

LAPORAN PROJECT
SISTEM MIKROPROSESOR
ARDUINO IDE



Disusun oleh :

Nurkholis Alfian (04161053)

Toga Clinton Sihotang (04161073)

INSTITUT TEKNOLOGI KALIMANTAN

BAB I

LANDASAN TEORI

A. Arduino

Arduino merupakan rangkaian mikrokontroler dalam satu papan yang bersifat open source. Namun untuk memahami Arduino, terlebih dahulu harus memahami apa yang dimaksud dengan *physical computing*. *Physical computing* adalah pembuatan sebuah sistem atau perangkat fisik dengan menggunakan software dan hardware yang sifatnya interaktif yaitu dapat menerima rangsangan dari lingkungan dan merespon balik. *Physical computing* adalah sebuah konsep untuk memahami hubungan yang manusiawi antara lingkungan yang sifat alaminya adalah analog dengan dunia digital. Pada prakteknya konsep ini diaplikasikan dalam desain alat atau proyek-proyek yang menggunakan sensor dan *microcontroller* untuk menerjemahkan input analog ke dalam sistem software serta mengontrol gerakan alat-alat elektro-mekanik seperti lampu, motor dan sebagainya. Arduino adalah kombinasi dari hardware, bahasa pemrograman dan Integrated Development Environment (IDE) yang canggih.

B. IDE (Integrated Development Environment)

IDE singkatan dari Integrated Development Environment merupakan program komputer sebagai lingkungan pengembangan aplikasi atau program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. Menjadi seorang programmer diharuskan untuk mengetahui berbagai IDE untuk pembangunan perangkat lunak. Sekarang telah banyak IDE yang sudah digunakan secara bebas, seperti MS Visual Basic untuk Basic, MS Visual C++ untuk C, Borland Delphi untuk Delphi, Eclipse dan NetBeans untuk Java, Turbo Pascal untuk Pascal, dan masih banyak yang lainnya.

Biasanya sebuah IDE, atau secara bebas dapat diterjemahkan sebagai Lingkungan Pengembangan Terpadu, setidaknya memiliki fasilitas:

- Editor, fasilitas ini gunanya untuk menuliskan kode atau fungsi-fungsi yang dapat diterjemahkan oleh mesin komputer sebagai kode programming.
- Compiler, gunanya untuk menjalankan bentuk kode program yang dituliskan di Editor dan mengubah dalam bentuk binary yang kemudian ditampilkan sesuai perintah-perintah kode mesin.
- Debugger, gunanya untuk mengecek dan mengetes jalannya kode program untuk mencari/mendebug kesalahan dari program.
- Frame/Views, fasilitas ini gunanya untuk menunjukan atau menampilkan hasil dari perintah-perintah kode program yang dieksekusikan oleh Compiler dari Editor.

C. Arduino IDE

Arduino IDE (Integrated Development Environment) adalah sebuah software yang sangat berperan untuk menulis program, meng-compile menjadi kode biner dan meng-upload ke dalam memory microcontroller. IDE software dapat dijalankan pada sistem operasi berbasis windows maupun linux. IDE juga berguna untuk uploading sketch kepada board arduino dengan perantara USB port yang ada pada board .

Arduino IDE ini berguna sebagai text editor untuk membuat, mengedit, dan juga mevalidasi kode program. bisa juga digunakan untuk meng-upload ke board Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino “sketch” atau disebut juga source code arduino, dengan ekstensi file source code.ino .

Ada banyak projek dan alat-alat dikembangkan oleh akademisi dan profesional dengan menggunakan Arduino, selain itu juga ada banyak modul-modul pendukung (sensor, tampilan, penggerak dan sebagainya) yang dibuat oleh pihak lain untuk bisa disambungkan dengan Arduino. Arduino berevolusi menjadi sebuah platform karena ia menjadi pilihan dan acuan bagi banyak praktisi .

BAB II

PEMBAHASAN

2.1 Hardware

Arduino Uno merupakan mikrokontroler berbasis ATmega328 yang memiliki 14 digital input atau output pins. Penggunaan Arduino yaitu dengan koneksi USB, menggunakan chip FTDI untuk melakukan pemrograman, dan biasanya pada chip Arduino sudah dimasukkan bootloader, sehingga dapat dilakukan pemrograman langsung ke dalam chip menggunakan software Arduino.



Software Arduino digunakan dengan beberapa proses diantaranya menyambungkan laptop atau komputer dan Arduino Uno dengan menggunakan USB *connestion*. Namun sebelumnya telah dilakukan penginstalan software Arduino. USB *connestion* disambungkan dengan modul Arduino selanjutnya dilakukan pengaturan ports dan driver pada device manager dengan beberapa langkah yaitu membuka start menu, control panel, sistem dan security, sistem, device manager. Membuka lembar kerja dan menyimpannya sesuai rumus serta pengukuran yang akan ditentukan.

2.2 Software

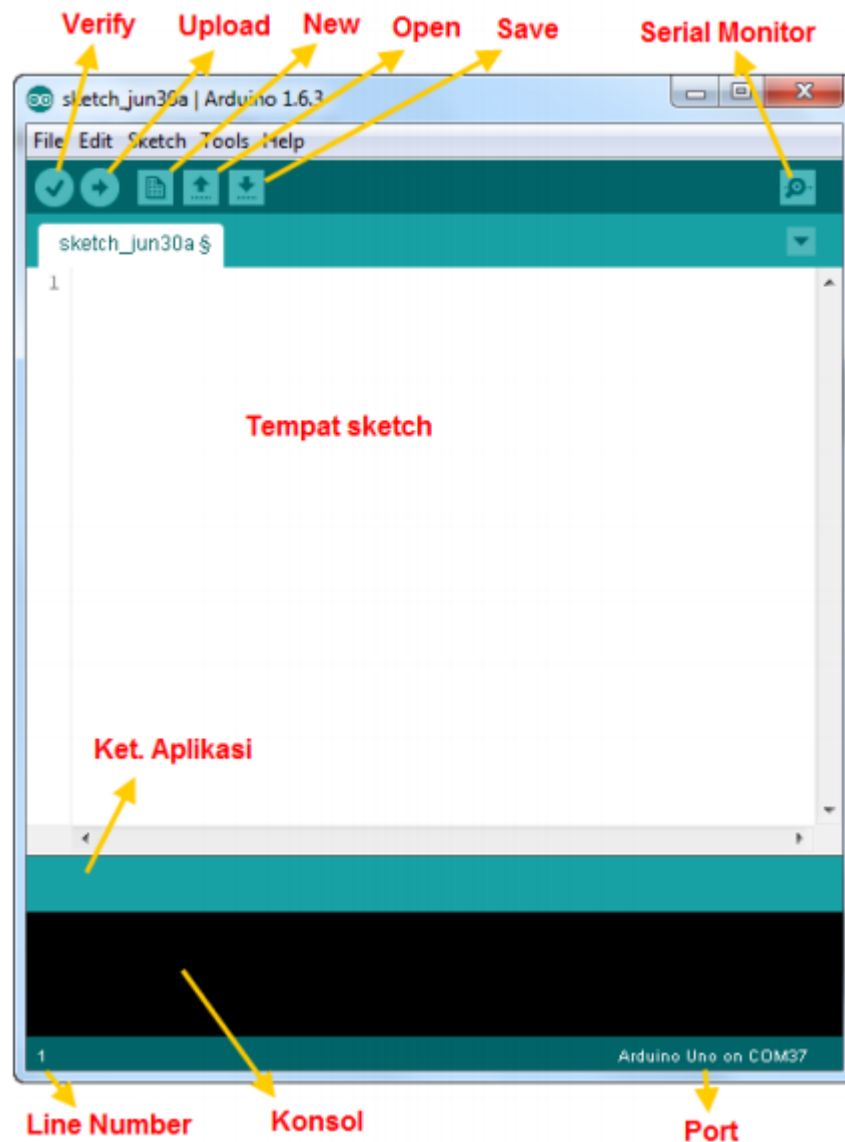
Software yang digunakan pada arduino adalah driver dan IDE, walaupun masih ada beberapa software lain yang sangat berguna selama pengembangan arduino. IDE atau Integrated Development Environment merupakan suatu program khusus untuk suatu komputer agar dapat membuat suatu rancangan atau sketsa program untuk papan Arduino. Arduino melakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (Sketch) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama Bootlader yang berfungsi sebagai penengah antara compiler Arduino dengan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut Wiring yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

2.2.1 Interface Arduino IDE

Program yang ditulis dengan menggunakan Arduino Software (IDE) disebut sebagai sketch. Sketch ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi .ino. Teks editor pada Arduino Software memiliki fitur” seperti cutting/paste dan seraching/replacing sehingga memudahkan kamu dalam menulis kode program.

Pada Software Arduino IDE, terdapat semacam message box berwarna hitam yang berfungsi menampilkan status, seperti pesan error, compile, dan upload program. Di bagian bawah paling kanan Software Arduino IDE, menunjukkan board yang terkonfigurasi beserta COM Ports yang digunakan.



Gambar 1. Interface Arduino IDE

Interface Arduino IDE tampak seperti gambar 1. Dari kiri ke kanan dan atas ke bawah, bagian-bagian IDE Arduino terdiri dari:

- **Verify** : pada versi sebelumnya dikenal dengan istilah Compile. Sebelum aplikasi diupload ke board Arduino, biasanya untuk memverifikasi terlebih dahulu sketch yang dibuat. Jika ada kesalahan pada sketch, nanti akan muncul error. Proses Verify / Compile mengubah sketch ke binary code untuk diupload ke mikrokontroler.

- Upload : tombol ini berfungsi untuk mengupload sketch ke board Arduino. Walaupun tidak mengklik tombol verify, maka sketch akan di-compile, kemudian langsung diupload ke board. Berbeda dengan tombol verify yang hanya berfungsi untuk memverifikasi source code saja.
- New Sketch : Membuka window dan membuat sketch baru
- Open Sketch : Membuka sketch yang sudah pernah dibuat. Sketch yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file .ino
- Save Sketch : menyimpan sketch, tapi tidak disertai mengcompile.
- Serial Monitor : Membuka interface untuk komunikasi serial, nanti akan diskusikan lebih lanjut pada bagian selanjutnya
- Keterangan Aplikasi : pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal "Compiling" dan "Done Uploading" ketika mengcompile dan mengupload sketch ke board Arduino
- Konsol : Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang sketch akan muncul pada bagian ini. Misal, ketika aplikasi mengcompile atau ketika ada kesalahan pada sketch yang buat, maka informasi error dan baris akan diinformasikan di bagian ini.
- Baris Sketch : bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada sketch.

Informasi Port : bagian ini menginformasikan port yang dipakai oleh board Arduino.

Menu Bar

Ada 5 menu yang akan membantu melakukan programing dengan aplikasi ini, yaitu 'File', 'Edit', 'Sketch', 'Tools', dan 'Help'. Menu File digunakan untuk berinteraksi dengan file *.ino. Menu Edit digunakan untuk mengedit program yang sedang ditulis di sketch editor. Menu Sketch untuk mem-verify, meng-upload sketch. Menu Tools untuk memanggil tools pendukung software Arduino ini. Menu Help berisi bantuan/catatan/keterangan yang mungkin dibutuhkan jika ada pertanyaan. Tiap menu memiliki sub-submenu sendiri.

Menu File Arduino IDE terdiri dari:

- **New**, berfungsi untuk membuat membuat sketch baru dengan bare minimum yang terdiri void setup() dan void loop().
- **Open**, berfungsi membuka sketch yang pernah dibuat di dalam drive.
- **Open Recent**, merupakan menu yang berfungsi mempersingkat waktu pembukaan file atau sketch yang baru-baru ini sudah dibuat.
- **Sketchbook**, berfungsi menunjukan hirarki *sketch* yang kamu buat termasuk struktur foldernya.
- **Example**, berisi contoh-contoh pemrograman yang disediakan pengembang Arduino, sehingga kamu dapat mempelajari program-program dari contoh yang diberikan.
- **Close**, berfungsi menutup jendela Arduino IDE dan menghentikan aplikasi.
- **Save**, berfungsi menyimpan *sketch* yang dibuat atau perubahan yang dilakukan pada *sketch*
- **Save as...**, berfungsi menyimpan *sketch* yang sedang dikerjakan atau *sketch* yang sudah disimpan dengan nama yang berbeda.
- **Page Setup**, berfungsi mengatur tampilan page pada proses pencetakan.
- **Print**, berfungsi mengirimkan file sketch ke mesin cetak untuk dicetak.
- **Preferences**, disini kam dapat merubah tampilan *interface* IDE Arduino.
- **Quit**, berfungsi menutup semua jendela Arduino IDE. *Sketch* yang masih terbuka pada saat tombol *Quit* ditekan, secara otomatis akan terbuka pada saat Arduino IDE dijalankan.

Menu Edit Arduino IDE terdiri dari:

- **Undo/Redo**, berfungsi untuk mengembalikan perubahan yang sudah dilakukan pada *Sketch* beberapa langkah mundur dengan *Undo* atau maju dengan *Redo*.
- **Cut**, berfungsi untuk meremove teks yang terpilih pada editor dan menempatkan teks tersebut pada *clipboard*.
- **Copy**, berfungsi menduplikasi teks yang terpilih kedalam editor dan menempatkan teks tersebut pada *clipboard*.

- ***Copy for Forum***, berfungsi melakukan *copy* kode dari editor dan melakukan *formatting* agar sesuai untuk ditampilkan dalam forum, sehingga kode tersebut bisa digunakan sebagai bahan diskusi dalam forum.
- ***Copy as HTML***, berfungsi menduplikasi teks yang terpilih ke dalam editor dan menempatkan teks tersebut pada *clipboard* dalam bentuk atau format HTML. Biasanya ini digunakan agar code dapat diembedddkan pada halaman web.
- ***Paste***, berfungsi menyalin data yang terdapat pada *clipboard*, ke dalam editor.
- ***Select All***, berfungsi untuk melakukan pemilihan teks atau kode dalam halaman editor.
- ***Comment/Uncomment***, berfungsi memberikan atau menghilangkan tanda *//* pada kode atau teks, dimana tanda tersebut menjadikan suatu baris kode sebagai komen dan tidak disertakan pada tahap kompilasi.
- ***Increase/Decrease Indent***, berfungsi untuk mengurangi atau menambahkan indentasi pada baris kode tertentu. Indentasi adalah “tab”.
- ***Find***, berfungsi memanggil jendela window *find and replace*, dimana kamu dapat menggunakannya untuk menemukan variabel atau kata tertentu dalam program atau menemukan serta menggantikan kata tersebut dengan kata lain.
- ***Find Next***, berfungsi menemukan kata setelahnya dari kata pertama yang berhasil ditemukan.
- ***Find Previous***, berfungsi menemukan kata sebelumnya dari kata pertama yang berhasil ditemukan.

Menu Sketch Arduino IDE terdiri dari:

- ***Verify/Compile***, berfungsi untuk mengecek apakah *sketch* yang kamu buat ada kekeliruan dari segi sintaks atau tidak. Jika tidak ada kesalahan, maka sintaks yang kamu buat akan dikompilasi ke dalam bahasa mesin.
- ***Upload***, berfungsi mengirimkan program yang sudah dikompilasi ke Arduino Board.

- ***Uplad Using Programmer***, menu ini berfungsi untuk menuliskan *bootloader* kedalam IC Mikrokontroler Arduino. Pada kasus ini kamu membutuhkan perangkat tambahan seperti *USBasp* untuk menjembatani penulisan program *bootloader* ke IC Mikrokontroler.
- ***Export Compiled Binary***, berfungsi untuk menyimpan file dengan ekstensi **.hex**, dimana file ini dapat disimpan sebagai arsip untuk di upload ke board lain menggunakan tools yang berbeda.
- ***Show Sketch Folder***, berfungsi membuka folder *sketch* yang saat ini dikerjakan.
- ***Include Library***, berfungsi menambahkan library/pustaka kedalam *sketch* yang dibuat dengan menyertakan sintaks `#include` di awal kode. Selain itu kamu juga bisa menambahkan library eksternal dari file **.zip** kedalam Arduino IDE.
- ***Add File...***, berfungsi untuk menambahkan file kedalam *sketch* arduino (file akan dikopikan dari drive asal). File akan muncul sebagai tab baru dalam jendela *sketch*.

Menu Tools Arduino IDE terdiri dari:

- ***Auto Format***, berfungsi melakukan pengatran format kode pada jendela editor
- ***Archive Sketch***, berfungsi menyimpan sketch kedalam file **.zip**
- ***Fix Encoding & Reload***, berfungsi memperbaiki kemungkinan perbedaan antara pengkodean peta karakter editor dan peta karakter sistem operasi yang lain.
- ***Serial Monitor***, berfungsi membuka jendela serial monitor untuk melihat pertukaran data.
- ***Board***, berfungsi memilih dan melakukan konfigurasi board yang digunakan.
- ***Port***, memilih port sebagai kanal komunikasi antara software dengan hardware.

- **Programmer**, menu ini digunakan ketika kamu hendak melakukan pemrograman chip mikrokontroler tanpa menggunakan koneksi Onboard USB-Serial. Biasanya digunakan pada proses *burning bootloader*.
- **Burn Bootloader**, mengizinkan kamu untuk mengkopikan program bootloader kedalam IC mikrokontroler

Menu Help Arduino IDE

Disini mampu mendapatkan bantuan terhadap permasalahan mengenai pemrograman. Menu help berisikan file-file dokumentasi yang berkaitan dengan masalah yang sering muncul, serta penyelesaiannya. Selain itu pada menu help juga diberikan link untuk menuju Arduino Forum guna menanyakan serta mendiskusikan berbagai masalah yang ditemukan.

Sketch Pada Arduino IDE

Arduino Software IDE, menggunakan konsep *sketchbook*, dimana *sketchbook* menjadi standar peletakan dan penyimpanan file program. Sketch yang telah kamu buat dapat dibuka dengan dari **File -> Sketchbook**, atau dengan menu **Open**.

Tabs, Multiple Files, dan Compilations Pada Arduino IDE

Mekanisme ini mengizinkan user untuk melakukan manajemen sketch, dimana lebih dari satu file dibuka dalam tab yang berbeda.

Uploading

Merupakan mekanisme untuk mengkopikan file **.hex** atau file hasil kompilasi kedalam IC mikrokontroler Arduino. Sebelum melakukan *uploading*, yang perlu kamu pastikan adalah jenis board yang kamu gunakan dan COM Ports dimana keduanya terletak pada menu **Tools -> Board** dan **Tools -> Port**.

Library

Library/ Pustaka merupakan file yang memberikan fungsi ekstra dari sketch yang kamu buat, semisal agar Arduino dapat bekerja dengan *hardware* tertentu dan

melakukan proses manipulasi data. Untuk menginstal *Library* pihak ketiga alias *Library* bukan dari Arduino, dapat dilakukan dengan Library Manager, Import file **.zip**, atau kopi paste secara manual di folder libraries pada Documents di platform Windows.

Serial Monitor

Serial monitor merupakan suatu jendela yang menunjukkan data yang dipertukarkan antara arduino dan komputer selama beroperasi, sehingga kamu bisa menggunakan serial monitor ini untuk menampilkan nilai hasil operasi atau pesan *debugging*. Selain melihat data, kamu juga bisa mengirimkan data ke Arduino melalui serial monitor ini, caranya dengan memasukkan data pada *text box* dan menekan tombol *send* untuk mengirimkan data. Hal penting yang harus kamu perhatikan adalah menyamakan *baudrate* antara serial monitor dengan Arduino board. Untuk menggunakan kemampuan komunikasi serial ini, pada Arduino, di bagian fungsi void `setup()`, diawali dengan instruksi **Serial.begin** diikuti dengan nilai *baudrate*.

Preferences

Preferences mengatur tentang beberapa hal dalam penggunaan Arduino Software IDE, seperti ukuran font, lokasi dimana menyimpan sketchbook, bahasa yang digunakan pada Arduino Software IDE, dan masih banyak lagi. Kamu bisa mengatur preferences pada menu file yang dapat dijumpai pada platform Windows dan Linux. Language Support

Language Support merupakan pilihan bahasa yang dapat disesuaikan pada Software Arduino IDE. Bahasa Indonesia sudah ada loh. Language Support ini dapat ditemukan pada menu **file -> preferences** atau dengan menekan **Ctrl+Comma**.

Boards

Pemilihan board pada Arduino Software IDE, berdampak pada dua parameter yaitu kecepatan CPU dan baudrate yang digunakan ketika melakukan kompilasi dan meng-upload *sketch*.

Sketch Arduino

Pada arduino bahasa pemrograman yang digunakan adalah bahasa C/C++. Program pada Arduino terbagi menjadi tiga bagian utama yaitu **Structure**, **Values** (berisi variable dan konstanta) dan yang terakhir **function**.

1. Structure. struktur kode pada arduino yaitu berisi fungsi `setup()` dan `loop()`.

a. Setup()

fungsi ini dipanggil pertama kali ketika menjalankan sketch. digunakan sebagai tempat inisialisasi *variable*, *pin mode*, penggunaan *library* dan lainnya. fungsi ini dijalankan sekali ketika *board* dinyalakan atau di reset.

```
*/  
  
// the setup function runs once when you press reset or power the  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}
```

Gambar 2. Fungsi Setup

b. loop()

Setelah membuat fungsi `setup()` sebagai tempat inisialisasi variabel dan menetapkan nilai maka selanjutnya fungsi `loop()` seperti namanya fungsi ini akan melakukan perulangan berturut-turut, memungkinkan program untuk mengubah dan menanggapi. digunakan untuk mengontrol *board* Arduino.

```
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by makin  
  delay(1000); // wait for a second  
}
```

Gambar 3. Fungsi Loop

2. Values. Berisi variable atau konstanta sesuai dengan type data yang didukung oleh Arduino.

3. Function. Segmentasi kode ke fungsi memungkinkan programmer untuk membuat potongan-potongan modular kode yang melakukan tugas yang terdefinisi dan kemudian kembali ke asal kode dari mana fungsi itu “dipanggil”. Umumnya menggunakan fungsi adalah ketika salah satu kebutuhan untuk melakukan tindakan yang sama beberapa kali dalam sebuah program.

2.2.2 Komunikasi Serial

Komunikasi serial merupakan komunikasi data dengan pengiriman data satu per satu pada satuan waktu. Transmisi data pada komunikasi serial dilakukan per bit. Komunikasi serial membutuhkan dua jalur yaitu transmit (Tx) dan receive (Rx). Komunikasi serial memiliki dua mode yaitu: sinkron dan Asinkron. Mode sinkron proses pengiriman data bersamaan dengan sinyal clock, sehingga pengiriman satu karakter dengan karakterlainnya memiliki jeda waktu yang sama. Sedangkan mode Asinkron dimana proses pengiriman data tanpa sinyal clock. Transmitter yang mengirim data harus menyepakati suatu standar Universal Asynchronous Receive Transmit (UART) Atmega 328 menyediakan serial komunikasi UART TTL (5V). Software arduino mencakup sebuah serial monitor yang memungkinkan data tekstual terkirim ke dan dari board arduino.

Dalam pengaturan UART untuk mengkoneksikan arduino dengan perangkat lain terdapat perintah-perintah:

- a. **Start Bit** merupakan penanda awal dimana akan dilakukan suatu proses pengiriman bit data.
 - b. **Data Bit** merupakan data yang akan di kirim
 - c. **Parity Bit** berfungsi sebagai “flag” atau bisa dikatakan sebagai penanda
 - d. **Stop Bit** berguna sebagai penanda proses pengiriman bit data telah selesai
 - e. **Bit Rate** merupakan jumlah dari bit yang terkirim atau diterima per satuan waktu (second)
 - f. **Baud Rate** Banyaknya perubahan data yang terjadi per satuan waktu.
- Perintah terakhir yang juga dibutuhkan dalam menentukan pembacaan nomor port serial yang terhubung dengan PC yang dikenal dengan Serial Communication Port Number.

2.2.3 Serial Software

Dalam pengembangan sebuah proyek mikrokontroler, Library serial arduino digunakan untuk media komunikasi dengan hardware serial port. Library serial menjadikan penggunaan port serial lebih sederhana dengan menyekatnya dari

hardware lainnya yang lebih kompleks. Contoh peng-kodean Serial Library pada sketch arduino.

2.2.4 Serial Hardware

Port USB (Universal Serial Bus) yang ada pada arduino adalah hardware yang merupakan port masukan/keluaran baru yang dibuat untuk mengatasi kekurangan-kekurangan port serial maupun paralel yang sudah ada (Hyde & John, 2009). USB digunakan sebagai port komunikasi serial dengan komputer PC, membuatnya lebih mudah di hubungkan dengan aplikasi antarmuka (interfaces) apapun, baik yang dirancang berbasis website menggunakan PHP dan MySQL maupun antarmuka berbasis desktop menggunakan microsoft Visual Studio.

2.2.5 Fungsi Dasar

1. `PinMode(pin, SET)`

Fungsi ini digunakan untuk menginisialisasi sebuah pin, dan menentukan pin tersebut akan digunakan sebagai input ataupun output. Nilai SET dapat diisi OUTPUT atau INPUT, tergantung dari kebutuhan. Sedangkan nilai pin adalah nomor pin pada mikrokontroler yang akan diset sebagai input atau output. Contoh: `pinMode(13, OUTPUT)` artinya menentukan pin digital 13 pada Arduino berfungsi sebagai output.

2. `DigitalWrite(pin, VAL)`

Fungsi ini digunakan untuk menuliskan nilai secara digital pada suatu pin. Nilai VAL dapat berupa HIGH (ON) atau LOW (OFF) dan nilai pin adalah nomor pin pada Arduino yang akan diset. Contoh: `digitalWrite(13, HIGH)` artinya pin digital 13 diset pada kondisi menyala.

3. `DigitalRead(pin)`

Fungsi ini digunakan untuk membaca nilai input/masukan yang diberikan ke Arduino. Nilai yang terbaca oleh Arduino melalui `digitalRead()` bergantung pada voltase pada pin yang diatur. Ketergantungan pada nilai voltase ini disebut Logic Level. Pada Arduino, batasan nilai yang mencukupi untuk mencapai HIGH adalah di antara 5-3 volt, sedangkan batasan nilai yang mencapai nilai LOW adalah di antara 0-1,5 volt. Contoh: `digitalRead(13)`

artinya Arduino akan membaca input yang diberikan melalui pin 13, hasilnya HIGH atau LOW.

4. `AnalogWrite(pin, VAL)`

Fungsi `analogWrite()` adalah fungsi yang digunakan untuk menuliskan suatu nilai berupa angka pada sebuah komponen, misalnya LED. Pengguna dapat mengatur seberapa terang cahaya dari lampu LED saat menyala, tergantung pada nilai yang dituliskan. Fungsi ini akan berguna ketika mulai bermain dengan sensor, di mana nilai yang terbaca seringkali berupa analog (memiliki banyak nilai, misal 0-1023), bukan digital (hanya memiliki 2 nilai, 0 (LOW) dan 1(HIGH)). Contoh: `analogWrite(13, 1023)` artinya Arduino akan memberikan instruksi nilai maksimal pada pin 13, sehingga komponen yang terkoneksi di pin 13 akan menyala maksimal (1023).

5. `analogRead(pin)`

Fungsi ini mirip dengan fungsi `digitalRead()`, yaitu membaca nilai masukan pada suatu pin. Bedanya adalah fungsi `analogRead()` akan menghasilkan nilai dari 0 hingga 1023, yang merepresentasikan voltase 0 v hingga 5 v. Contoh: `analogRead(0)` artinya Arduino akan membaca nilai input dari pin 0.

6. `delay(time)`

Fungsi ini digunakan untuk memberikan jeda antar fungsi. Nilai `time` adalah waktu lamanya jeda dalam satuan ms (milisekon), di mana 1 detik setara dengan 1.000 milisekon.

7. `Serial.begin(baudrate)`

Pengguna dapat melakukan komunikasi serial antara Arduino dengan PC, dengan menggunakan Serial Monitor yang disediakan pada Arduino IDE. Pada Serial Monitor, kita bisa melihat data yang dikirim dari Arduino ke PC. Selain itu, kita juga bisa mengirim data ke Arduino dengan cara mengetikkannya pada textbox di bagian atas Arduino IDE. Untuk memakai serial, yang pertama harus dilakukan adalah melakukan inisiasi dengan menggunakan fungsi `Serial.begin(baudrate)`. Variabel `baudrate` merupakan rasio modulasi, dan harus dicocokkan dengan `baudrate hardware` yang akan dikomunikasikan.

Contoh: `Serial.begin(9600)` artinya komunikasi akan berjalan pada rasio modulasi/baudrate 9600.

8. `Serial.available()`

Fungsi ini digunakan untuk mengetes apakah ada input data dari hardware yang disambungkan ke serial port, misalnya dari PC. Fungsi ini akan menghasilkan 1 apabila ada masukan, dan 0 apabila tidak ada masukan.

9. `Serial.read()`

Fungsi ini berfungsi untuk membaca karakter pada serial port. Karakter yang dibaca akan disimpan dalam bentuk ASCII (misalnya karakter '0' memiliki representasi ASCII yaitu 48).

10. `Serial.print()` dan `Serial.println()`

Fungsi ini digunakan untuk menuliskan suatu kalimat ke Serial Monitor, tetapi tidak mengirimkan data apapun, alias hanya digunakan untuk memberikan teks visual pada pengguna. `Serial.print("text")` digunakan untuk menulis "text", sedangkan `Serial.println("text")` dipakai untuk menuliskan kata "text" dan diakhiri dengan enter (kalimat selanjutnya ada di baris berikutnya).

11. `Serial.write(VALUE)`

Untuk mengirimkan data dari arduino ke PC, kita bisa menggunakan fungsi `Serial.write(VALUE)`. Nilai VALUE adalah data yang ingin dikirimkan dari arduino ke PC, dengan ukuran 1 byte.

2.2.6 Type data

1. Boolean

Tipe boolean hanya dapat digunakan untuk menampung dua nilai saja true atau false. Kedua nilai tersebut adalah konstanta yang sudah disediakan di bahasa Arduino. Variabel yang bertipe boolean biasa digunakan pada kondisi pernyataan seperti if, while, dan for. Kebutuhan memori 1 byte.

2. Char

Tipe char berguna untuk menyimpan sebuah nilai karakter seperti 'A', '9', dan '*'. Perlu diketahui, konstanta karakter ditulis di dalam tanda petik tunggal. Didalam memori, karakter disimpan dalam bentuk bilangan (ASCII), yang berkisar antara 0 hingga 127. Dengan demikian char

huruf='A' dengan char huruf=65; adalah sama saja. Kebutuhan memori 1 byte.

3. Unsigned char

Serupa dengan tipe char, tetapi bilangan akan dikodekan dalam bentuk bilangan positif antara 0 sampai dengan 255. Kebutuhan memori 1 byte.

4. Byte

Tipe data ini berguna untuk menampung bilangan bulat yang berkisar antara 0 hingga 255. Kebutuhan memori 1 byte.

5. Int

Tipe data ini berguna untuk menampung bilangan bulat yang berkisar antara -32768 hingga 32767.

6. Unsigned int

Tipe data ini berguna untuk menampung bilangan bulat yang berkisar antara 0 hingga 65535. Konstanta bertipe unsigned int bisa ditulis dengan akhiran U atau u. Contoh : 33U. Kebutuhan memori 2 byte.

7. Long

Tipe data ini berguna untuk menampung bilangan bulat yang berkisar antara -2.147.483.648 hingga 2.147.483.647. Konstanta bertipe long ditulis dengan akhiran L atau l. Contoh : 45678901L. Kebutuhan memori 4 byte.

8. Unsigned long

Tipe data ini berguna untuk menampung bilangan bulat yang berkisar antara 0 hingga 4.294.967.295. Konstanta bertipe unsigned long bisa ditulis dengan akhiran UL atau ul. Kebutuhan memori 4 byte.


9. Float

Tipe data ini berguna untuk menyimpan bilangan real. Angka yang bisa disimpan dari -3.4028235E+38 hingga 3.4028235E+38. Tingkat presisi hingga 6-7 digit. Kebutuhan memori 4 byte.

2.3 Penggunaan IDE Arduino

Untuk dapat memahami fitur-fitur IDE yang paling penting, akan membuat program sederhana yang membuat dioda pemancar cahaya (LED)

berkedip. LED merupakan sumber cahaya murah dan efisien, dan Arduino sudah dilengkapi dengan beberapa LED. Satu LED yang berkedip menunjukkan apakah Arduino saat ini memiliki daya, dan dua LED lainnya berkedip saat data ditransmisikan atau diterima melalui koneksi serial. Dalam proyek kecil pertama, akan membuat LED Arduino yang berkedip. Status LED terhubung ke digital IO pin 13. Pin digital bertindak sebagai salah satu jenis switch : TINGGI atau RENDAH. Jika diatur ke TINGGI, output pin diatur ke 5 volt, menyebabkan arus mengalir melalui LED, sehingga LED menyala. Jika diatur kembali ke LOW, aliran arus berhenti, dan LED akan mati. Berikut kode di editor :

The image is a screenshot of the Arduino IDE's code editor. At the top, there is a menu bar with 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for saving, running, and other functions. The main area of the editor shows the 'Blink' sketch code. The code is as follows:

```
const unsigned int LED_PIN = 13;
const unsigned int PAUSE = 500;

void setup() {
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_PIN, HIGH);
  delay(PAUSE);
  digitalWrite(LED_PIN, LOW);
  delay(PAUSE);
}
```

Gambar 4. Kode untuk menyalakan LED

program ini bekerja. Dalam dua baris pertama kita mendefinisikan dua konstanta int menggunakan const. LED_PIN mengacu pada jumlah IO pin digital yang digunakan, dan PAUSE mendefinisikan jumlah waktu yang diperlukan setiap LED berkedip dalam milidetik. Setiap program Arduino membutuhkan fungsi bernama setup(). Sebuah definisi fungsi selalu mematuhi skema berikut :

Nilai <return data> <function name ' (' <list dari parameters> ') '

Dalam kasus ini, nama fungsinya adalah `setup()`, dan jenis nilai kembalinya kosong membatalkan : ia tidak mengembalikan apa-apa. `setup()` tidak mengharapkan adanya argumen, jadi parameternya juga kosong.

2.4 Kompilasi dan Program Uploading

Sebelum mulai mengkompilasi dan meng-upload program ke Arduino, maka harus mengkonfigurasi dua hal dalam IDE : jenis Arduino yang digunakan dan port serial Arduino yang terhubung ke. Mengidentifikasi jenis Arduino mudah, karena dicetak di papan tulis. Jenis populer adalah Uno, Duemilanove, Diecimila, Nano, Mega Mini, NG, BT, LilyPad, Pro, atau Pro Mini. Dalam beberapa kasus, harus memeriksa apa mikrokontroler Arduino menggunakan ATmega168 atau sebuah ATmega328. Untuk menemukan jenis mikrokontroler dicetak pada mikrokontroler itu sendiri.

Ketika telah mengidentifikasi dengan tepat jenis Arduino, kemudian memilih dari menu `Tools > Board`. Sekarang Anda harus memilih port serial Arduino Anda terhubung untuk dari `> menu Serial Port Tools`. Pada Mac OS X, nama port serial dimulai dengan `/dev/cu.usbserial` atau `/dev/cu.usbmodem`. Pada sistem Linux, itu harus `/dev/ttyUSB0`, `/dev/ttyUSB1`, atau sesuatu yang serupa tergantung pada jumlah port USB komputer. Pada sistem Windows, itu sedikit lebih rumit untuk mengetahui kanan port serial, tapi masih tidak sulit. Pergi ke Device Manager, dan mencari USB Serial Port di bawah Ports (COM & LPT) entri menu Biasanya port bernama COM1, COM2, atau sesuatu yang serupa. Setelah telah memilih port serial kanan, klik tombol Verify, dan akan dapat dilihat output berikut di daerah pesan IDE (yang Arduino IDE menyebut program sketsa) : Binary ukuran sketsa : 1010 bytes (dari 32256 byte maksimum) Ini berarti IDE berhasil telah menyusun kode sumber ke dalam 1.010 byte kode mesin yang kita dapat meng-upload ke Arduino.

Jika melihat pesan kesalahan sebagai gantinya, periksa apakah telah mengetik di Program benar. Tergantung pada papan Arduino yang digunakan, maksimum byte mungkin berbeda. Misalnya pada Arduino Duemilanove, biasanya 14336 byte. Sekarang klik tombol Upload, dan setelah beberapa detik, maka akan muncul output berikut di bagian isi pesan :

Binary ukuran sketsa : 1010 bytes (dari 32256 byte maksimum) Ini adalah persis pesan yang sama mendapat setelah kompilasi program, dan itu memberitahukita bahwa 1.010 byte kode mesin dipindahkan berhasil ke Arduino. Dalam kasus kesalahan, memeriksa apakah Anda memilih jenis Arduino benar dan port serial yang benar dalam Menu Tools.

Selama proses upload, TX dan RX LED akan berkedip selama beberapa detik. Ini adalah normal, dan itu terjadi setiap kali Arduino dan komputer Anda berkomunikasi melalui port serial. Ketika Arduino mengirimkan informasi, ternyata pada TX LED. Ketika mendapat beberapa bit, ternyata pada RX LED. Karena komunikasi ini cukup cepat, LED mulai berkedip. Menyusun dan meng-upload PROGRAM 40 sementara program ini berjalan. Pin dimulai dalam keadaan LOW dan tidak arus keluaran apapun. Kita mengaturnya agar TINGGI dalam perangkat lunak menggunakan `digitalWrite ()` dan biarkan keluaran 5 volt untuk 500 milidetik. Akhirnya, kita mengatur kembali ke LOW untuk 500 milidetik dan ulangi seluruh proses. Diakui, LED statusnya tidak terlihat sangat spektakuler. Jadi, dalam bagian berikutnya, kita akan melampirkan”nyata”LED ke Arduino.

BAB III

KESIMPULAN

Dari pembahasan mengenai materi IDE (Integrated Development Environment) Arduino maka dapat ditarik kesimpulan sebagai berikut

1. IDE Arduino sebuah software yang sangat berperan untuk menulis program, meng-compile menjadi kode biner dan meng-upload ke dalam memory microcontroller.
2. Sktech memiliki 2 struktur 2 bagian utama yaitu void setup untuk menginisasi variabel mengatur pin, baudrate dan lainnya serta void loop sebagai program utama

Daftar Pustaka

Agung, M Bangun. 2014. *ARDUINO FOR BEGINNER*. Banten

Djuandi, Feri. *Pengenalan Arduino*, Tobuku, Jakarta, 2011.

Noviardi. *Aplikasi Kominikasi Serial Arduino Uno R3 Pada Pengontrolan Dengan Menggunakan Visual Studio 2012 Dan Sql Server 2008.*, Jurnal Teknik Elektro ITP, Volume 5, No. 1; 2016

<https://djukarna4arduino.wordpress.com/2015/01/19/struktur-program-sketch-pad-a-arduino/> diakses 14 Oktober 2018