

MIKROKONTROLER

Apa itu Mikrokontroler?

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Didalamnya terkandung sebuah inti prosessor, memori (sejumlah kecil RAM, memori program, atau keduanya) dan perlengkapan input output. **Mikrokontroler** adalah salah satu dari bagian dasar dari sebuah sistem komputer. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan komputer mainframe, Mikrokontroler dibangun dari elemen-elemen dasar yang sama. Secara sederhana, komputer akan menghasilkan output spesifik berdasarkan inputan yang diterima dalam program yang dikerjakan.

Seperti umumnya komputer, Mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan kepadanya. Artinya, bagian terpenting dan utama dari suatu sistem terkomputerisasi adalah program itu sendiri yang dibuat oleh seorang programmer. Program ini menginstruksikan komputer untuk melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks yang diinginkan oleh programmer.

Dengan kata lain, Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja Mikrokontroler sebenarnya membaca dan menulis data. Sebagai contoh, bayangkan diri anda saat mulai belajar membaca dan menulis ketika anda sudah bisa melakukan hal itu anda bisa membaca tulisan apapun dan andapun bisa menulis hal-hal sebaliknya. Begitu pula jika anda sudah mahir membaca dan menulis data anda dapat membuat program untuk membuat suatu sistem pengaturan otomatis menggunakan Mikrokontroler sesuai keinginan anda.

Mikrokontroler merupakan komputer di dalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan epektifitas biaya. Secara harfiahnya bisa disebut "**pengendali kecil**" dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat dikendalikan oleh Mikrokontroler ini.

Pemanfaatan Mikrokontroler

Mikrokontroler ada pada perangkat elektronik disekeliling kita misalnya handphone, MP3 player, DVD, televisi, AC, dll. Mikrokontroler juga dipakai untuk keperluan mengendalikan robot. Baik robot mainan, maupun robot industri. Mikrokontroler juga digunakan dalam produk dan alat yang dikendalikan secara otomatis, seperti, sistem control mesin, remote control, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat input output terpisah, kehadiran Mikrokontroler membuat control elektrik untuk berbagai proses lebih ekonomis. Dengan penggunaan mikrokontroler ini, maka :

- Sistem elektronik akan menjadi lebih ringkas
- Rancangan bangunan sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah di modifikasi
- Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak

Namun demikian tidak sepenuhnya mikrokontroler bisa mereduksi komponen IC TTL dan CMOS yang sering kali masih diperlukan untuk aplikasi kecepatan tinggi atau sekedar untuk menambah jumlah saluran masukan dan keluaran (I/O). dengan kata lain, mikrokontroler adalah versi mini dari mikro dari sebuah komputer karena mikrokontroler sudah mengandung beberapa peripheral yang langsung bisa dimanfaatkan, misalnya port parallel, port serial, komprator, konversi digital ke analog (DAC), konversi analog kedigital dan sebagainya hanya menggunakan sistem minimum yang tidak rumit atau kompleks.

Agar mikrokontroler dapat berfungsi, mikrokontroler tersebut memerlukan komponen eksternal yang kemudian disebut dengan sistem minimum. Untuk membuat sistem minimum paling tidak dibutuhkan sistem clock dan reset, walaupun pada beberapa mikrokontroler sudah menyediakan sistem clock internal, sehingga tanpa rangkaian eksternal pun mikrokontroler sudah beroperasi.

Untuk merancang sebuah sistem berbasis mikrokontroler, kita memerlukan perangkat keras dan perangkat lunak yaitu :

1. Sistem minimum mikrokontroler
2. Software pemrograman dan aplikasi kompilator, serta downloader

Yang dimaksud dengan sistem minimum adalah sebuah rangkaian mikrokontroler yang sudah dapat digunakan untuk menjalankan sebuah aplikasi. Sebuah IC mikrokontroler

tidak akan berarti bila hanya berdiri sendiri. Pada dasarnya sebuah sistem minimum mikrokontroler AVR memiliki prinsip yang sama, yang terdiri dari 4 bagian, yaitu :

1. Prosesor, yaitu mikrokontroler itu sendiri.
2. Rangkaian reset agar mikrokontroler dapat menjalankan program mulai dari awal.
3. Rangkaian clock, yang digunakan untuk memberikan detak pada CPU.
4. Rangkaian catu daya, yang digunakan untuk memberi sumber daya.

Pada mikrokontroler jenis-jenis tertentu (AVR misalnya), poin no 2 , 3 sudah tersedia didalam mikrokontroler tersebut dengan frekuensi yang sudah si setting dari vendornya (misal; 1 MHz, 2 MHz, 4MHz, 8 MHz), sehingga pengguna tidak memerlukan rangkaian tambahan. Namun bila ingin merancang sistem dengan spesifikasi tertentu (misal ingin komunikasi dengan PC atau handphone) pengguna harus menggunakan rangkaian clock yang sesuai dengan karakteristik PC atau HP tersebut, biasanya menggunakan kristal 11,0592 MHz, untuk menghasilkan komunikasi yang sesuai dengan baud rate PC atau HP tersebut.

MIKROKONTROLER ARDUINO

Apa Itu Arduino

Arduino adalah kit elektronik atau papan rangkaian elektronik open source yang didalamnya terdapat komponen utama, yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan ATMEL. Mikrokontroler ini sendiri adalah chip atau IC (integrated circuit) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses input tersebut dan kemudian menghasilkan output sesuai dengan yang diinginkan. Jadi mikrokontroler bertugas sebagai 'otak' yang mengendalikan input, proses dan output sebuah rangkaian elektronik.

Secara umum, arduino terdiri dari 2 bagian yaitu :

1. Hardware berupa papan input/output (I/O) yang open source. atau board arduino
2. Software arduino yang open source, meliputi software arduino IDE untuk menulis program dan driver untuk koneksi dengan komputer.

Sejarah Singkat Arduino

Pembuatan arduino dimulai pada tahun 2005, dimana sebuah situs perusahaan komputer Olivetti di Ivrea Italia, membuat perangkat untuk mengendalikan proyek desain interaksi siswa supaya lebih mudah dibandingkan dengan sistem yang ada pada saat itu. Dilanjutkan pada bulan Mei 2011, dimana sudah lebih dari 300.000 unit Arduino terjual.

Pendiri dari Arduino itu sendiri adalah **Massimo Banzi** dan **David Caurtielles** sebagai founder. Awalnya mereka memberi nama proyek itu dengan sebutan **Arduin dari Ivrea** tetapi seiring dengan perkembangan zaman, nama proyek itu diubah menjadi Arduino yang berarti "**Teman yang Kuat**" atau dalam versi bahasa Inggrisnya dengan sebutan "**Hardwin**".

Proyek pengkabelan diciptakan oleh seorang seniman sekaligus programmer asal Kolombia bernama **Hernando Barragan**. Pengkabelan ini adalah proyek tesis Hernando pada desain interaksi Institute Ivrea. Hal tersebut dimaksudkan untuk menjadi versi elektronik pengolahan yang digunakan di lingkungan pemrograman dan mengambil pola sintaks Processing. Dengan berkembangnya teknologi, Arduino menjadi sangat populer dikalangan mahasiswa dan pelajar saat ini. Mereka mengembangkan Arduino dengan bootloader dan software dan user Friendly sehingga menghasilkan sebuah board

mikrokontroler yang bersifat open source yang bisa dipelajari dan dikembangkan oleh mahasiswa , pelajar, profesional, pemula, dan penggemar elektronika maupun robotic diseluruh dunia. IDE (*Integrated Development Environment*) diciptakan oleh **Casey Reas** dan **Ben Fry**, berapa programmer yang lain juga terlihat seperti Tom Igoe , Gianluca Martino , David Mellis , dan Nicholas Zambett.

Kelebihan Arduino

Tentu saja ada banyak mikrokontroler maupun platform mikrokontroler tersedia, misalnya saja Basic Stamp-nya Parallax, BX-24-nya Netmedia, Phidget, MIT's Handyboard, dan lain sebagainya. Semua alat tersebut bertujuan untuk menyederhanakan berbagai macam kerumitan maupun detail rumit pada pemrograman mikrokontroler sehingga menjadi paket mudah digunakan (easy-to-use) Arduino juga menyederhanakan proses bekerja dengan mikrokontroler, sekaligus menawarkan berbagai macam kelebihan antara lain :

- Murah. Papan (perangkat keras) Arduino biasanya dijual relative murah
- Sederhana dan Mudah Programnya. Perlu diketahui bahwa lingkungan pemrogram di arduino mudah digunakan untuk pemula , dan cukup fleksibel bagi mereka yang sudah tingkat lanjut. Untuk guru, dosen, Arduino berbasis pada lingkungan pemrogram Processing, sehingga jika mahasiswa atau murid-murid terbiasa menggunakan Processing tentu saja akan mudah menggunakan Arduino. Bahkan didalam dos/kotak Arduino terdapat tulisan bahwa arduino diperuntukkan bagi seniman, desainer, penghobi, dan siapa saja. Sungguh membesrkan hati dan membangkitkan semangat bahwa penggunaanya tidak harus teknisi berpengalaman atau ilmuan ber otak jenius.
- Perangkat Lunaknya Open Source. Perangkat lunak Arduino IDE dipublikasikan sebagai open source, tersedia bagi para pemrogram berpengalaman untuk pengembangan lebih lanjut bahasanya bisa dikembangkan lebih lanjut melalui pustaka-pustaka C++ yang berbasis pada bahasa C untuk AVR.
- Perangkat Kerasnya Open Source. Perangkat keras Arduino berbasis mikrokontroler ATMEGA8, ATMEGA168, ATMEGA328 dan ATMEGA1280. Dengan demikian , siapa saja bisa membuatnya (dan kemudian bisa menjualnya) perangkat keras Arduino ini, apalagi bootloader tersedia langsung dari perangkat

lunak Arduino IDE-nya. Bisa juga menggunakan breadboard untuk membuat perangkat arduino beserta peripheral-periferer lain yang dibutuhkan.

- Tidak perlu perangkat chip programmer. Karena didalamnya sudah ada bootloader yang akan menangani upload program dari komputer.
- Sudah memiliki sarana komunikasi USB. Sehingga pengguna laptop yang tidak memiliki port serial/RS323 bisa menggunakannya.
- Bahasa pemrograman yang relative mudah, karena software Arduino dilengkapi dengan kumpulan library yang cukup lengkap.
- Memiliki modul siap pakai (shield) yang bisa ditancapkan pada board Arduino. Misalnya shield GPS, Ethernet , SD Card, dll.

Penggunaan dan Pemanfaatan Arduino

Kegunaan Arduino tergantung kepada kita yang membuat program. Arduino bisa digunakan untuk mengontrol LED, Mengontrol lampu lalu lintas, bisa juga digunakan untuk mengontrol helikopter. Sudah banyak contoh yang sudah pernah dibuat antaranya MP3 Player. Pengontrol motor , mesin CNC, Monitor kelembaban tanah, pengukur jarak, pengerak servo , balon udara, pengontrol suhu, monitor energy, stasiun cuaca, pembaca RFID, drum elektronik , GPS logger, monitoring mesin , dan masih banyak lagi

Jenis-jenis Perangkat Keras Arduino (Arduino Hardware)

Saat ini ada bermacam-macam bentuk dan jenis papan Arduino yang disesuaikan dengan pembentuknya, tidak hanya board (papan) Arduino yang disediakan juga terdapat modul siap pakai (shield), juga aksesoris seperti USB adapter dan sebagainya. Berikut jenis-jenis papan Arduino yang ada di pasaran.

Papan/Board Arduino

Arduino Uno

Arduino Uno adalah papan mikrokontroler berbasis ATmega328 yang memiliki 14 pin digital input/output (dimana 6 pin dapat digunakan sebagai output PWM) , 6 input analog, clock speed 16 MHz, koneksi USB, Jack listrik, header ICSP, dan tombol reset. Board ini menggunakan daya yang terhubung ke komputer dengan kabel USB atau daya eksternal dengan adaptor AC-DC atau baterai.

Arduino Uno adalah pilihan yang baik untuk pertama kali atau bagi pemula yang ingin mengenal Arduino. Disamping sifatnya yang reliable juga harganya murah.

Spesifikasi Board Arduino Uno

▪ Mikrokontroler	ATmega328
▪ Tegangan Operasi	5V
▪ Tegangan Input (disarankan)	7-12V
▪ Batas Tegangan Input	14(dimana 6 pin output PWM)
▪ Pin Analog Input	6
▪ Arus DC per I/O pin	40 mA
▪ Arus DC untuk pin 3.3V	50 mA
▪ Flash memory	32 KB(ATmega328) dimana 0,5
KB digunakan	oleh bootloader
▪ SRAM	2 KB (ATmega328)
▪ EEPROM	1 KB (ATmega328)
▪ Clock	16 MHz

Sumber (Catu Daya)

Arduino dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Sumber daya eksternal (non-USB) dapat berasal dari adapter AC ke DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan power jack, dapat juga dihubungkan pada power pin (Gnd dan Vin)

Board Arduino Uno dapat beroperasi pada pasokan eksternal dari 6 sampai 20 volt. . Meskipun pin 5V dapat disuplai kurang dari 5 volt, board mungkin tidak stabil. Jika menggunakan tegangan lebih dari 12 V , regulator tegangan bisa panas dan merusak Board. Kisaran yang disarankan adalah 7 sampai 12 volt.

Adapun pin power suplai pada Arduino Uno adalah :

- VIN. Tegangan input board arduino ketika menggunakan sumber daya (5 Volts dari sambungan USB atau dari sumber regulator lain). Anda dapat mensuplai tegangan pada pin ini jika suplai tegangan lewat power jack , dapat mengakses melalui pin ini.
- 5V. keluaran pin ini telah diatur sebesar 5V dari regulator pada board. Board dapat disuplai melalui DC jack power (7-12V). konektor USB (5V), atau pin VIN (7-12 V).

menyuplai tegangan melalui pin 5V atau 3.3V bypasses regulator, dapat merusak board.

- 3V3. Suplai 3,3 volt dihasilkan oleh regulator pada board menarik arus maksimum 50 mA.

Memory

ATMEGA328 mempunyai memori 32 KB (dengan 0.5 KB digunakan untuk bootloader). Juga mempunyai 2 KB SRAM dan 1 KB EEPROM (yang mana dapat di baca tulis dengan library EEPROM).

Input dan Output

Setiap pin digital pada board Arduino Uno dapat digunakan sebagai input ataupun output. Dengan menggunakan fungsi pin model () digital write(), dan digital read(). Pin-pin ini beroperasi pada tegangan 5Volt. Setiap pin mampu memberikan atau menerima arus maksimum dan memiliki resistor pull-up internal (secara default tidak terhubung) dari 20-50 kOhm. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) dan serial TTL. Pin ini terhubung ke pin yang sesuai dari chip ATmega 8U2 USB-to-TTL Serial
- Interupsi Eksternal: 2 dan 3. Pin ini dapat dikonfigurasi untuk memuncu interrupt pada nilai yang rendah , tetapi naik atau turun, atau perubahan nilai.
- PWM:3, 5, 6, 9, 10, dan 11. Menyediakan 8-bit output PWM dengan fungsi analogWrite()
- SPI: 10 (SS), 11 (MOSI), 12(MISO), 13 (SGK). Pin ini mendukung komunikasi SPI menggunakan library SPI.
- Led: 13. Terdapat LED pin digital 13 pada board. Ketika pin bernilai tinggi (HIGH), LED menyala (ON), ketika pin bernilai rendah (LOW), LED akan mati (OFF)
- Arduino Uno memiliki 6 input analog ,belebel A0 sampai A5, yang masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara default , 5 volt dari ground

Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau Mikrokontroler lainnya. ATmega328 menyediakan UART TTL (5V) komunikasi serial, yang tersedia di pin digital 0 (RX) dan 1 (TX).

Bahasa Pemrograman Arduino

Banyak bahasa pemrograman yang bisa digunakan untuk program mikrokontroler, misalnya bahasa basic, assembly, atau bahasa C. Namun dalam pemrograman Arduino bahasa yang dipakai adalah bahasa C. Bahasa C adalah bahasa yang sangat lazim dipakai sejak awal komputer diciptakan dan sangat berperan dalam perkembangan software.

Bahasa C telah membuat bermacam-macam sistem operasi dan compiler untuk banyak bahasa pemrograman. Misalnya sistem operasi unix, linux, dsb. Bahasa C adalah bahasa pemrograman yang sangat ampuh yang kekuatannya mendekati bahasa assembler. Bahasa C menghasilkan file kode yang sangat kecil dan dieksekusi dengan sangat cepat. Karena itu, Bahasa C sering digunakan pada sistem operasi dan pemrograman mikrokontroler.

Bahasa C adalah multi-platform karena bahasa C bisa diterapkan pada lingkungan windows, unix, linux, atau sistem operasi lain tanpa mengalami perubahan source code. (kalaupun ada perubahan, biasanya sangat minim). Karena Arduino menggunakan bahasa C yang multi-platform, software arduino pun bisa dijalankan pada saat sistem operasi yang umum, misalnya Windows, Linux, dan MacOS.

Di internet banyak Library Bahasa C untuk Arduino yang bisa di download dengan gratis. Setiap library Arduino biasanya disertai dengan contoh pemakaiannya. Keberadaan Library-library ini bukan hanya membantu kita membuat mikrokontroler, tetapi bisa dijadikan sebagai saran untuk mendalami pemrograman bahasa C pada mikrokontroler.

Berikut ini adalah sedikit penjelasan yang ditujukan kepada anda yang hanya mampu sedikit pengalaman pemrograman dan membutuhkan penjelasan singkat mengenai karakter bahasa C dan software Arduino. Penjelasan lebih detail, lengkap, dan mendalam, lihat situs resmi Arduino.

Struktur

Setiap program arduino (biasa disebut Sketch) mempunyai dua buah fungsi yang harus ada.

Void Setup() { }

Semua kode di dalam kurung kurawal akan dijalankan hanya satu kali ketika program arduino dijalankan untuk pertama kalinya.

Void Loop () { }

Fungsi ini akan dijalankan setelah setup (fungsi void setup) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi , dan lagi secara terus-menerus sampai catu daya (power) di lepaskan.

Syntax

Berikut ini adalah bahasa C yang dibutuhkan untuk format penulisan.

- // (komentar satu baris.
- Kadang di perlukan untuk memberikan catatan pada arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan di belakangnya akan diabaikan oleh program.
- Contoh penggunaan :
// proyek Blink LED, uji cobapertama kali
- /* */ (komentar banyak baris)
- Jika anda punya banyak catatan, hal ini dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak diantara dua symbol tersebut akan diabaikan oleh program.
- Contoh penggunaan:
/* Kode program proyek sensor cahaya,
LED padam kondisi lingkungan terang dan nyala
Jika kondisi lingkungan gelap */
- { } (Kurung kurawal)
Digunakan untuk mendefenisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).

Contoh penggunaan kurung kurawal:

```
Void loop () {  
  Serial.println (val)  
}
```

- ; (titik koma)

Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka tidak akan bisa dijalankan).

- Contoh penggunaan titik koma : `delay (1000);`

Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas. Variable inilah yang digunakan untuk memindahkan.

- **Int (integer)**

Digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak mempunyai angka decimal dan menyimpan nilai dari -32.768 dan 32.767

- **Long (long)**

Digunakan ketika integer tidak mencukupi lagi. Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari 2.147.483.648 dan 2.147.483.647.

- **Boolean (Boolean)**

Variable sederhana yang digunakan untuk menyimpan nilai true (benar) atau False (salah). Sangat berguna karena hanya menggunakan 1 bit dari RAM

- **Float**

Digunakan untuk angka decimal (floating point). Memakai 4 byte (32 bit) dari RAM mempunyai rentang dari -3,4028235E+38 dan -3,4028235E+38.

- **Char (character)**

Menyimpan 1 karakter menggunakan kode ASCII (misalnya 'A' = 65). Hanya memakai 1 byte (8 bit) dari RAM.

- **Byte**

Angka antara 0 dan 255. Sama halnya dengan char , namun byte hanya menggunakan 1 byte memory.

- **Unsigned int**

Sama dengan int , menggunakan 2 byte tetapi Unsigned int ini tidak dapat digunakan untuk menyimpan angka negative , batas nya dari 0 sampai 65,35.

- **Unsigned Long**

Sama saja dengan long, namun tidak bisa menampung angka negative , batas nya dari 0 sampai 4.294.967.295.

- **Double**

Angka ganda dengan persisi maksimum $1,7976931348623157 \times 10^{308}$

- **String**

String digunakan untuk menyimpan informasi teks, dengan karakter ASCII , bisa menggunakan string untuk mengirim pesan via serial port , atau menampilkan teks pada layar LCD.

- **Array**

Array adalah kumpulan variable dengan tipe yang sama. Setiap variable dalam kumpulan variable tersebut terdapat elemen, dapat diakses melalui indeks. Misalnya kita ingin menginisialisasi pin 3 , pin 5 , pin 6 , pin 7, maka dengan menggunakan array int pins[]={3 , 5, 6, 7};

Operator Matematika

Operator yang digunakan untuk memanipulasi angka (bekerja seperti matematika yang sederhana).

- = membuat suatu menjadi sama dengan nilai yang lain (misalnya: $x = 10 * 2$, x sekarang sama dengan 20)
- % menghasilkan sisa dari hasil pembagian suatu angka dengan angka yang lain (misalnya: $12 \% 10$, ini akan menghasilkan angka 2)

- + penjumlahan
- - pengurangan
- / pembagian

Oprator Pembanding

Digunakan untuk membandinhkan nilai logika.

- == sama dengan (misalnya: $12==10$ adalah FALSE (salah) $12==12$ adalah TRUE (benar))
- != tidak sama dengan (misalnya: $12!=10$ adalah TRUE (benar) atau $12!=12$ adalah FALSE(salah))
- < lebih kecil dari (misalnya: $12<10$ adalah FALSE (salah) aatau $12<12$ adalah FALSE (salah) atau $12 < 14$ adalah TRUE (benar))
- > lebih besar dari (misalnya: $12 > 10$ adalah TRUE (benar) aatau $12>12$ adalah FALSE (salah) atau $12 >14$ adalah FALSE (salah))

Struktur Pengaturan

Pemrograman sangat tergantung pada pengaturan yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan (banyak lagi yang lain dan bisa dicari diinternet.

1. If...else, dengamn format seperti berikut ini:

```
If (kondisi) { }
```

```
Else { }
```

Dengan struktur seperti di atas program akan menjalankan kode yang ada pada kurung kurawal jika kondisinya TRUE, dan jika tidak (FALSE) maka akan diperiksa apakah kondisi pada *else if* dan jika kondisinya FALSE maka kode pada else yang akan dijalankan.

2. For, dengan format seperti berikut in:

```
For (int i = 0; i <#pengulangan;i++) { }
```

Digunakan bila anda ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti #pengurangan dengan jumlah pengulangan yang diinginkan. Melakuan perhitungan ke atas dengan i++ atau kebawah dengan i

Digital

1. PinMode(pin, mode)

Digunakan untuk menetapkan mode dari suatu pin, pin adalah nomor pin yang akan digunakan dari 0-19 (pin analog 0-5 adalah 14-19). Mode yang bisa digunakan adalah INPUT atau OUTPUT.

2. Digital Write(pin , value)

Ketika sebuah pin ditetapkan sebagai OUTPUT, pin tersebut dapat dijadikan HIGH (ditarik menjadi 5 volts) atau LOW (diturunkan menjadi ground)

Analog

Arduino adalah mesin digital, tetapi mempunyai kemampuan untuk beroperasi didalam alam analog (menggunakan trik). Berikut ini cara untuk menghadapi hal yang bukan digital.

1. analogWrite (pin,value)

beberapa pin pada arduino UNO mendukung PWM (*pulsewidth modulation*) yaitu pin 3 , 5 , 6 ,7, 10, 11. Ini dapat mengubah pin hidup (on) atau mati (off) dengan sangat cepat sehingga membuatnya dapat berfungsi layaknya keluarananalog. Value (nilai) pada format kode tersebut adalah angka antara 0 (0%duty cycle ~ 0V) dan 255 (100% duty cycle ~5V)

2. analogRead (pin)

ketika pin analog ditetapkan sebagai INPUT ,anda dapat membaca keluaran voltasenya. Keluarannnya berupa angka antara 0 (untuk 0 volts) dan 1023 (untuk 5 volts)

panduan bahasa pemrograman Arduino beserta dengan contoh-contohnya,juga dilengkapi library bisa dibaca dihalaman <http://arduino.cc/en/Reference/Homepage>.

Cara Mendapatkan Arduino

Papan rangkaian atau board Arduino bisa dibuat sendiri kalau anda memiliki chip programmer (untuk meng-upload bootloader ke dalam chip Atmel yang masih kosong). Kalau merasa kesulitan untuk membuat PCB dan mendapatkan komponen, anda bisa langsung membeli board yang sudah jadi dan siap pakai, seperti yang digunakan dalam workshop kali ini..

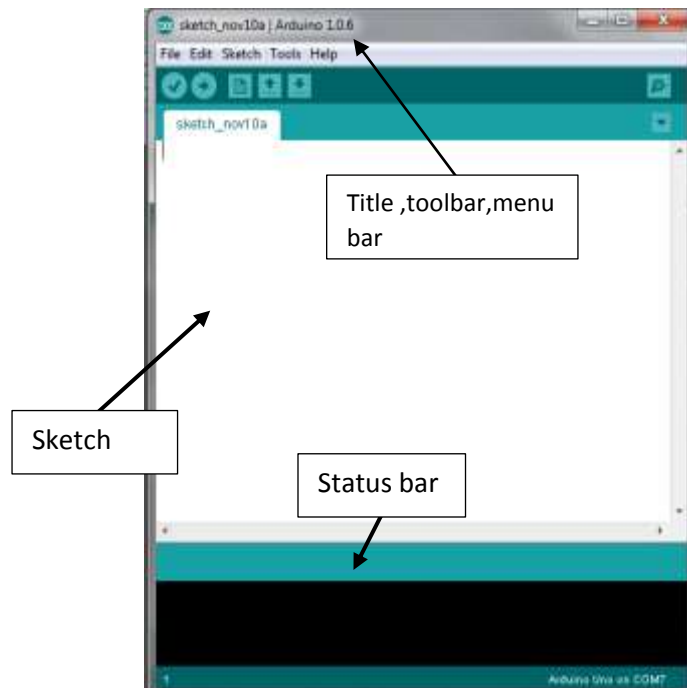
SOFTWARE ARDUINO IDE

Software ini digunakan untuk menulis program pada board arduino. IDE (*integrated Development Emvironment*) adalah sebuah software yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan mengupload ke dalam memory mikrokontroler. Software ini dapat di download secara gratis.

Instalasi SoftwareArduino IDE

Langkah-langkah instalasi IDE Arduino adalah:

1. Download Software IDE Arduino pada website resmi Arduino <http://arduino.cc/en/Main/Software> , hasil download berupa file kompresi, Arduino-1.0.6-windows.zip kemudian ekstrak hasil download. Atau ambil pada folder master program yang pada DVD Workshop.
2. Ekstrak file zip hasil download pada salah satu folder misal pada folder D. pada file kompres zip terdapat arduino.exe.
3. Jalankan aplikasi dengan mengklik dua kali arduino.exe.
4. Setelah klik dua kali arduino.exe diperoleh jendela utama dari software IDE arduino seperti gambar berikut



Software IDE arduino adalah software yang ditulis dengan menggunakan java. Jendela utama IDE arduino terdiri dari tiga bagian utama, yaitu :

- Bagian atas, yakni toolbar , pada bagian atas juga terdapat menu file, edit, sketch, tools, dan help
- Bagian tengah, yaitu tempat penulisan kode program atau sketch
- Bagian bawah berupa status bar/ jendela pesan (message windows) atau tes konsul yang berisi status dan pesan error.

Penjelasan tentang toolbar

Fungsi pada bagian toolbar adalah :

Verify : mengecek kode sketch yang error sebelum meng-upload ke board arduino

Upload : meng-upload sketch pada board Arduino

New : membuat sebuah sketch baru

Open : membuat daftar sketch pada sketchbook untuk dibuka

Save : menyimpan kode atau sketch pada sketchbook

Serial Monitor : Menampilkan data serial yang dikirimkan dari board Arduino

Sedangkan bagian dari IDE menu diperlihatkan seperti berikut:

Bagian menu file : terdiri dari new, open , sketchbook, example, save, save as, dan seterusnya

Bagian menu edit : terdiri dari cut, copy , copy for forum , copy as HTML, pate, select all dan seterusnya.

Bagian menu sketch : terdiri dari verify/compile, show,sketch folder, add file , dan add library.

Bagian menu tool : pada bagian tools terdapat tipe board yang kita gunakan untuk meng-upload program, seperti board arduino Uno, Arduino Duenminalove, Arduino Mega , dan seterusnya

Catatan

Sebuah kode program Arduino umumnya disebut denggan istilah **sketch**. Kata “sketch” memiliki arti yang sama dengan “Kode Program”

Instalasi Drivers USB Arduino pada Windows 7

Untuk menghubungkan atau mengomunikasikan komputer/laptop dengan board arduino, diperlukan instalasi driver. Tentunya peralatan yang dibutuhkan adalah board arduino, dalam workshop ini digunakan Arduino Uno R3 dan juga dibutuhkan kabel USB standar A-B (sama dengan kabel yang anda gunakan untuk printer USB).

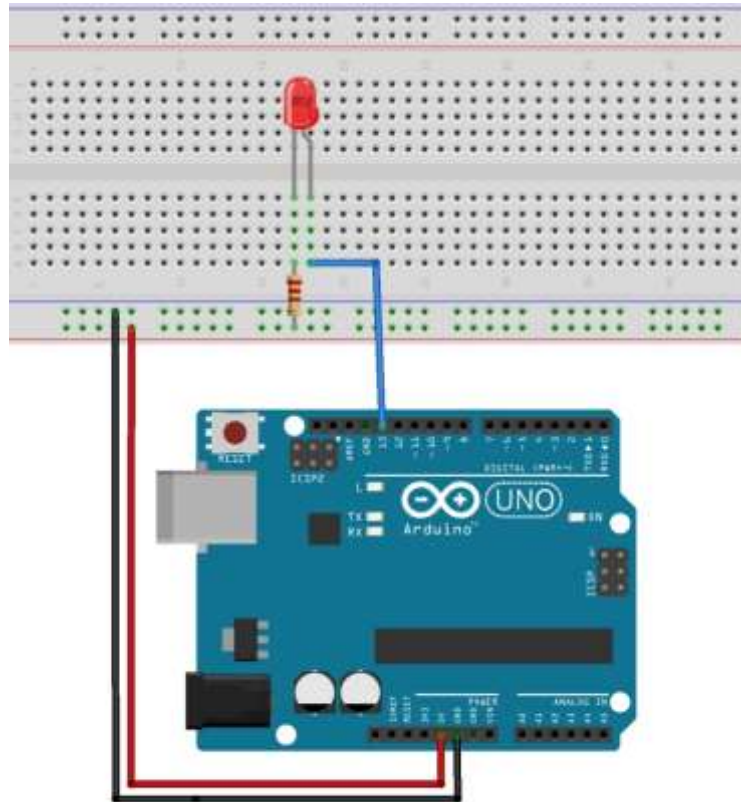
Langkah-langkah instalasi driver untuk board arduino Uno (sama juga untuk jenis board Arduino Duemilanove, nano atau diecimila) dengan sistem operasi windows 7 sebagai berikut :

1. Buka atau jalankan software arduino IDE, dengan mengklik dua kali arduino.exe
2. Hubungkan board Arduino dengan komputer menggunakan kabel USB standar (A-B) dan yakin bahwa LED warna hijau pada board arduino nyala (berlabel ON/PWR). Board arduino akan menarik sumber daya listrik dari port USB.
3. Selanjutnya, windows secara otomatis akan mendeteksi kalau tidak ditemukan driver board arduino. Jangan khawatir ini kondisi normal. Silahkan tutup saja jendela ini.
4. Klik tombol start menu pada komputer anda dan buka control panel, masuk ke menu sistem and security.
5. Kemudian klik sistem. Setelah tampilan sistem muncul, klik Device Manager. Lihat pada bagian other device. Anda akan melihat sebuah tulisan Unknown device. Ini menandakan bahwa board arduino anda belum dikenal windows 7.
6. Klik kanan pada port "Unknown device" dan pilih opsi "update driver software"
7. Kemudian , pilih opsi "Browse my komputer for driver software"
8. Telusuri folder arduino 1.0.6 yang anda ekstrak tadi.
9. Pilih next. Windows akan meneruskan instalasi driver.
10. Setelah klik next akan muncul jendela piringan dari windows security, pilih install this driver software anyway.
11. Klik close untuk mengakhiri instalasi. Selamat anda telah menginstal driver Arduino pada komputer anda, dan selanjutnya siap mengomunikasikan atau meng-upload program pada board arduino anda.

12. Untuk mengecek apakah driver telah benar terinstal, masuk kembali device manager. Untuk melihat port (com) dari arduino.

Uji Sketch Pertama

1. buka software IDE Arduino dengan menjalankan sebuah file bernama arduino.exe pada lokasi software arduino di komputer anda.
2. setelah jendela utama terbuka klik file →example→basics→blink
3. selanjutnya akan muncul kode blink program pada kotak entri/penulisan sketch
4. kode program siap di upload ke papan arduino



Gambar rangkaian LED

Kode program led berkedip

/* simulasi dan praktik pertama led berkedip

untuk uji simulasi dan upload program

```
*/  
  
Int led = 13 ; // led terhubung pada pin 13  
  
Void setup () {  
  
    pinMode (led,OUTPUT); // set led sebagai output  
  
}  
  
Void loop () {  
  
    digitalWrite (led, HIGH); // set led on  
  
    delay (500) ; // tunnda untuk ½ detik  
  
    digitalWrite (led, LOW); // set led off  
  
    delay (500) ; // tunnda untuk ½ detik  
  
}
```

Penjelasan program

1. pada baris pertama , kode program
/* simulasi dan praktik pertama led berkedip
untuk uji simulasi dan upload program

*/

Kode ini /*..*/ digunakan untuk menuliskan catatan untuk komentar dengan banyak bari misal nama project, prinsip kerja, dll. Semua hal yang terletak di antara dua symbol tersebut akan diabaikan oleh program. Komentar untuk baris digunakan kode //.

2. Pada baris kedua

```
Int led = 13;
```

Kode ini dikenal sebagai variable. Variable adalah tempat menyimpan data. Pada kode ini kita mensetting variable jenis int atau integer. Integer digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak mempunyai angka decimal dan menyimpan nilai dari -32.768 dan 32.767. selanjutnya dari kode di atas diberi nama led dan mempunyai nilai 13. Pada pemrograman arduino , nama variable

led merupakan pendefinisian led sebagai nilai 13, ini digunakan untuk menandai nomor saluran yang digunakan untuk pin arduino terhubung dengan led. Dengan kata lain led dipasang pada pin 13.

3. Baris program berikutnya

```
Void setup () {  
    pinMode (led,OUTPUT);
```

Fungsi void setup (), merupakan fungsi yang dieksekusi pertama kali saat setelah arduino diberi catu daya, berfungsi untuk perintah-perintah instalasi sebelum program utama dieksekusi isi dari fungsi ini adalah pendefinisian pin 13 dengan syntax : pinMode(led, OUTPUT); memiliki arti bahwa mode dari pin yaitu saluran D13 (digital nomor 13 dari pin input/output) difungsikan sebagai saluran keluaran atau output. Dengan kata lain, menset pin 13 sebagai output .

4. Baris program berikutnya

```
Void loop () {  
    digitalWrite (led, HIGH);  
    delay (500) ;  
    digitalWrite (led, LOW);  
    delay (500) ;
```

fungsi ini akan dijalankan setelah setup (fungsi Void setup) selesai. Fungsi void loop (), merupakan fungsi dari program utama yang akan dieksekusi secara berurutan dari atas sampai akhir dan kembali lagi dari atas sampai akhir, begitu seterusnya karena merupakan loop yang tidak pernah berakhir. Baris pertama dari fungsi ini adalah digitalWrite(led HIGH); artinya tuliskan logika 1 (high) pada pin keluaran nomor 13, dengan demikian led yang mengeksekusinya syntax delay (500) ; yaitu waktu tunda selama 500 milisekon atau ½ detik , dengan demikian led menyala selama setengah detik , berikutnya adalah digitalWrite (led, LOW); artinya tuliskan logika 0 (low) pada pin keluaran nomor 13, dengan delay (500) mengakibatkan led padam selama ½ detik . keseluruhan program, led berkedip hidup selama setengah detik , padam selama setengah detik . demikian seterusnya catu daya pada Arduino dilepas.

Upload Program Pertama “Led Berkedip” Ke Board Arduino

Jika pertama kali belajar bahasa pemrograman apapun (basic,java,html,dll) biasanya pelajaran pertama kita , yaitu program “hello world”. Nah pada Arduino ,

program “hello world” ini adalah led berkedip (blinking led) , program ini kita gunakan untuk menguji papan arduino yang kita beli , apa benar-benar berfungsi sebagaimana mestinya.

Daftar komponen dan peralatan yang diperlukan

1. Papan Arduino Uno R3
2. Kabel USB standar A-B

Pada papan Arduino Uno terdapat sebuah led kecil yang terhubung ke pin digital no 13. Led ini terdapat digunakan sebagai output saat seorang pengguna membuat/menguji sebuah program dan ini membutuhkan sebuah penanda dari jendela program tersebut. Ini adalah cara yang praktis saat pengguna melakukan uji coba.

Langkah-langkah upload uji program pertama ke papan arduino sebagai berikut:

1. Jalankan software IDE arduino.
2. Colokkan kabel USB yang tersambung dengan papan arduino ke komputer anda. Pastikan windows sudah dapat mengenal papan arduino anda (serial port)
3. Selanjutnya klik tools→board→Arduino Uno.
Kemudian ketik atau maArduino Uno.
4. Kemudian ketik atau masukkan kode program “led berkedip”
5. Sebelum melakukan upload ke papan arduino sebaiknya kita cek program/sketch yang kita masukkan , apa ada yang eror atau tidak, caranya klik icon verify.
Bila kode program/sketch tidak ada yang erro, maka jendela teks konsul terdapat keterangan ukuran byte program.
6. Pada toolbar, klik ikon Upload untuk membuar sketch tersebut kedalam papan arduino , berkedip-kedip kan? Berhasi...!!!!

Jadi sangat mudah menguji apakah sebuah papan arduino yang baru kita beli dalam kondisi baik atau tidak. Cukup sambungkan ini dengan komputer dan perhatikan apakah led indicator daya menyala konstan led dengan pin 13 itu menyala berkedip-kedip atau tidak, jika menyala

berkedip berarti board arduino dan instalasi drivernya telah benar, dan kita bisa melanjutkan ke pembuatan sketch-sketch lainnya.

Proyek 1 Blink Tiga Led

Pada proyek 1 kita akan menggunakan komponen eksternal (resistor ataupun led eksternal), karena pada praktik uji coba sebelumnya hanya menggunakan led internal yang terdapat pada papan arduino.

1. Kode Program

Tulis kode program untuk Blink Tiga LED pada Arduino IDE di computer anda :

```
/*  
  
Proyek 1- Blink Tiga LED  
  
Tiga LED berkedip pada digital pin yang berbeda  
  
*/  
  
Int led1 = 3; // inialisasi setiap pin LED  
  
Int led2 = 5;  
  
Int led3 = 7;  
  
void setup()  
{  
  
    pinMode (led1, OUTPUT); // set LED pin sebagai output  
  
    pinMode (led2, OUTPUT); // set LED pin sebagai output  
  
    pinMode (led3, OUTPUT); // set LED pin sebagai output  
  
}  
  
Void loop ()  
{  
  
    // Detiap LED berkedip 500 milisecond (1/2 detik)
```

```
    blinkLED (led1 , 500);  
  
    blinkLED (led2 , 500);  
  
    blinkLED (led3 , 500);  
  
}  
  
// durasi kedipan LED dalam millisecond  
  
Void blinkLED (int pin , int duration )  
  
{  
  
    digitalWrite(pin, HIGH) ; // LED on delay (duration);  
  
    digital Write(pin,LOW); // LED off delay (duration);  
  
}
```

2. Penjelasan Program:

1. Pada baris 1-3 menggunakan variable int atau integer.

```
Int led1 = 3;
```

```
Int led2 = 5;
```

```
Int led3 = 7;
```

dimana variable integer int led1 = 3 menyiapkan nilai 3 yang menandai nilai LED pertama disetting pada pin 3 pada papan Arduino . demikian halnya LED kedua dan ketiga diinisialisasikan masing-masing sebagai 5 pin dan 7 pin [ada papan Arduino.

2. Pada baris berikutnya, menggunakan fungsi Void setup() yang diikuti perintah,

```
pinMode (led1, OUTPUT);
```

```
pinMode (led2, OUTPUT);
```

```
pinMode (led3, OUTPUT);
```

yang menset pin 3, pin 5 , pin 7 sebagai keluaran digital atau output digital.

3. Pada baris berikutnya fungsi void loop() yang diikuti perintah

```
blinkLED (led1 , 500);
```

```
blinkLED (led2 , 500);
```

```
blinkLED (thirdLedPin , 500);
```

yang merupakan fungsi loop yang memanggil blinkLED agar setiap LED dibuat berkedip 500 milisekon (1/2 detik). Anda dapat merubah nilai 500 menjadi 1000, 1500, untuk kedipan yang lebih lambat. Kemudian dilanjutkan dengan , void BlinkLED(int pin, int duration), yang berarti mensetting blinkLED agar ketiga pin LED bergantian/berpindah (peralihan diantara ketiga pin) dalam durasi beberapa milisekon dan meng-ON dan OFF kan LED dengan perintah seperti empat baris berakhir dari kode program.

Praktik Proyek Blink Tiga Led

Bahan dan komponen

Daftar komponen yang digunakan pada proyek ini yaitu:

1 Pada Arduino Uno R3

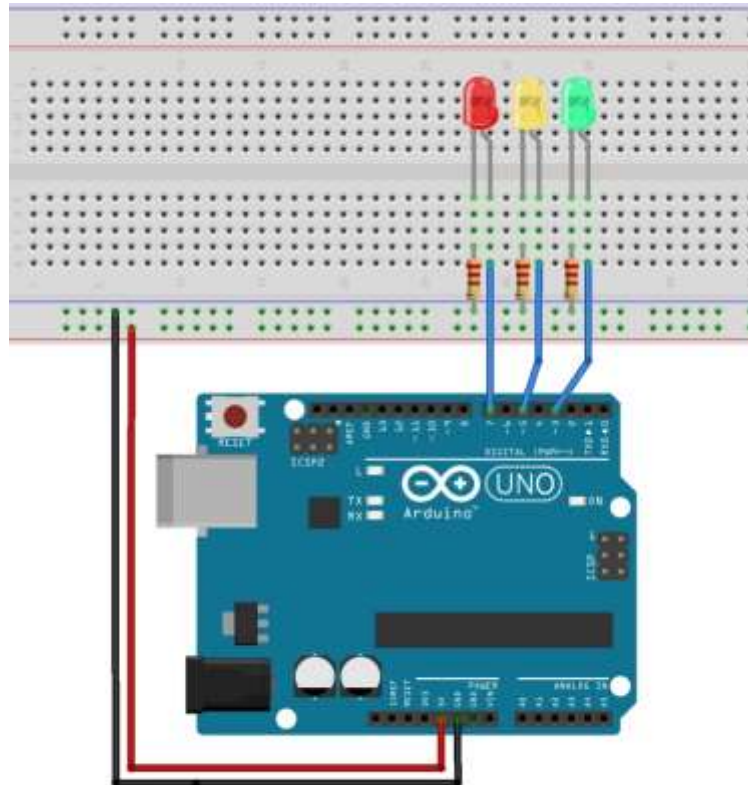
3 LED Merah

1 Breadboard

3 Resistor 220 ohm

→ kabel Jumper

Membuat Rangkaian



Sebelum menghubungkan setiap komponen kedalam rangkaian , lepaskan colokan kabel USB Arduino komputer. Rangkailah komponen satu persatu sesuai dengan rangkaian yang ditunjukkan pada gambar 5.13

Langkah-langkah Membuat Rangkaian:

1. Susun resistor 220 ohm terlebih dahulu pada breadboard seperti gambar rangkaian.
2. Pasang 3 led merah pada breadboard. Perhatikan polaritas kaki LED. Kaki positif masing-masing led dihubungkan dengan pin 3, pin 5, pin 7, sementara kaki negative dihubungkan dengan ground melewati resistor.
3. Gunakan kabel jumper untuk menghubungkan papan arduino dengan breadboard.
4. Setelah semua terpasang dengan baik, colokkan kabel USB yang terhubung dari papan arduino ke komputer.
5. Selanjutnya, jalankan software IDE arduino dan ketikkan kode program proyek Blink Tiga LED pada tempat penulisan sketch kemudian lakukan verifikasi program untuk mengecek error.

6. Klik icon upload pada toolbar , untuk mengupload sketch ke papan arduino. Lihat hasilnya. Perhatikan LED pada breadboard, tiga LED berkedip silih berganti , Selamat! Proyek blink Tiga LED anda berjalan

Catatan :

Coba anda buat dan uji program dengan 5 LED berkedip dengan durasi kedipan setiap LED 100 milisekon.

Proyek 2 Mini Traffic Light

6.1 KODE PROGRAM

Program berikut ini menggunakan digital output aduino untuk aplikasi sebuah mini trafficligh, diman LED merahakan menyala sebagai tanda berhenti, lalu di susul LED kuning sebagai tanda hati-hati , selanjutnya LED hijau menyala sebagai tanda jalannya kendaraan. Kode program mini Traffic Light yang akan kita simulasikan dan praktikkan pada papan Arduino

Penjelasan Program

1. Pada baris 1-5 menggunakan variable ini atau integer,
Int LedDelay = 10000;
Int redpin = 10;
Int yellowpin = 9;
Int greenpin = 8;

Diman variable integer ledDelay merupakan pendefenisian nilai tunda nyala setiap LED selama 10 detik, variable integer redpin menyimpan nilai 10, yang menandai LED merah disetting/ditanvapkan pada pin 10 board Arduino. Demikian halnya LED kuning dan LED hijau di setting atau diinisialisasikan unt hijau di setting atau diinisialisasikan untuk pin 9 dan pin 8.

2. Pada baris berikutnya

```
Void setup ()  
    pinMode (redpin, OUTPUT);  
    pinMode (yellowpin, OUTPUT);  
    pinMode (greenpin, OUTPUT);
```

yang menseset pin 8, pin 9, dan pin 10 sebagai keluaran digital atau output digital.

3. Pada baris berikutnya

Void loop ()

```
digitalWrite(redpin,HIGH);  
delay (ledDelay);  
digitalWrite(yellowpin, HIGH);  
delay (2000)  
digitalWrite(greenpin , HIGH);  
digitalWrite (redpin, LOW);  
digitalWrite (yellowpin , LOW);  
delay(ledDelay);  
digitalWrite (yellowpin , HIGH);  
digitalWrite (greenpin, LOW);  
delay(2000);  
digitalWrite (yellowpin , LOW);
```

fungsi dengan kode digitalWrite (pin, value) diartikan ketika sebuah pin ditetapkan sebagai OUTPUT, pin tersebut dapat dijadikan HIGH (5 volts) atau LOW (0 volt). Dengan demikian, digitalWrite (redPin, HIGH); menyatakan bahwa pin output LED merah dijadikan HIGH (5 volt) atau LED menyala. Selanjutnya kode delay (ledDelay); seperti kita lihat sebelumnya ledDelay disimpan sebagai variable integer selama 10 detik, maka lampu LED akan menyala selama 10 detik. Nah, perintah delay (ledDelay) menyatakan bahwa LED akan tetap menyala dalam beberapa milisekon setelah sebelumnya nyala 10 detik. Jika spesifik ditulis **delay(2000); berarti LED dinyalakan atau OFF dalam 2 detik atau 2000 milisekon.** Selanjutnya, lihat keterangan di samping kode.

Prinsip kerja program : Pertama LED merah menyala selama 10 detik, selanjutnya LED kuning menyala 2 detik (led merah tetap menyala beberapa detik), lampu merah dan kuning off, dan lampu hijau menyala 10 detik, kemudian OFF sesaat dalam 2 detik, selanjutnya lampu kuning menyala demikian seterusnya secara berulang-ulang. Untuk lebih jelas, lihat gambar 6.1 dan hasil simulasinya nanti.

6.3 PRAKTIK PROYEK MINI TRAFFIC LIGHT

Bahan dan Komponen

Daftar komponen yang kira pakai pada proyek ini,yaitu :

1 Board Arduino Uno – R3

1 led merah

1 led kuning

1 led hijau

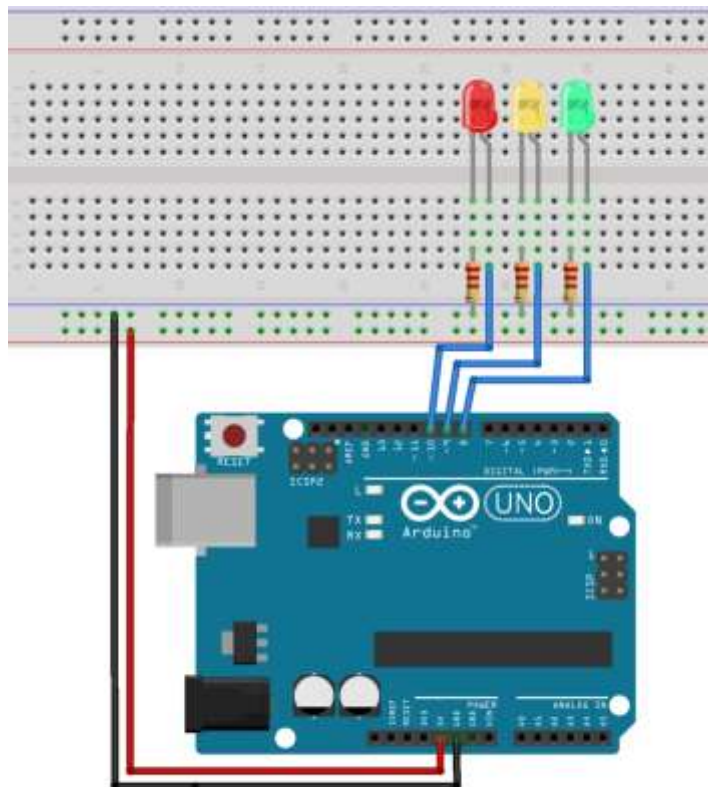
1 Beradboard

3 resistor 22 ohm

→ Kabel Jumper

Menbuat Rangkaian

Sebelum menghubungkan setiap komponen kedalam rangkaian lepaskan colokan kabel USB Arduino dari komputer. Rangkailah komponen satu persatu sesuai dengan rangkain yang ditunjukkan pada gambar 6.16



Gambar 6.16. Rangkaian Traffick Light

Langkah-langkahnya :

1. Susun resistor 220 ohm terlebih dahulu pada breadboard seperti gambar rangkaian
2. Pasang LED merah , LED kuning , LED hijau. Perhatikan polaritas kaki LED. Kaki positif masing-masing LED dihubungkan dengan pin 10 , pin 9 , dan pin 8, sementara kaki negative dihubungkan dengan ground
3. Gunakan kabel jumper untuk menghubungkan papan Arduino dengan Breadboard.
4. Setelah semua terpasang dengan baik, colokkan kabel USB yang terhubung dari papan Arduino ke komputer.
5. Selanjutnya, jalankan software IDE arduino dan ketikkan kode program proyek Mini Traffic Light pada tempat penulisan sketch
6. Lakukan verifikasi terlebih dahulu untuk mengecek apakah ada kode program yang error
7. Klik icon upload pada toolbar , untuk mengupload sketch ke papan arduino. Lihat hasilnya. Perhatikan LED pada breadboard, LED pada breadboard , Selamat! Proyek Mini traffic Light anda berjalan

Proyek 3 Brightness LED

7.1 KODE PROGRAM

Semua pin Arduino yang dapat digunakan sebagai digital input , dapat juga digunakan sebagai digital output. Namun tidak semua pin input/output (I/O) digital tersebut bisa kita gunakan sebagai analog output. Untuk tipe papan arduino Uno – R3 yang kita pakai sekarang , memiliki 14 pin input/output, dimana 6 pin dari 14 pin digital tersebut dapat digunakan sebagai output analog atau yang dikenal sebagai output PWM (pulse Width Modulation). Yaitu 3, 5, 6, 9 , 10 dan 11. Pin ini menyediakan 8-bit output PWM dengan fungsi `analogWrite()`.

Fungsi analog write dari PWM sebenarnya tidak benar-benar analog, meskipun dapat berperilaku seperti analog, karena teknik pulse width modulation (PWM) mengemulasi sebuah sinyal analog dengan menggunakan pulsa digital. Nah program yang kita simulasi dan praktikkan disini, menggunakan fungsi `analogWrite ()` untuk mengontrol intensitas cahaya LED yang terhubung pada pin Arduino adapun kode program Brightness LED yang akan kita simulasikan dan praktikkan dapat di lihat sebagai berikut :

penjelasan Program

1. Pada baris 1-5 menggunakan variable itu atau integer

```
Const int firstLed = 3;  
Const int secondLed = 5;  
Const int thirdLed = 6;  
Int brightness = 0;  
Int increment = 1;
```

Yang menginisialisasi setiap LED pada pin PWM Arduino, yaitu masing-masing LED pada pin 3, pin 5, pin 6. Selanjutnya menginisialisasi atau mendeklarasikan nilai awal tingkat intensitas cahaya LED sama dengan 0, demikian juga inisialisasi nilai kenaikan intensitas cahaya LED sama dengan 1

2. Pada baris ke 6 menggunakan fungsi void setup (), tanpa ada kode sketch, karena penggunaan fungsi analogWrite () tidak dibutuhkan atau tidak perlu deklarasi pin sebagai output.
3. Pada baris berikutnya, menggunakan fungsi void loop () yang diikuti kode pengaturan

```
If (brightness > 255),  
Increment = -1;  
Else if (brightness < 1 )  
Increment = 1;
```

Yang menyatakan jika tingkat intensitas/kecerahan lebih besar 225, maka hitung mundur atau penurunan intensitas, sebaliknya jika intensitas lebih kecil 1 maka hitung naik atau naikan intensitas cahaya LED. Dengan kata lain nilai minimum untuk analogWrite adalah 0 (tegangan 0 volt pada pin). Selanjutnya kode, `brightness = brightness + increment;` berarti ketika variable brightness mencapai nilai maksimum maka kecerahan/intensitas akan mulai menurun, karena tanda selisih yang berubah dari + 1 ke -1. Kemudian kode berikut menyatakan nyalakan setiap LED sesuai tingkat kecerahan yang telah diatur dari fungsi if-else diatas dengan perintah ,

```
analogWrite(firstLed, brightness);  
  
analogWrite(secondLed, brightness);
```

```
analogWrite(thirdtLed, brightness)
```

kemudian kode `delay(10)`; tunda selama 10 ms setiap perubahan kenaikan atau penurunan tingkat intensitas/kecerahan LED.

7.2 PRAKTIK PROYEK BRIGHNESS LED

Bahan dan Komponen

Daftar komponen yang kita pakai pada proyek ini, yaitu ;

1 board Arduino Uno R3

1 LED merah

1 breadboard

3 resistor 220 ohm

→ kabel Jumper

Membuat Rangkaian

Sebelum menghubungkan setiap komponen kedalam rangkaian lepaskan colokan kabel USB Arduino dari komputer. Rangkailah komponen satu persatu sesuai dengan rangkaian yang ditunjukkan pada gambar 7.4

Adapun langkah-langkah membangun rangkaian sebagai berikut

1. Susun dan pasang terlebih dahulu semua LED pada breadboard perhatikan polaritas kaki LED saat pemasangan
2. Susun seri setiap resistor 220 ohm dengan LED pada breadboard seperti gambar rangkaian
3. Gunakan kabel jumper untuk hubungkan papan arduino dengan breadboard.
4. Setelah semua terpasang dengan baik, colokkan kabel USB yang terhubung dari papan Arduino ke komputer.
5. Selanjutnya, jalankan software IDE arduino dan ketikkan kode program proyek brightness LED pada tempat penulisan sketch. Verifikasi program terlebih dahulu dalam pengecekan kode error.

6. Klik icon UPLOAD pada toolbar , untuk meng-upload sketch ke papan arduino. Lihat hasilnya, perhatikan LED pada Breadboard. Selamat !! Proyek brightness Led anda berjalan.

Proyek 4 Lampu Pemadam Mini

8.1 Kode Program

Penjelasan Program

1. Pada baris 1-3 menggunakan variable int atau integer,

```
Int ledPin1 = 9;
```

```
Int ledPin2 = 10;
```

```
Int ledPin3 = 11;
```

Yang menginisialisasi pin 9 , pin 10 , pin 11 , dan pada papan arduino.

2. Pada baris berikutnya, menggunakan fungsi void setup (), yaitu diikuti perintah.

```
pinMode (ledPin1, OUTPUT);
```

```
pinMode (ledPin2, OUTPUT);
```

```
pinMode (ledPin3, OUTPUT);
```

yang menset pin 9 , pin 10 , pin 11 sebagai keluaran atau output

3. Pada baris berikutnya, menggunakan fungsi void loop() yang diikuti perintah,

```
analogWrite (ledPin1, random (120) + 135);
```

```
analogWrite (ledPin2, random (120) + 135);
```

```
analogWrite (ledPin3, random (120) + 135);
```

loop program utama mengirimkan sebuah nilai acak antara 0 dan 120, ditambah 135 untuk mendapatkan nyala LED pada pin-pin PWM 9, 10, dan 11

4. Pada baris terakhir menggunakan perintah delay (random (100)), yang berarti menunda secara acak setiap perubahan nyala LED setiap 100 milisekon. Loop utama kemudian berulang terus ,sehingga menyebabkan efek berkedip, seperti pada lampu tanda mobil pemadam

8.3 PRAKTEK PROYEK LAMPU PEMADAM MINI

Bahan dan Komponen

Bahan dan komponen yang kita pakai dari started kit, yaitu :

1 Board Arduino Uno – R3

1 led merah

1 led kuning

1 led hijau

1 Breadboard

3 resistor 22 ohm

→ Kabel Jumper

Menbuat Rangkaian

Sebelum menghubungkan setiap komponen kedalam rangkaian lepaskan colokkan kabel USB Arduino dari komputer. Rangkailah komponen satu persatu sesuai dengan rangkaian yang ditunjukkan pada gambar 8.4

Langkah-langkah membangun rangkaian :

1. Susun resistor 220 ohm terlebih dahulu pada breadboard seperti gambar rangkaian
2. Pasang Led Merah dan Led Kuning. Perhatikan polaritas kaki LED. Kaki positif masing-masing Led dihubungkan dengan pin 9 , pin 10 dan pin 11, sementara kaki negative dihubungkan dengan ground
3. Gunakan kaki jumper untuk hubungkan papan Arduino dengan breadboard
4. Setelah semua terpasang dengan baik, colokkan kabel USB yang terhubung dari papan Arduino ke komputer.
5. Selanjutnya, jalankan software IDE Arduino dan ketikkan kode program proyek lampu pematik mini pada tempat penulisan sketch. Kemudian lakukan verifikasi sketch/kode program untuk mengecek dan memastikan apakah ada program yang error atau bermasalah.
6. Klik icon UPLOAD pada toolbar , untuk meng-upload sketch ke papan arduino. Lihat hasilnya, perhatikan LED pada Breadboard. Selamat !! Proyek Lampu pematik mini anda berjalan.

Proyek 5 Multiple LED

9.1 kode Program

Penjelasn program

1. Pada baris pertama menggunakan varabel int atau integer,

Int timer = 500;

Yang menginisialisasikan variable timer dengan nilai waktu 500 mikrisekon atau ½ detik.

2. Pada baris ke-2 kita membuat dan mendeklarasikan sebuah array dari integer dengan 10 elemen kode `int ledPins[]={0,1,2,3,4,5,6,7,8,9}`; Elemen pertama diset sama dengan 0, elemen kedua diset sama dengan 1, elemen ketiga diset sama dengan 2 dan seterusnya. Pada baris ke-3,kita melakukan inialisasi setiap elemen sengan kode `int pincount = 10`; Array adalah kumpulan variable dengan tipe yang sama. Setiap variable dalam kumpulan variable tersebut disebut dengan elemen.
3. Pada baris ke-4 menggunakan fungsi void `setup()`, yang diikuti perintah

For (int thisPin = 0; this pin <pinCount;

Meggunkan for loop untuk mengakses 10 elemen yang telah diinisialisasikan (pincount) tadi. Dimana loop ini dibuat dengan variable thispin dengan nilai start/awal sama dengan nol dan berakhir dengan 10. Kemudian pada baris ke-5 menggunakan kode `pinModel(ledPins[thisPin], OUTPUT)`; yang mengdeklarasikan semua pin sebagai output.

4. Pada baris berikutnya , menggunakan fungsi void `loop()` yang diikuti dengan perintah

For (int thisPin = 0 ; thisPin < pinCount;

This Pin++)

DigitalWrite(ledpins [thisPin], HIGH);

Delay(timer);

DigitalWrite(ledpins [thisPin], LOW);

Loop program untuk meng-ON kan pin atau menset semua pin pada logika 1 atau HIGH dari pin lebih rendah ke pin yang lebih tinggi dengan waktu nyala pin (LED) ½ detik. Kemudian pin OFF. Setelah itu, dilanjutkan sebaliknya dengan loop dari pin lebih tinggi ke pin rendah , dengan kode yang sama untuk meng-ON kan dan meng-OFF kan LED(pin), yang berbeda hanya kode ini **for (int thisPin = pinCount – 1; thisPin >=0;thisPin--)**

9.3 PRAKTIK PROYEK MULTIPLE LED

Bahan dan Komponen

Daftar komponen yang kita pakai, yaitu:

1 board Arduino Uno R3

5 LED merah

5 LED Kuning

1 Breadboard

10 Resistor 220 ohm

→ Kabel Jumper

Membuat Rangkaian

Sebelum menghubungkan setiap komponen kedalam rangkaian, lepaskan colokkan kavel USB Arduino dari komputer. Rangkailah komponen satu persatu sesuai dengan rangkaian yang ditunjukkan ppada gambar 9.4 langkah langkah membangun rangkaian :

1. Susun dan pasang terlebih dahulu Led Merah dan Led Kuning secara selang seling. Perhatikan popularitas kaki LED
2. Susun resistor 220 ohm pada breadboard seperti gambar rangkaian.
3. Gunakan kebel jumper untuk hubungkan papan arduino dengan Breadboard.
4. Setelah terpasang dengan baik, colokkan kabel USB yang terhubung dari papan Arduino ke komputer.
5. Selanjutnya, jalankan software IDE arduino dan ketikkan kode program proyek Multiple LED pada tempat penulisan sketch.
6. Verifikasi program terlebih dahulu untuk pengecekan kode error.
7. Klik icon UPLOAD pada toolbar , untuk meng-upload sketch ke papan arduino. Lihat hasilnya, perhatikan LED pada Breadboard. Selamat !! Proyek Multiple LED anda berjalan.

Proyek 6 Skalar Push Button

10.1 Kode Program

Aplikasi sederhana yang akan kita buat pada topic ini , yaitu bagaiman penggunaan satu buah skalar tekan sebagai digital input untuk mengontrol satu buah led, kemudian dilanjutkan dengan aplikasi tiga buah skalar tekan sebagai digital input yang masing-msing

mengontrol 1 LED.nah dari sini kita bisa membandingkan listing setiap kode program, kemudian kita analisis dan belajar melihat perbedaan diantara keduanya, sehingga betul-betul bisa memahami dan bisa mengembangkannya pada aplikasi-aplikasi lain yang lebih kompleks.

10.1.1 Kode program : proyek 6.1 Satu Skalar Tekan

Berikut kode program proyek satu skalar tekan untuk satu LED yang akan kita simulasi dan praktikkan :

Penjelasan Program;

1. Pada baris 1-3 menggunakan variable int atau integer,
Int LED = 13 ;
Int button = 7;
Int val =0;
Yang menginisialisasikan LED pada pin 13 dan pin 7 sebagai input skalar tekan. Kemudian variable val yang menyimpan nilai nol sebagai pernyataan pada pin input
2. Pada baris berikutnya, menggunakan fungsi void setup () yang diikuti perintah
pinMode(LED, OUTPUT);
pinMode(BUTTON, INPUT);
yang menset pada pin 13 sebagai output dan menset pin 7 sebagai input
3. Pada baris berikutnya, menggunakan fungsi void loop(), yang diikuti perintah ,
val=digitalRead(BUTTON) yang menyimpan dan membaca nilai input, kemudian dilanjutkan dengan fungsi loop if (**val == HIGH**); **{digitalWrite (LED,HIGH);** yang menyatakan jika variable val bernilai atau sama dengan HIGH, maka LED ON, jika sebaliknya LED OFF dengan kode **else{digitalWrite(LED, LOW);**

10.1.2 Kode Program : proyek 6.2 Tiga Skalar Tekan

Penjelasan Program :

1. Pada baris 1 – 2, kita membuat dua array, yaitu satu array integer untuk pin yang dihubungkan dengan input skalar tekan dan satu lagi array integer untuk pin yang dihubungkan dengan LED, dengan kode:
Int inputPins[] = {2 , 4, 6}
Int ledPins [] = { 10, 11 , 12};

Pin input/pin skalar tekan terdiri dari tiga yaitu elemen utama yang diset sama dengan 2, elemen kedua diset sama dengan 4 dan elemen ketiga diset sama dengan 6. Demikian halnya dengan pin LED terdiri dari tiga elemen, yaitu elemen pertama diset sama dengan 10, elemen kedua diset sama dengan 11, dan elemen ketiga diset sama dengan 12

2. Pada baris berikutnya, menggunakan fungsi void setup () yang diikuti perintah, `for(int index = 0; index < 3; index++)`, yang digunakan untuk menyimpan pernyataan dari seluruh array, yang panjang/element array nya sama dengan 3, kemudian dilanjutkan dengan kode, **pinMode (ledPins[indeks], OUTPUT)**; yang mendeklarasikan atau menset pin LED sebagai output, dan kode **pinMode (ledPins[indeks], INPUT)**; mendeklarasikan skalar tekan sebagai input.

Selanjutnya terdapat kode **digitalWrite (inputPins[indeks],HIGH)**; kode ini digunakan untuk mengaktifkan pull-up resistor internal pada papan arduino. pull-up resistor adalah resistor internal yang terdapat pada papan Arduino yang bisa kita gunakan pada digital input atau skalar tekan (push button switch) tanpa resistor eksternal. Anda harus pahami bahwa skalar tekan (push button switch) itu adalah tipe skalar yang hanya kontak sesaat saja ditekan dan setelah dilepas maka akan kembali lagi menjadi normal Open (NO), biasanya skalar tipe NO ini memiliki rangkaian penguncinya. Nah resistor internal inilah yang menjadi pengunci skalar tidak ditekan untuk menjamin tidak terjadi kondisi mengembang pada pin digital.

Untuk lebih jelas, perbedaan antara penggunaan resistor internal dan resistor eksternal pada saklar tekan anda bisa lihat pada prakteknya nanti disini akan tampak perbedaan antara proyek 6 - 1 dan proyek 6 - 2

3. Pada baris berikutnya, menggunakan fungsi void loop() yang diikuti perintah, **for (int index = 0; index < 3; index++)**, yang digunakan untuk menyimpan pernyataan array, dilanjutkan kode **int val = digitalRead (inputPins [index]);** untuk membaca nilai input saklar tekan, dilanjutkan pengaturan dengan kode :

```
If (val == HIGH)
```

```
    digitalWrite (ledPins [index], HIGH );
```

```
else
```

```
    digitalWrite (ledPins [index], LOW );
```

yang menyatakan jika variable val bernilai atau sama dengan HIGH, maka LED ON, jika sebaliknya maka LED OFF

10.3 PRAKTIK PROYEK SAKLAR PUSH BUTTON

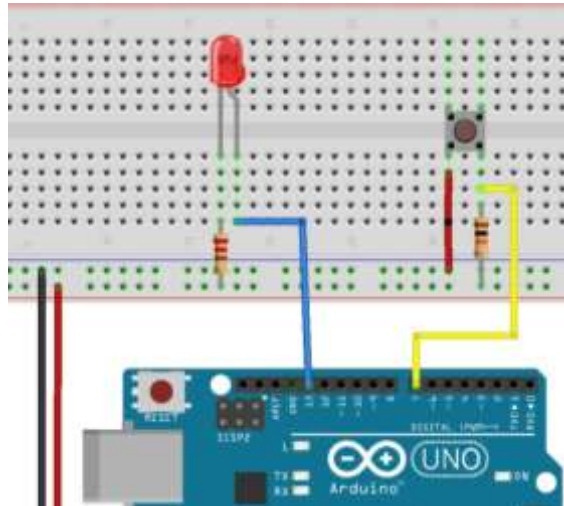
Pada penggunaan saklar tekan (push button) sebagai digital input diperlukan tambahan resistor atau resistor eksternal sebagai rangkaian pengunci untuk menjamin tidak terjadi keadaan mengambang pada pin saklar tidak di tekan. Resistor ini berfungsi sebagai pull-up atau pull-down, tergantung penempatannya. Mengapa kita membutuhkan resistor pull-up atau pull down didalam rangkaian, karena untuk menghindari masukan mengambang pada board Arduino.

Mari kita Uraikan disini, kita bisa representasikan sebuah saklar ini seperti rangkaian logika yang memberikan output ON atau OFF (bilangan 1 dan 0). Kondisi off adalah tegangan mendekati nol volt pada output, keadaan ON (atau 1) diwakili oleh tingkat yang lebih tinggi, lebih dekat dengan tegangan suplai (5 volt). Ketika saklar terbuka, tidak ada arus yang mengalir dan tidak ada tegangan yang dapat diukur pada output. Ketika anda menutup saklar, arus dapat mengalir, sehingga tegangan dapat diukur pada output.

Keadaan terbuka dapat dianggap sebagai nol dan keadaan tertutup sebagai 1 dalam rangkaian logika. Dalam rangkaian logika. Dalam rangkaian logika, tegangan 5 volt mewakili logika 1, dan tegangan 0 volt mewakili logika 0. Oleh karena itu, penggunaan resistor eksternal pull up atau pull down pada input digital dari sebuah saklar tekan bertujuan untuk menjamin bahwa rangkaian tidak dalam keadaan mengambang, yaitu keadaan acak yang bisa berlogika 1 atau HIGH bisa juga berlogika 0 atau LOW.

Pull Up Resistor

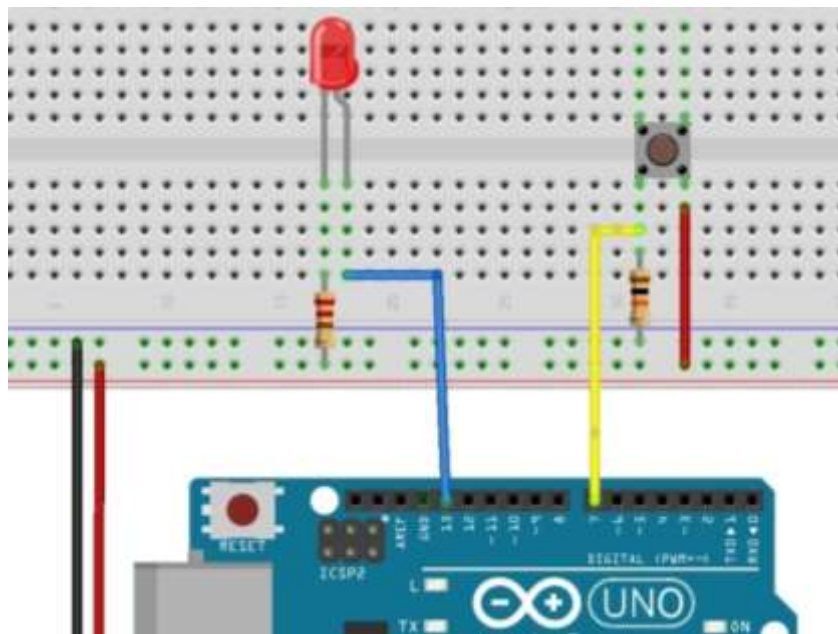
Perhatikan gambar 10.7, resistor pull up pada rangkaian saklar disini berarti kita mengfungsikan input mikrokontroler Arduino supaya secara default mendapat logika LOW, ketika mendapat trigger (saat saklar di tekan) maka akan berubah menjadi logika HIGH. Berdasarkan gambar, pin Arduino dihubungkan dengan GND (ground) melalui saklar tekan. Ketika tombol ditekan, pin akan menjadi LOW, tetapi pada saat dilepas maka kondisi akan terhubung dengan +5v melalui resistor dan pin menjadi HIGH



Gambar 10.7 Pull Up

Coba bayangkan jika tidak terdapat resistor , menyebabkan kondisi pin akan mengambang (float), hal ini kadang bisa menimbulkan kesalahan. Nah , disinilah pentingnya fungsi resistor untuk menjamin tidak terjadinya keadaan mengambang. Rangkaian pull up dan pull down umumnya digunakan resistor 1-k Ω .

Pull Down Resistor



Gambar 10.8 Pull Down

Perhatikan gambar 10.8 resistor pull down pada rangkaian saklar disini menggunakan input mikrokontroler Arduino supaya secara default mendapat logika HIGH , saat saklar

ditekan maka akan berubah menjadi logika LOW. Pin Arduino dihubungkan dengan +5v melalui saklar tekan. Ketika saklar ditekan, pin akan menjadi HIGH dan resistor dihubungkan dengan ground agar pada saat dilepas ini menjadi LOW.

Pull up Resistor Internal Arduino

Pada papan Arduino memiliki resistor pull up internal dari 20 -50 k Ω dan untuk menggunakan perlu diaktifkan kedalam program sketch. Jika diaktifkan, rangkaian saklar pada papan arduino seperti gambar 10.9

Untuk mengaktifkan resistor pull up internal pada pin, kita harus terlebih dahulu mengubah pin mode menjadi pin untuk INPUT dan digital Write menjadi HIGH, sebagai berikut :

```
pinMode (pin, INPUT);
```

```
digitalWrite (pin, HIGH);
```

Jika anda mengubah pin Mode dari INPUT ke OUTPUT setelah mengaktifkan resistor pull up internal, pin akan tetap dalam keadaan HIGH, demikian juga sebaliknya

10.3.1 Praktik Proyek 6 – 1 : satu saklar tekan

Bahan dan Komponen

Daftar komponen yang kita gunakan, yaitu :

1 Board Arduino Uno-R3

1 LED Merah

1 Breadboard

1 Resistor 220 Ω

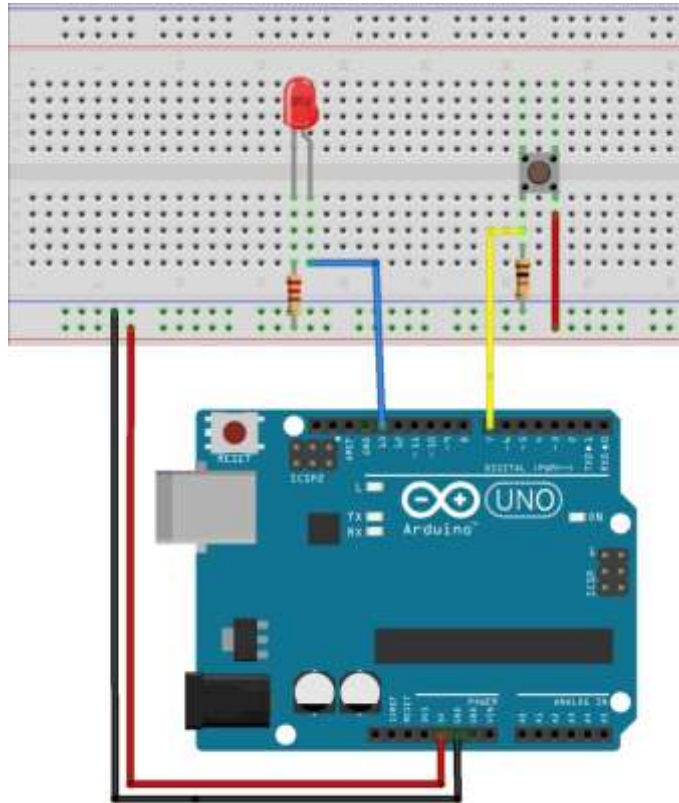
1 Resistor 10 k Ω

1 Saklar push button

→ Kabel Jumper

Membuat Rangkaian

Sebelum menghubungkan setiap komponen kedalam rangkaian, lepaskan colokan kabel USB Arduino dari komputer. Rangkailah komponen satu persatu sesuai dengan rangkaian yang ditunjukkan pada gambar 10.10



Gambar 10.10 Rangkaian Nyala LED dengan Button

pada rangkaian ini kita gunakan resistor eksternal hubungan pull up resistor, langkah-langkah membuat rangkaian yaitu :

1. Susun resistor 220 ohm dan led secara seri pada breadboard seperti pada rangkaian.
2. Pasang saklar tekan dan Resistor 10 k Ω
3. Gunakan kabel jumper untuk menghubungkan papan Arduino dengan Breadboard.
4. Setelah semua terpasang baik, colokkan kabel USB yang terhubung dari papan arduino ke Komputer.
5. Selanjutnya, jalankan software IDE Arduino dan ketikkan kode program proyek satu saklar tekan pada tempat penulisan sketch. Selain itu , lakukan verifikasi program untuk pengecekan error pada kode program .
6. Klik ikon Upload pada toolbar, untuk meng-upload sketch ke papan Arduino. Perhatikan LED pada Breadboard ketika tombol saklar ditekan atau dilepas.

10.3.2 Praktik Proyek 6 – 2 : Tiga Saklar Tekan

Pada rangkain ini tidak menggunakan resistor eksternal pada saklar tekan , karena pada kode program/sketch kita sudah aktifkan pull-up resistor internal pada arduino

Bahan dan Komponen

Daftar komponen yang kita gunakan, yaitu :

1 Board Arduino Uno-R3

1 LED Merah

1 LED Kuning

1 LED Hijau

1 Breadboard

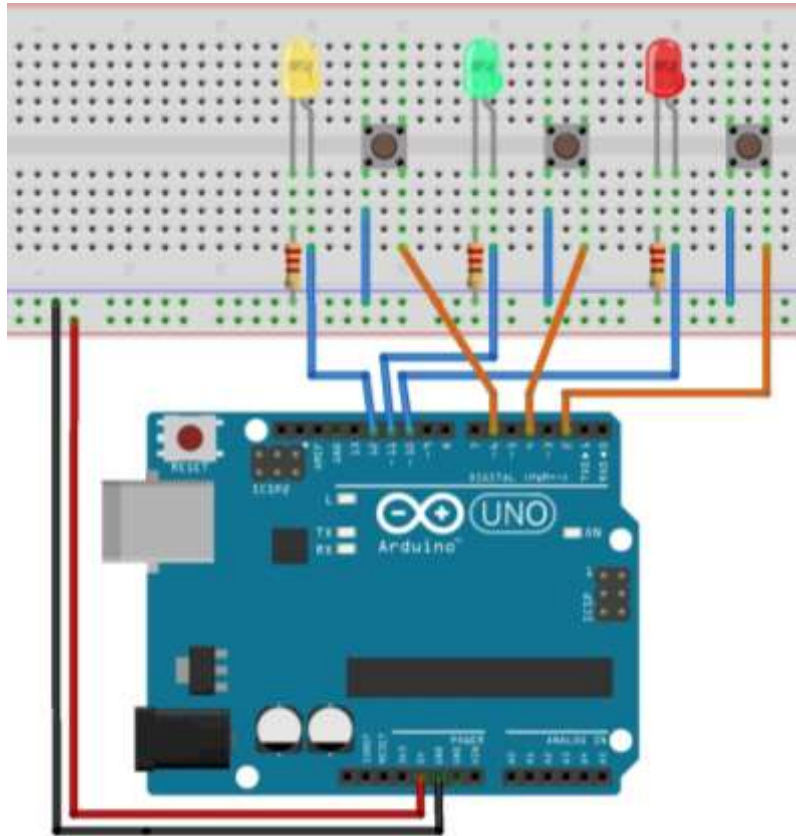
3 Resistor 220 Ω

3 Saklar push button

→ Kabel Jumper

Membuat Rangkaian

Sebelum menghubungkan setiap komponen kedalam rangkaian, lepaskan colokan kabel USB Arduino dari komputer. Rangkailah komponen satu persatu sesuai dengan rangkaian yang ditunjukkan pada gambar 10.12.



Langkah-langkah membangun rangkaian :

1. Susun semua resistor 220 ohm dan LED secara seri pada Breadboard seperti gambar rangkaian
2. Pasang 3 saklar tekan , kemudian gunakan kabel jumper untuk menghubungkan papan Arduino dengan Breadboard.
3. Setelah semua terpasang dengan baik, colokkan kabel USB yang terhubung dari papan Arduino ke komputer.
4. Selanjutnya, jalankan software IDE Arduino ketikkan kode program proyek tiga Saklar tekan pada tempat penulisan sketch.
Lakukan verifikasi program sebelum meng-upload ke papan Arduino
5. Klik ikon Upload pada toolbar, untuk meng-upload sketch ke papan Arduino. Perhatikan LED pada Breadboard ketika tombol saklar ditekan atau dilepas.

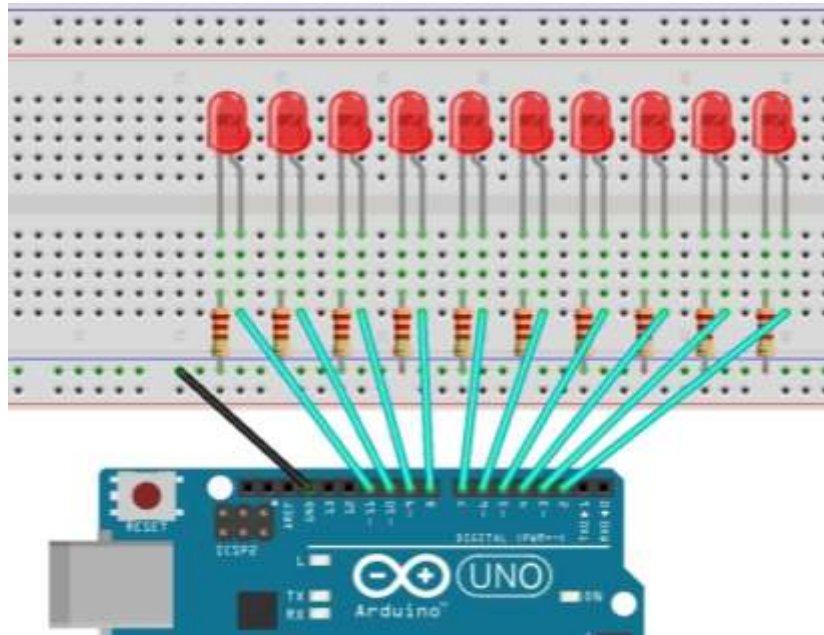
Proyek 7 LED Bar Graph

11.1 Kode Program

Aplikasi sederhana yang kita buat pada topic ini , yaitu bagaimana penggunaan komponen potensiometer sebagai analog input untuk memberikan keluaran tegangan antara 0- 5 volts kemudian menggunakan tegangan keluaran ini sebagai sebuah variable didalam program LED Bar Graph. Kita ketahui bahwa Arduino Uno mempunyai 6 pin analog yang dapat membaca input berupa voltase (dari 0 – 5 volts) dan mengonversikan keangka digital antara 0 (0 volt) dan 1023 (5 volts) , yaitu pembacaan dengan resolusi 10 bit. Analog

Potensiometer sebagai input berbeda dengan input digital yang hanya dapat membaca dua kondisi, yaitu 0 atau low atau tidak ada tegangan dan tegangan 1 atau HIGH atau tegangan 5v. input analog dapat membaca angka digital hingga 1023. Di mana tegangan analog 5v setara 1023 , setiap bagian ekuivalen 0,00488 v. setara dengan 1 , tegangan input analog 0,00977 v setara dengan 2, . . . dan seterusnya tegangan input analog 4.99512v adalah setara dengan 1023. Jadi pada dasarnya input , adalah skala 0 – 5 v mengkonversikannya ke angka digital (0 – 1023)

LED Bar Graph yang kita buat pada proyek ini adalah barisan 10 LED yang disusun dimana LED akan menyala proposional dengan posisi potensiometer. Dengan kata lain, potensiometer digunakan untuk mengendalikan serangkaian output digital yang tergantung pada tegangan input analog. Bar Graph sering kita lihat pada peralatan audio. Berikut adalah kode program LED Bar Graph.



Penjelasan Program

1. Pada baris 1-2 menggunakan variable int atau integer
`Const int analogpin = 0 ;`
`Const int ledCount = 10;`
Yang menginisialisasikan analog input pada pin 0 (pin tempat potensiometer dipasang) dan ini sialisasi panjang atau elemen array jumlah LED sebanyak 10 elemen
2. Pada baris 3, kita membuat atau mendeklarasikan sebuah array dari integer dengan 10 elemen dengan kode `int ledPins[] = {2 , 3 , 4 , 5 , 6 , 7 , 8 , 9 , 10 , 11 };` Elemen pertama diset sama dengan 2 , Elemen pertama diset sama dengan 3 , Elemen pertama diset sama dengan 4 , dan seterusnya
3. Pada baris ke 4 menggunakan fungsi void setup (), diikuti kode, `for (int this Pin = 0; thisPin < pinCount;thisPin++)` yang berarti menggunakan for loop untuk mengakses 10 elemen yang telah diinisialisasi (pincount) tadi. Di mana loop dibuat dengan variable thisPin dengan nilai start/awal sama dengan 0 dan berakhir dengan 10. Kemudian pada baris ke 5 menggunakan kode `pinMode (ledPins[thisPin] , OUTPUT);` yang mendeklarasikan semua pin sebagai output.
4. Pada baris berikutnya menggunakan fungsi void loop () yang diikuti kode, `int sensorReading = analogRead(analogPin);` untuk membaca analog input potensiometer. Dilanjutkan dengan kode , `int ledLevel = map(sensorReading, 0 , 1023, 0 ledCount);` yang memetakan atau penskalaan hasil pembacaan analog

input dari nilai range range ke range tinggi (0 – 1023) menjadi nilai output (0 – 10) kemudian dilanjutkan dengan kode loop for

```
for (int thisLed = 0; thisLed < ledCount;
thisLed++)
    if (thisLed < ledLevel)
        digitalWrite (ledPins [thisLed]. HIGH);
    else
        digitalWrite (ledPins [thisLed]. LOW);
```

yang dinyatakan jika elemen pada array lebih kecil dari ledLevel , maka ON kan LED, jika sebaliknya OFF kan LED.

Proyek 8 SENSOR CAHAYA

12.1 KODE PROGRAM

Pada proyek input analog sebelumnya kita telah menggunakan potensiometer untuk mendapatkan tegangan input analog 0 – 5 volt. Kali ini kita akan menggunakan sensor cahaya LDR (light dependant resistor) untuk mendapatkan tegangan input analog yang nilainya tergantung intensitas cahaya yang diterima sensor.

Aplikasi sederhana dan praktis yang akan kita buat pada topic ini, yaitu bagaimana memahami pemanfaatan sensor cahaya LDR untuk menyalakan LED bila tidak mendapat cahaya atau kondisi lingkungan cukup terang. Sebagaimana kita ketahui bahwa LDR (light dependant resistor) merupakan komponen resistor yang nilai hambatannya bisa berubah karena intensitas cahaya. Jika intensitas cahaya yang mengenainya cukup terang, nilai hambatannya sekitar 1Ω dan bila LDR diletakkan pada tempat yang gelap, nilai hambatannya mencapai $10\text{ M}\Omega$. prinsip nya sama saja dengan potensiometer , namun LDR hambatannya tergantung intensitas cahaya , berikut ini kode program sensor cahaya LDR :

Penjelasan Program

1. Pada baris 1-4 kita membuat variable integer dengan kode ;

```
Const int pinLDR = 0
```

```
Const int pinLed = 8;
```

```
Int sensor VaLue = 0;
```

```
Int outputValue = 0;
```

Inisialisasi sensor LDR untuk dihubungkan pada pin analog 0 Arduino , berikutnya inisialisasi LED pada pin digital 8 , selanjutnya penskalaan nilai baca/nilai input sensor sma dengan 0 dan nilai output sama dengan 0

2. Pada baris ke 5 mrnggunakan fungsi void setup (), fungsi yang harus ada pemrograman Arduino , meskipun tidak ada perintah di antara dalam kurung, fungsi ini harus tetap ditulis
3. Pada baris berikutnya , menggunakan fungsi void loop () yang diikuti perintah ,
sensorValue = analodRead (pinLDR) ; untuk membaca nilai sensor , berikutnya kode output value = map(sensor value, 0 , 1023, 0 , 500): untuk melakukan pemetaan atau penskalaan nilai sensor (0- 1023) menjadi output (0-500) . berikut dilanjutkan struktur pengaturan dengan kode,
If (outputValue <=300)

```
    digitalWrite (pinLed, HIGH);
```

```
else
```

```
    digitalWrite (pinLed, LOW);
```

```
    delay (200);
```

yang menyatakan nilai output (variable output value) lebih kecil atau sama dengan 300, maka deteksi kondisi lingkungan yang diterima sensor adalah gelap/redup, dengan demikian nyalakan LED, jika sebaliknya padamkan LED. Angka 300 di atas dpat diubah-ubah disesuaikan dengan kondii cahaya lingkungan, juga tergantung karakteristik sensor LDR yang kita gunakan (cara membaca nilai snsor via komunikasi serial dijelaskan pada sub bab 12.3), semakin kecil nilai output value maka semakin gelapintensitas cahaya yang diterima sensor dan demikian juga sebaliknya. Sementara kode delay (200) ; menyatakan tunda selama 200 milisekon sebelum loop berikutnya agar analog to digital converter siap membaca data berikutnya.

12.3 PRAKTIK PROYEK SENSOR CAHAYA

Bahan dan Komponen

Daftar komponen yang kita gunakan, yaitu :

1 Board Arduino Uno-R3

1 LED Merah

1 Breadboard

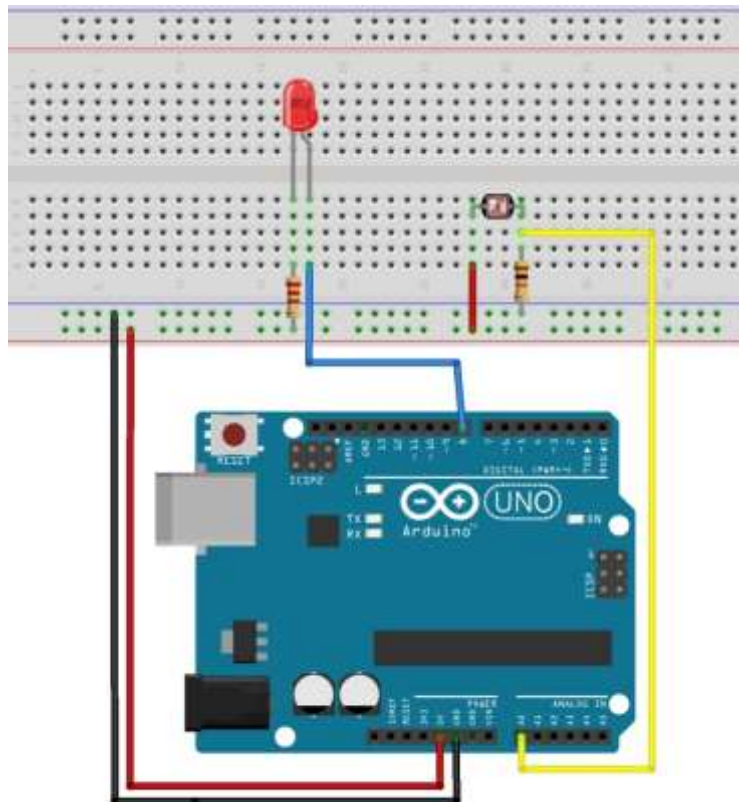
1 Resistor 220 Ω

1 Resistor 10 k Ω

→ Kabel Jumper

Membuat Rangkaian

Sebelum menghubungkan setiap komponen kedalam rangkaian, lepaskan colokan kabel USB Arduino dari komputer. Rangkailah komponen satu persatu sesuai dengan rangkaian yang ditunjukkan pada gambar 12.4



Langkah-langkah membangun rangkaian :

1. Pasang LED terlebih dahulu seperti gambar rangkaian. Perhatikan polaritas kaki LED
2. Susun seri resistor 220 ohm pada kaki LED dan pasang seri resistor 10 k Ω pada sensor LDR
3. Gunakan kabel jumper untuk menghubungkan pada Arduino dengan Breadboard.

4. Setelah semua terpasang dengan baik, colokkan kabel USB yang terhubung dari papan Arduino ke komputer.
5. Selanjutnya, jalankan software IDE arduino dan kerikkan lode proyek sensor cahaya pada tempat penulisan sketch. Verifikasi kode program sebelum upload ke papan arduino
6. Klik ikon Upload pada toolbar, untuk mengupload sketch ke papan arduino . lihat hasilnya , tutup permukaan sensor LDR dengan tangan anda, perhatikan Led akan menyala , kemudian lepaskan tangan anda LED akan padam,

Jika led tidak menyala pada saat permukaan sensor / lingkungan gelap, maka nilai setting sensor ≤ 300 tidak sesuai. Oleh karena itu, angka ini harus anda ubah menjadi lebih kecil sampai dicapai nilai yang tepat. Untuk mengetahui pembacaan nilai sensor dapat dilakukan dengan cara berkomunikasi serial antara papan Arduino dengan komputer. Pada software IDE Arduino terdapat serial monitor untuk melihat interaksi dari proses yang terjadi pada papan Arduino dan ditampilkan pada layar komputer melalui serial monitor.

Langka-langkah untuk melihat pemrosesan pada papan arduino dalam hal ini pembacaan nilai sensor pada pin input , kita haru menambahkan kode serial. Begin (9600), dibawah fungsi void setup (). Kode serial. Begin (9600), merupakan kode untuk membuka port komunikasi serial untuk mengirim data kembali ke komputer dengan kecepatan 9600 bit perdetik. Selanjutnya pada void loop () , kita juga menambahkan kode serial.println(), untuk selanjutnya ditampilkan pada serial monitor software Ide Arduino.

Dengan demikian kode program sensor cahaya diatas kita ubah/tambahkan menjadi

Nah , untuk melihat pembacaan nilai sensor , upload perubahan kode program diatas ke papan Arduino, kemudian klik ikon serial monitor pada software Ide arduino

Selanjutnya akan muncul jendela serial monitor yang menampilkan pembacaan nilai sensor pada pin input. Berikut hasil pembacaan sensor cahaya saat kondisi lingkungan terang atau siang hari

Untuk melihat nilai pembacaan sensor pada kondisi gelap, tutup permukaan sensor cahaya, kemudan lihat hasilnya pada jendela serial monitor seperti pada gambar 12.8

Berdasarkan hasil pembacaan diatas ,kita sudah bisa mensetting nilai sensor secara tepat atau mendekati nilai kepekaanya , disini saya setting nilai sensor dengan mengambil nilai

rata-rata terkecil pada kondisi lingkup gelap yakni sebesar ≤ 300 . Sebenarnya semakin kecil nilai setting ini, semakin gelap kondisi lingkungan yang dibaca sensor untuk mengaktifkan atau menyalakan LED

Proyek 9 Kendali Motor Servo

13.1 KODE PROGRAM

Tidak seperti motor DC umumnya, dengan motor servo kita dapat mengatur posisi poros (sudut) secara spesifik . pengaturamn motor servo ini sangat berguna untuk mengendalikan robot lengan. Control pesawat tak berawak , mainan mobil remot control atau objek yang ingin kita pindahkan pada sudut tertentu.

Contoh program servo yang akan kita simulasikan dan praktikkan disini adalah contoh dasar yang ada pada library Arduino .

Motor servo dapat diklasifikasikan menurut ukuran atau torsi. Ada servo mini, servos standar, dan servo besar. Motor servo ukuran mini dan standar bisaanya dapat dihubungkan langsung pada Arduino tanpa perlu catu dayaeksternal ataupun driver. Berikut kode program pengendalian posis motor servo

Penjelasan Program

1. Pada baris pertama kode `#include <servo.h>` merupakan kode untuk memanggil atau menggunakan library yang ada pada Arduino
2. Pada baris ke 2 kita membuat objek servo dengan nama `myservo`
3. Pada baris ketiga kita membuat variable integer dengan nama "pos" dengan nilai variable sama dengan 0 (posisi minimum = 0 derajat) yang menyimpan posisi awal servo
4. Pada bari ke 4 fungsi void `setup ()` yang diikuti lode `myservo.attach(9);` kode ini berarti kita akan memasang objek servo pada pin 9
5. Pada baris ke 5 fungsi void `loop ()` yang diikuti kode

```
For (pos = 0; pas < 180 ; pos += 1)  
  Myservo . write (pos) ;  
  Delay (15);
```

Kode diatas merupakan fungsi for loop yang menyatakan bahwa motor servo akan berputar dari posisi minimum 0 derajat ke posisi maksimum 180 derajat dengan langkah/step tiap 1 derajat. Kemudian `myservo.write(pos);` berarti pin

mengeluarkan outut/posisi motor sesuai dengan variable "pos". kemudian tunda 15 ms di antara setiap perintah /kode servo. Selanjutnya untuk putaran/posisi motor servo dari 180 derajat keposisi 0 derajat kita gunakan kode berikut

```
For ( pos =180 ; pos >=1 ; pos -=1)
```

```
My servo.write (pos);
```

```
Delay (15);
```

Demikian seterusnya akan berulang , posisi motor dari 0 derajat ke posisi 180 derajat dengan lengkap/step setiap 1 derajat kemudian sebaliknya dari 180 derajat ke 0 derajat

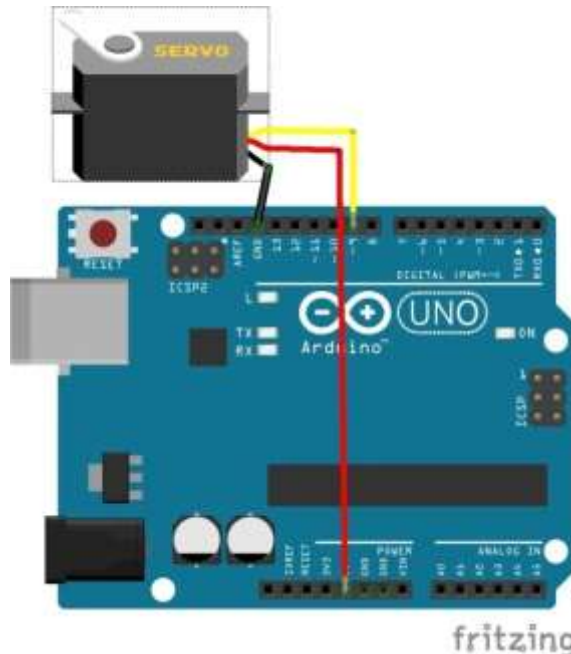
13.3 PRAKTIK PROYEK KENDALI MOTOR SERVO

1 board Arduino Uno R3

1 mikro servo

Membuat rangkaian

Sebelum menghubungkan setiap komponen ke dala rangkaian, lepaskan colokan kabel USB Arduino kekomputer. Rangkailah komponen satu persatu sesuai dengan rangkaian yang ditnjukkan pada gambar 13.3



Langkah-langkah membuat rangkaian:

1. Hubungkanlah kabel merah dari motor servo ke pin 5 V papan Arduino
2. Hubungkan kabel hitam dari motor servo ke pin ground papan Arduino
3. Hubungkanlah kabel kuning dari motor servo ke pin 9 V papan Arduino

4. Setelah semua terpasang dengan baik, colokkan kabel USB yang terhubung dari papan Arduino ke komputer
5. Selanjutnya jalankan software IDE arduino buka file→example→servo→sweep
6. Klik ikon Upload pada toolbar untuk meng-upload sketch ke papan arduino. Lihat hasilnya, perhatikan putaran servo

PRAKTIK MEMBANGUN ROBOT BERODA

MENGGUNAKAN MOTOR DC DAN DRIVER L298

1. Praktik manuver robot:

```
int motorKiri1 = 5;
int motorKiri2 = 4;
int motorKanan3 = 3;
int motorKanan4 = 2;

void setup() {
  pinMode(motorKiri1, OUTPUT);
  pinMode(motorKiri2, OUTPUT);
  pinMode(motorKanan3, OUTPUT);
  pinMode(motorKanan4, OUTPUT);
}

void loop()
{
  maju();
  berhenti();
  mundur();
  berhenti();
  belok_kiri();
  berhenti();
  belok_kanan();
  berhenti();
  putar_kiri();
  berhenti();
  putar_kanan();
}

void berhenti()
{
  // motor berhenti
  digitalWrite(motorKiri1, LOW);
  digitalWrite(motorKiri2, LOW);
  digitalWrite(motorKanan3, LOW);
  digitalWrite(motorKanan4, LOW);
  delay(200);
}

void maju()
{
  // motor maju
  digitalWrite(motorKiri1, LOW);
  digitalWrite(motorKiri2, HIGH);
```

```
digitalWrite(motorKanan3, LOW);  
digitalWrite(motorKanan4, HIGH);  
delay(200);  
}
```

```
void mundur()  
{  
  // motor maju  
  digitalWrite(motorKiri1, HIGH);  
  digitalWrite(motorKiri2, LOW);  
  digitalWrite(motorKanan3, HIGH);  
  digitalWrite(motorKanan4, LOW);  
  delay(200);  
}
```

```
void belok_kiri()  
{  
  // motor maju  
  digitalWrite(motorKiri1, LOW);  
  digitalWrite(motorKiri2, LOW);  
  digitalWrite(motorKanan3, LOW);  
  digitalWrite(motorKanan4, HIGH);  
  delay(200);  
}
```

```
void belok_kanan()  
{  
  // motor maju  
  digitalWrite(motorKiri1, LOW);  
  digitalWrite(motorKiri2, HIGH);  
  digitalWrite(motorKanan3, LOW);  
  digitalWrite(motorKanan4, LOW);  
  delay(200);  
}
```

```
void putar_kiri()  
{  
  // motor maju  
  digitalWrite(motorKiri1, HIGH);  
  digitalWrite(motorKiri2, LOW);  
  digitalWrite(motorKanan3, LOW);  
  digitalWrite(motorKanan4, HIGH);  
  delay(500);  
}
```

```
void putar_kanan()  
{  
  // motor maju  
  digitalWrite(motorKiri1, LOW);  
  digitalWrite(motorKiri2, HIGH);  
  digitalWrite(motorKanan3, HIGH);  
  digitalWrite(motorKanan4, LOW);  
  delay(500);  
}
```

2. OBSTACLE AVOIDER (jaga jarak) dengan sensor PING (ultrasonic)

```
int motorKiri1 = 5;
int motorKiri2 = 4;
int motorKanan3 = 3;
int motorKanan4 = 2;
int pingTriger = 9;
int pingEcho = 8;

void setup(){
  Serial.begin(9600);
  pinMode(motorKiri1,OUTPUT);
  pinMode(motorKiri2,OUTPUT);
  pinMode(motorKanan3,OUTPUT);
  pinMode(motorKanan4,OUTPUT);
}

void loop() {
  long duration, cm;
  pinMode(pingTriger, OUTPUT);
  digitalWrite(pingTriger, LOW);
  delayMicroseconds(2);
  digitalWrite(pingTriger, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingTriger, LOW);
  pinMode(pingEcho, INPUT);
  duration = pulseIn(pingEcho, HIGH);
  cm = microsecondsToCentimeters(duration);

  if(cm > 50 )
  {
    Serial.println(cm); //maju
    digitalWrite(motorKiri1,HIGH);
    digitalWrite(motorKiri2,HIGH);
    digitalWrite(motorKanan3,HIGH);
    digitalWrite(motorKanan4,HIGH);
  }
  if(cm > 30 && cm < 50 )
  {
    Serial.println(cm); //berhenti
    digitalWrite(motorKiri1,LOW);
    digitalWrite(motorKiri2,LOW);
    digitalWrite(motorKanan3,LOW);
    digitalWrite(motorKanan4,LOW);
  }
  else{
    Serial.println(cm); //mundur
```

```

    digitalWrite(motorKiri1,HIGH);
    digitalWrite(motorKiri2,LOW);
    digitalWrite(motorKanan3,HIGH);
    digitalWrite(motorKanan4,LOW);
}
}
long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}

```

3. OBSTACLE AVOIDER AUTONOMOUS

(Perlu tambahan library NewPing dan Servo)

```

#include <NewPing.h>
#include <Servo.h>
int motorKiri1 = 5;
int motorKiri2 = 4;
int motorKanan3 = 3;
int motorKanan4 = 2;

const int dangerThresh = 30; //jarak minimal obstacles (cm)

Servo myservo;
long duration;
#define TRIGGER_PIN 9
#define ECHO_PIN 8
#define MAX_DISTANCE 100

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
unsigned int time;
int jarak;
int jarakDepan;
int jarakKiri;
int jarakKanan;

void setup()
{
    pinMode(motorKiri1,OUTPUT);
    pinMode(motorKiri2,OUTPUT);
    pinMode(motorKanan3,OUTPUT);
    pinMode(motorKanan4,OUTPUT);
    myservo.attach(6);
}

void maju(){
    digitalWrite (motorKiri1, HIGH);
    digitalWrite (motorKiri2, LOW);
    digitalWrite (motorKanan3, HIGH);
    digitalWrite (motorKanan4, LOW);
}

```



```
void mundur(){
  digitalWrite (motorKiri1 , LOW);
  digitalWrite (motorKiri2, HIGH);
  digitalWrite (motorKanan3, LOW);
  digitalWrite (motorKanan4, HIGH);
}

void belok_kiri(){
  digitalWrite (motorKiri1, LOW);
  digitalWrite (motorKiri2, HIGH);
  digitalWrite (motorKanan3, HIGH);
  digitalWrite (motorKanan4, LOW);
  delay(200);
}

void belok_kanan(){
  digitalWrite (motorKiri1, HIGH);
  digitalWrite (motorKiri2, LOW);
  digitalWrite (motorKanan3, LOW);
  digitalWrite (motorKanan4, HIGH);
  delay(200);
}

void berhenti(){
  digitalWrite (motorKiri1 ,LOW);
  digitalWrite (motorKiri2, LOW);
  digitalWrite (motorKanan3, LOW);
  digitalWrite (motorKanan4, LOW);
}

void bandingkanJarak()
{
  if (jarakKanan > jarakKiri && jarakKanan > jarakDepan) //kiri nabrak
  {
    belok_kanan();
    delay(200);
  }
  if (jarakKiri > jarakKanan && jarakKiri > jarakDepan) //kanan nabrak
  {
    belok_kiri();
    delay(200);
  }
  else if (jarakDepan > jarakKanan && jarakDepan > jarakKiri)
  {
    maju();
  }
}

void scan(){
```

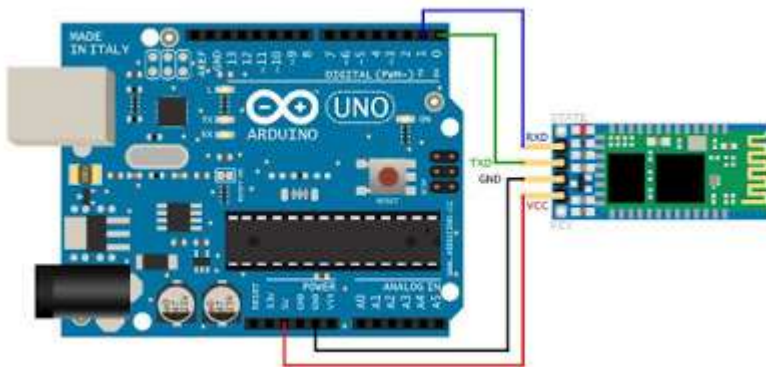
```

    time = sonar.ping();           //mengirim ping dan menyimpan sebagai
time
    jarak = time / US_ROUNDTRIP_CM; //ubah waktu menjadi jarak
    if(jarak == 0){                //tidak ada sinyal ping
        jarak = 100;              //anggap jarak maksimal
    }
}

void loop()
{
    myservo.write(90);
    scan();
    jarakDepan = jarak;
    delay(500);
    if (jarakDepan > dangerThresh) maju();
    else berhenti();
    myservo.write(0);
    scan();
    delay(700);
    jarakKiri = jarak;
    delay(500);
    myservo.write(180);
    scan();
    delay(700);
    jarakKanan = jarak;
    delay(500);
    myservo.write(90);
    bandingkanJarak();
}

```

PRAKTIK KONEKSI BLUETOOTH



1. Control LED dengan Bluetooth

```

int led=13;
int DataBluetooth;

```

```

void setup() {
  Serial.begin(9600);
  Serial.println("Menyalakan dan mematikan led dengan bluetooth");
  Serial.println("1 : ON, 0 = Off");
  pinMode(led,OUTPUT);
}

void loop() {
  if (Serial.available()){
    DataBluetooth=Serial.read();
    if(DataBluetooth=='1'){
      digitalWrite(led,HIGH);
      Serial.println("LED On");
    }
    if (DataBluetooth=='0'){
      digitalWrite(led,LOW);
      Serial.println("LED Off");
    }
  }
  delay(100);
}

```

2. Praktik ROBOT dengan BLUETOOTH dan ANDROID

```

int motorKiri1 = 5; // pin 2 on L293D IC
int motorKiri2 = 4; // pin 7 on L293D IC
int motor1EnablePin = 6; // pin 1 on L293D IC
int motorKanan3 = 3; // pin 10 on L293D IC
int motorKanan4 = 2; // pin 15 on L293D IC
int motor2EnablePin = 11; // pin 9 on L293D IC
int state;
int Speed = 100;
int flag=0; //makes sure that the serial only prints once the state
int stateberhenti=0;
void setup() {
  // sets the pins as outputs:
  pinMode(motorKiri1, OUTPUT);
  pinMode(motorKiri2, OUTPUT);
  pinMode(motor1EnablePin, OUTPUT);
  pinMode(motorKanan3, OUTPUT);
  pinMode(motorKanan4, OUTPUT);
  pinMode(motor2EnablePin, OUTPUT);
  // sets enable1Pin and enable2Pin high so that motor can turn on:
  //digitalWrite(Motor1EnablePin, HIGH);
  //digitalWrite(Motor2EnablePin, HIGH);
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

```

```
void loop() {
  //membaca data, dan menyimpan sbg state
  if(Serial.available() > 0){
    state = Serial.read();
    if(state > 5){
      Speed = state;}
    flag=0;
  }
  // jika '1' motor maju
  if (state == 1) {
    maju();
    if(flag == 0){
      Serial.println("maju!");
      flag=1;
    }
  }

  // jika '2' motor belok kiri
  else if (state == 2) {
    belok_kiri();
    if(flag == 0){
      Serial.println("belok kiri");
      flag=1;
    }
    delay(1500);
    state=3;
    stateberhenti=1;
  }
  // jika '3' motor berhenti
  else if (state == 3 || stateberhenti == 1) {
    berhenti();
    if(flag == 0){
      Serial.println("berhenti!");
      flag=1;
    }
    stateberhenti=0;
  }
  // jika '4' motor kanan
  else if (state == 4) {
    belok_kanan();
    if(flag == 0){
      Serial.println("belok kanan");
      flag=1;
    }
    delay(1500);
    state=3;
    stateberhenti=1;
  }
  // jika '5' motor mundur
  else if (state == 5) {
```

```
    mundur();
    if(flag == 0){
        Serial.println("Mundur!");
        flag=1;
    }
}
//For debugging purpose
//Serial.println(state);
}

void maju(){
    digitalWrite(motorKiri1, HIGH);
    digitalWrite(motorKiri2, LOW);
    digitalWrite(motorKanan3, LOW);
    digitalWrite(motorKanan4, HIGH);
    analogWrite(motor1EnablePin, Speed);
    analogWrite(motor2EnablePin, Speed);
    //Serial.println("maju!");
}

void mundur(){
    digitalWrite(motorKiri1, LOW);
    digitalWrite(motorKiri2, HIGH);
    digitalWrite(motorKanan3, HIGH);
    digitalWrite(motorKanan4, LOW);
    analogWrite(motor1EnablePin, Speed);
    analogWrite(motor2EnablePin, Speed);
    //Serial.println("mundur!");
}

void belok_kanan(){
    digitalWrite(motorKiri1, LOW);
    digitalWrite(motorKiri2, LOW);
    digitalWrite(motorKanan3, LOW);
    digitalWrite(motorKanan4, HIGH);
    analogWrite(motor1EnablePin, Speed);
    analogWrite(motor2EnablePin, Speed);
    //Serial.println("belok kanan");
}

void belok_kiri(){
    digitalWrite(motorKiri1, HIGH);
    digitalWrite(motorKiri2, LOW);
    digitalWrite(motorKanan3, LOW);
    digitalWrite(motorKanan4, LOW);
    analogWrite(motor1EnablePin, Speed);
    analogWrite(motor2EnablePin, Speed);
    //Serial.println("belok kiri");
}
```

```
}  
  
void berhenti(){  
    digitalWrite(motorKiri1, LOW);  
    digitalWrite(motorKiri2, LOW);  
    digitalWrite(motorKanan3, LOW);  
    digitalWrite(motorKanan4, LOW);  
    analogWrite(motor1EnablePin, Speed);  
    analogWrite(motor2EnablePin, Speed);  
    //Serial.println("berhenti");  
}
```