdurohman M. 2008. "Seminar Nasional Aplikasi Teknologi Informasi(SNATI 2008)". ISSN:1907-5022. Yogyakarta.
- [7]. Ahmadi, Aziz. 2009. Kendali Penerangan Rumah jarak jauh menggunakan Short Message Service (SMS). Jurnal Fisika dan aplikasinya (Neutrino) Vol.1, No.2 2009.ISSN 1979-6374, EISSN 2460-5999, Department of Physics, Faculty of Sains and Technlogy, Maulana Malik Ibrahim State Islamic University of Malang.
- [8]. Kilian, Christopher T.1996. " Modern Control Technology".
- [9]. O. W. Pratama and H. K. S. Safitri, "Sistem Kendali Gerak Robot Menggunakan PC Berbasis Bluetooth," *J. Elektron. Otomasi Ind.*, vol. 1, pp. 51–57, 2014

- [10]. Webster, J.G., 1978, Medical Instrumentation Application and Design, Houghton Mifflin Co.
- [11]. Tompkins, W.J., and J.G. Webster, 1981, Design of Microcomputer Based Medical Instrumentation, New Jersey: PHI.
- [12]. Aston, R., 1991, Principles of Biomedical Instrumentation and Measurement, New York: Merrill.
- [13]. Tompkins, J., 1974, Biomedical Digital Signal Processing, Prentice Hall Inc. Network Design, Tata McGraw
- [14]. Hill.
- [15]. Barbara L. Christie, 2009, Introduction To Biomedical Instrumentation, Cambridge university
- [16]. Proceeding The 6th- Electrical Power, Electronics, Communications, Controls and Informatics International Seminar 2012, 30-31 Mei, Universitas Brawijaya, Malang, Indonesia
- [17]. YouTube, Browsing Artikel terkait.
- [18]. Thomas Kugelstadt, 2008. "Chapter 16 Active Filter Design Techniques, Literature Number SLOA088," Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
- [19]. E.A. Suprayitno, R.Hendra, A. Arifin, "Analisa Sinyal Electrocardiography dan Phonocardiography Secara Simultan Menggunakan Continuous Wavelet Transform," Proceedings of the The 6th – EECCIS Seminar 2012 at Brawijaya University, Malang, pp. B18-1 - B18-6, 2012.

- [20]. Ranagayyan, "Biomedical Signal Analysis A Case-Study Approach," IEEE Press, John Wiley & Sons, INC, Canada, 2002

BIODATA PENULIS



Eko Agus Suprayitno, S.Si.,MT, lahir di Pasuruan, 13 Agustus 1987, anak pertama dari 2 bersaudara dari pasangan Bapak Karijono dan Ibu Sumiani. Penulis tercatat sebagai dosen tetap di Universitas Muhammadiyah Sidoarjo mulai tahun 2013, pada Program Studi Teknik Elektro. Latar belakang

Pendidikan penulis dimulai pada jenjang Strata-1 tahun 2005 di Universitas Airlangga Surabaya Jurusan Fisika. Jenjang pendidikan Master angkatan tahun 2010 diraih penulis di Institut Teknologi Sepuluh Nopember Surabaya Jurusan Teknik Elektro dengan konsentrasi Elektronika Medis dan lulus pada tahun 2012. Penulis juga aktif dalam melakukan penelitian, utamanya berkaitan dengan Elektronika Medis dan Robotika, serta puluhan hasil penelitiannya sudah dipublikasikan baik di Proceeding Seminar dan Jurnal secara Nasional maupun Internasional yang berhubungan dengan Instrumentasi Medis maupun otomasi mesin. Penulis Juga sudah mengajukan HAKI atas salah satu karya ciptaannya dan Penulis Juga merupakan Reviewer Nasional Ristekdikti 2018 untuk PKM 5 Bidang serta PKM AI & GT. Penulis dapat dihubungi di alamat email : eko.agus@umsida.ac.id



Rohman Dijaya. Skom., M.Kom, lahir di Gresik, 23 April 1990, Penulis tercatat sebagai dosen tetap di Universitas Muhammadiyah Sidoarjo mulai tahun 2016, pada Program Studi Teknik Informatika. Latar belakang Pendidikan penulis dimulai pada jenjang Strata-1

tahun 2009 di Universitas Muhammadiyah Gresik Jurusan Teknik Informatika. Jenjang pendidikan Master angkatan tahun 2014 diraih penulis di Institut Teknologi Sepuluh Nopember Surabaya Jurusan Teknik Informatika dengan konsentrasi Komputasi Cerdas dan Visual dan lulus pada tahun 2016. Penulis juga aktif dalam melakukan penelitian, utamanya berkaitan dengan Komputasi Cerdas dan Visual, sudah banyak hasil penelitiannya yang sudah dipublikasikan baik di Proceeding Seminar dan Jurnal secara Nasional maupun Internasional yang berhubungan dengan Komputasi Cerdas dan Visual. Penulis Juga sudah mengajukan HAKI atas salah satu karya ciptaannya. Penulis dapat dihubungi di alamat email : rohman.dijaya@umsida.ac.id

Referensi

- [1]. D. Juliasari Noni, “Aplikasi Monitoring Suhu dan Kelembaban Ruang Server PT Hero Supermarket Tbk.”.
- [2]. B. S. Putra *et al.*, “Desain Dan Implementasi Sistem Monitoring Dan Manajemen Baterai Mobil Listrik Design and Implementation of Electric Car Battery,” vol. 2, no. 2, pp. 1909–1916, 2015.
- [3]. B. Yuwono, *Optimalisasi Panel Sel Surya dengan menggunakan sistem pelacak berbasis mikrokontroler AT89C51*. 2005.
- [4]. D. T. B. Sihombing and I. S. T. K. Msi, “PERENCANAAN SISTEM PENERANGAN JALAN UMUM DAN TAMAN DI AREAL KAMPUS USU DENGAN MENGGUNAKAN TEKNOLOGI TENAGA SURYA (APLIKASI DI AREAL PENDOPO DAN LAPANGAN PARKIR). Departemen Teknik Elektro Fakultas Teknik Universitas Sumatera Utara,” *Singuda Ensikom*, vol. 3, no. 3, pp. 118–123, 2013.
- [5]. D. Y. STEVANUS, *PELUANG PEMANFAATAN LAMPU LED SEBAGAI LAMPU PENERANGAN YANG HEMAT ENERGI*. 2011.
- [6]. F. Mubarok, *PENGEMBANGAN MEDIA PEMBELAJARAN BERBASIS MOBILE APPLICATION MENGGUNAKAN APP INVENTOR PADA MATA PELAJARAN MEKANIKA TEKNIK UNTUK SISWA KELAS X STUDI KEAHLIAN TGB SMK NEGERI 3 YOGYAKARTA*. 2015.
- [7]. H. Susanto, R. Pramana, S. T. Mt, M. Mujahidin, and S. T. Mt, “Perancangan Sistem Telemetry Wireless Untuk

- Mengukur Suhu Dan Kelembaban Berbasis Arduino Uno R3 Atmega328P Dan Xbee Pro,” *Jur. Tek. Elektro, Fak. Tek. Univ. Marit. Raja Ali Haji*, 2013.
- [8]. O. W. Pratama and H. K. S. Safitri, “Sistem Kendali Gerak Robot Menggunakan PC Berbasis Bluetooth,” *J. Elektron. Otomasi Ind.*, vol. 1, pp. 51–57, 2014.
- [9]. R. H. Muhammad, R. S. Adi, and A. Kondisi, “Rancang Bangun Sistem Pengamanan Mobil Menggunakan ID Card Dengan Metode Radio Frequency Identification,” vol. 1, no. 1, pp. 39–44, 2017.
- [10]. R. Raban, P. Surya, and B. Converter, “DESAIN DAN IMPLEMENTASI CHARGER BATERAI PORTABLE MENGGUNAKAN MODUL IC XL6009E1 SEBAGAI BOOST CONVERTER DENGAN MEMANFAATKAN TENAGA SURYA.”
- [11]. T. A. Djoko Adi Widodo , Suryono, *PEMBERDAYAAN ENERGI MATAHARI SEBAGAI ENERGI LISTRIK LAMPU PENGATUR LALU LINTAS*, vol. 2, no. 2. 2010.