

BAB 2 LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan merupakan bab yang akan memaparkan teori-teori pendukung agar pembaca dapat memahami berbagai perangkat dan penjelasan istilah yang dipakai. Bab ini berisikan kajian pustaka dan dasar teori

1.1 Tinjauan Pustaka

Pada kajian pustaka membahas mengenai penelitian yang sudah pernah terlaksana dan bersinggungan dengan penelitian ini. Terdapat dua penelitian yaitu oleh (Daniel, Veronica, & Rony, 2016) dan (Mikhaylov & Huttunen, 2014). Pada Tabel 2.1 akan dijelaskan bagaimana persamaan dan perbedaan dari penelitian terdahulu dan penelitian yang akan direncanakan.

Tabel 0.1 Kajian Pustaka

| NO | Nama Penulis [Tahun], Judul | Persamaan | Perbedaan | |
|----|---|--|--|---|
| | | | Penelitian Terdahulu | Rencana Penelitian |
| 1 | Daniel, Veronica, & Rony [2016], Rancang Bangun Aplikasi Untuk Konfigurasi Otomatis Penggunaan Sensor Pada Arduino. | Pembuatan perangkat yang memberikan kemampuan dalam kefleksibelan penggunaan sensor untuk observasi dengan konfigurasi minimum yang diberikan <i>user</i> , dan sama dalam penggunaan perangkat mikrikontroler yaitu Arduino UNO | Membangun sebuah aplikasi berbasis web yang dapat digunakan sebagai antarmuka <i>user</i> terhadap informasi hasil observasi keadaan lingkungan dan sebagai antarmuka konfigurasi sistem tertanam dan sensor yang digunakan. | Tidak membangun aplikasi web, namun data yang diterima dari modul sensor akan diteruskan melalui modul komunikasi NRF24L01 atau ESP8266 yang akan ditampilkan hasilnya pada serial monitor. |

| NO | Nama Penulis [Tahun], Judul | Persamaan | Perbedaan | |
|----|--|---|---|---|
| | | | Penelitian Terdahulu | Rencana Penelitian |
| 2 | Konstantin Mikhaylov & Martti Huttunen [2014], Modular Wireless Sensor and Actuator Network Nodes with Plug-and-Play Module Connection | Memperkenalkan konsep baru dari modular <i>Wireless Sensor dan Actuator Network (WSAN) Node</i> dengan kemampuan koneksi modul <i>Plug-and-Play (P & P)</i> . | Ketika sensor terpasang maka <i>main controller</i> modul akan mengidentifikasi dengan <i>men-download driver</i> yang yang dibutuhkan tanpa mengetahui alamat sensor dan hanya menggunakan dua sensor. | Ketika sensor terpasang maka, akan mendeteksi melalui alamat yang sudah diberikan pada tiap-tiap sensor, dan Modul Antarmuka akan terus menerus <i>looping</i> untuk menemukan alamat yang tepat. Menggunakan empat sensor. |

1.2 Dasar Teori

Dalam bab dasar teori akan terdapat penjelasan yang mendukung pembuatan sistem berupa pengertian, cara kerja, spesifikasi, karakteristik dan lainnya untuk memudahkan pembaca memahami isi dari penelitian ini.

1.2.1 Konsep *Plug and play*

Plug and play atau yang sering disingkat dengan PnP merupakan suatu fitur yang dijalankan pada komputer agar penggunaanya tidak harus melakukan konfigurasi secara manual, dan tidak perlu meng-*install driver* untuk perangkat yang dihubungkan, dan bahkan komputer tersebut dapat secara otomatis mengenalinya. Sebagai contoh, ketika menghubungkan *mouse* ke USB port komputer, maka dalam beberapa saat komputer akan bekerja dengan fitur *Plug and play* untuk mengenali bahwa yang terhubung adalah sebuah *mouse*. Namun jika dalam komponen tersebut tidak memiliki fitur tersebut maka *user* diminta untuk melakukan langkah-langkah konfigurasi seperti meng-*install driver*, mengubah pengaturan perangkat, dan membutuhkan *software* pendukung kerja sistem. Hal ini juga dinilai tidak fleksibel dalam hal waktu karena proses pengerjaannya tidak langsung dapat dipakai.

Cara kerja *Plug and Play* pada Windows XP yaitu *user* cukup menghubungkan *hardware* dan menyerahkan tugas konfigurasi dan menjalankan *hardware* kepada *operating system*. Namun *hardware* komputer, *driver* perangkat, dan BIOS sistem semua harus didesain sedemikian supaya dapat menginstalasi perangkat baru

tanpa campur tangan *user*. Sebagai contoh, meskipun Windows XP menyediakan fungsi *Plug and Play*, jika tidak ada *driver Plug and Play* yang tersedia untuk perangkat yang dimaksud, maka *operating system* tidak bisa secara otomatis mengonfigurasi dan menjalankan perangkat.

Pada waktu *hardware terpasang*, misalnya saat *user* memasang kamera ke port USB, *Plug and Play Manager* melakukan langkah beberapa langkah untuk menginstalasi perangkat. Setelah menerima notifikasi adanya perangkat yang dipasang, *Plug and Play* mengecek *resource* yang dibutuhkan oleh perangkat (misalnya *interrupt*, *range memory*, *range I/O*, dan *channel DMA*) dan untuk memberi *resource* tersebut *Plug and Play Manager* mengecek nomor identifikasi perangkat. Kemudian mengecek *harddisk*, *drive floppy*, *drive CD-ROM*, dan *Windows Update* untuk mendapatkan *driver* yang sesuai dengan nomor identifikasi perangkat. Jika terdapat beberapa *driver*, *Plug and Play* memilih *driver* yang optimal dengan melihat ID *hardware*, *driver signature*, dan fitur *driver* lainnya yang paling tepat, kemudian menginstalasi *driver* dan menjalankan perangkat. (Chrisstenson, 2017).

1.2.2 Arduino UNO

Arduino UNO adalah *board Microcontroller* berbasis ATmega328 yang memiliki 14 pin *input* dari *output* digital. Enam pin *input* tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP header, dan tombol *reset*. Untuk mendukung *microcontroller* agar dapat digunakan, maka dilakukan dengan menghubungkan *Board* Arduino UNO ke komputer dengan kabel USB, dengan adaptor, maupun dengan baterai. Uno Arduino dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal secara otomatis. Eksternal (non-USB) daya dapat berasal baterai. (Ecadio, 2017)

Pemrograman *board* Arduino dilakukan dengan menggunakan Arduino *Software* (IDE). Chip ATmega328 yang terdapat pada Arduino UNO yang telah diisi program awal yang sering disebut *bootloader* untuk memudahkan dalam melakukan pemrograman lebih sederhana menggunakan Arduino *Software*, tanpa harus menggunakan tambahan *hardware* lain. Hanya dengan menghubungkan Arduino dengan kabel USB ke PC atau Mac/Linux, kemudian menjalankan Arduino *Software* (IDE), dan sudah dapat memulai program. Berikut ini adalah Tabel 2.2 yaitu penjelasan terkait spesifikasi Arduino UNO.

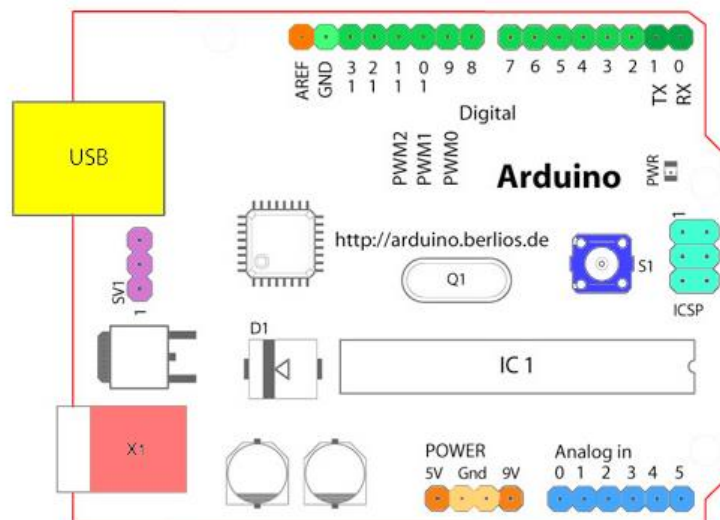
Tabel 0.2 Spesifikasi Arduino UNO

| | |
|---|-----------|
| Chip mikrokontroler | ATmega328 |
| Tegangan operasi | 5V |
| Tegangan <i>input</i> (yang direkomendasikan, <i>via</i> Jack DC) | 7V – 12V |
| Tegangan <i>input</i> (limit, <i>via</i> jack DC) | 6V – 20V |

| | |
|-------------------------|---|
| Digital I/O pin | 14 buah, 4 diantaranya menyediakan PWM |
| Analog <i>input</i> pin | 6 buah |
| Arus DC per pin I/O | 20 mA |
| Arus DC pin 3.3V | 50 mA |
| Memori <i>Flash</i> | 32 KB, 0.5 KB telah digunakan untuk <i>bootloader</i> |
| SRAM | 2KB |
| EEPROM | 1KB |
| <i>Clock speed</i> | 16Mhz |
| Dimensi | 69.6mm x 53.4mm |
| Berat | 25 g |

Sumber: Ecadio (2017)

Pada Gambar 2.1 dapat dilihat pin yang terdapat di dalam *board* Arduino UNO yang terbagi menjadi pin power, pin *Input* dan *output* berupa pin analog dan digital, serta pin lain yang mendukung kerja *Microcontroller* Arduino UNO.



Gambar 0.1 Board Arduino

Sumber: (www.arduino.cc, 2015)

Berikut ini adalah penjelasan dari pin Arduino UNO:

1. GND. Ini adalah ground atau negatif.
2. Vin. Ini adalah pin yang digunakan jika anda ingin memberikan power langsung ke *board* Arduino dengan rentang tegangan yang disarankan 7V - 12V
3. Pin 5V. Ini adalah pin *output* dimana pada pin tersebut mengalir tegangan 5V yang telah melalui regulator

4. 3V3. Ini adalah pin *output* dimana pada pin tersebut disediakan tegangan 3.3V yang telah melalui regulator
5. IOREF. Ini adalah pin yang menyediakan referensi tegangan mikrokontroler. Biasanya digunakan pada *board shield* untuk memperoleh tegangan yang sesuai, apakah 5V atau 3.3V

Pin *Input* dan *Output* pada Arduino UNO yang memiliki fungsi khusus:

1. Serial, terdiri dari 2 pin : pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) data serial.
2. External Interrupts, yaitu pin 2 dan pin 3. Kedua pin tersebut dapat digunakan untuk mengaktifkan interrupts. Gunakan fungsi `attachInterrupt()`
3. PWM: Pin 3, 5, 6, 9, 10, dan 11 menyediakan *output* PWM 8-bit dengan menggunakan fungsi `analogWrite()`
4. SPI : Pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan *SPI Library*
5. LED : Pin 13. Pada pin 13 terhubung *built-in led* yang dikendalikan oleh digital pin no 13.
6. TWI : Pin A4 (SDA) dan pin A5 (SCL) yang mendukung komunikasi TWI dengan menggunakan *Wire Library*

Arduino UNO memiliki 6 buah *input* analog, yang diberi tanda dengan A0, A1, A2, A3, A4, A5 yang diukur dari *ground* ke 5V. Beberapa pin lainnya pada *board* ini adalah:

1. AREF sebagai referensi tegangan untuk *input* analog.
2. Reset yang menghubungkan ke *LOW* untuk melakukan reset terhadap *microcontroller*. Sama dengan penggunaan tombol reset yang tersedia.

Arduino UNO R3 punya fasilitas komunikasi dengan komputer maupun *microcontroller* lainnya dimana Atmega328 menyediakan komunikasi serial UART TTL (5V) pada pin 0 (RX) dan pin 1 (TX). Pada Arduino *Software* (IDE) tersedia serial monitor untuk dapat menampilkan keluaran berupa data *text*. Penggunaan komunikasi serial dari digital pin, dapat menggunakan *SoftwareSerial library*. Dalam Arduino *Software* (IDE) juga dapat menggunakan *Wired Library* untuk memudahkan menggunakan bus I2C. Komunikasi ini menggunakan pin A4 (SDA) dan A5 SCL).

1.2.3 Inter Integrated Circuit (I2C)

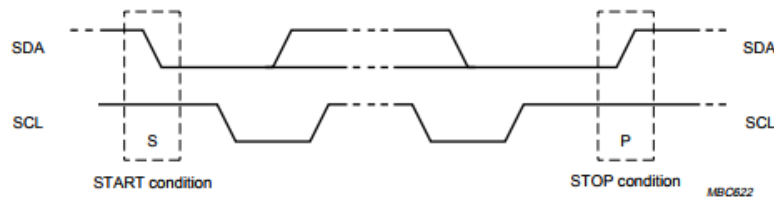
Inter Integrated Circuit atau biasa disebut dengan I2C merupakan salah satu komunikasi serial yang dikembangkan oleh Philips Semiconductor pada 1992. Komunikasi ini sering dipakai karena tuntutan desain elektronik yang semakin ringkas dan fleksibel. Keuntungan model komunikasi I2C adalah meminimalkan jalur hubungan antar komponen dengan membandingkannya jika menggunakan parallel bus.

Mode komunikasi yang disediakan I2C adalah *multimaster* dan *multislave*, yang artinya *master* ataupun *slave* dapat berjumlah lebih dari satu. Untuk mengetahui karakteristik I2C dapat dilihat pada Tabel 2.3

Tabel 0.3 Karakteristik I2C

| | |
|---------------------------------|--|
| Serial Bus | Data dikirim serial secara per-bit. |
| Terdiri dari dua jalur | SCL (<i>Serial Clock Line</i>) untuk menghantarkan sinyal <i>clock</i> . SDA (<i>Serial Data</i>) untuk mentransaksikan data |
| Jumlah Peserta Bus maksimal 127 | Peserta dialamatkan melalui 7 bit alamat. Alamat ditetapkan secara <i>hardware</i> dan hanya sebagian kecil dapat dirubah. |
| Pengirim dan Penerima | Setiap transaksi data terjadi antara pengirim (<i>Transmitter</i>) dan penerima (<i>Receiver</i>). Pengirim dan penerima adalah peserta bus. |
| Master and Slave | Device yang mengendalikan operasi transfer disebut <i>Master</i> , sementara <i>device</i> yang di kendalikan oleh Master disebut <i>Slave</i> . |

Transfer data pada I2C dilakukan apabila bus tidak dalam keadaan sibuk atau *idle*. Saat dilakukan transfer data antara *master* dan *slave*, keadaan data pada SDA harus stabil selama SCL dalam keadaan *high*. Keadaan perubahan “1” atau “0” pada SDA hanya dapat dilakukan selama SCL dalam keadaan *low*. Perubahan keadaan SDA saat SCL dalam keadaan *high* adalah perubahan yang dianggap sebagai sinyal *Start* atau sinyal *Stop*, terlihat pada Gambar 2.2 koneksi dari sinyal *start* dan *stop* pada I2C.

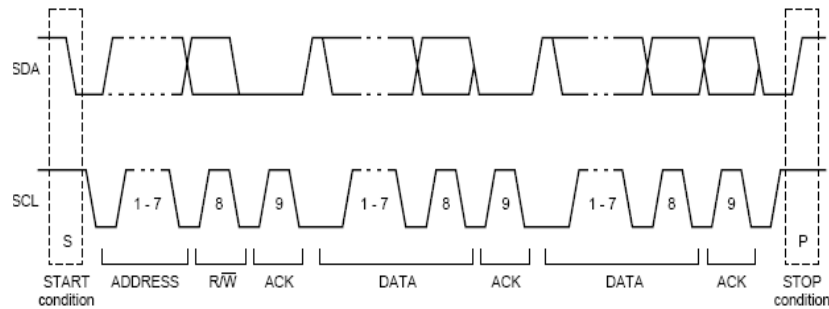


Gambar 0.2 Koneksi Sinyal START dan STOP

Sumber: (www.I2C-bus.org, 2015)

Dalam penggunaan komunikasi I2C pada *microcontroller* Arduino dapat menggunakan *library Wired*. Dalam hal ini I2C memiliki dua mode pada kode *master* dan *slave* yaitu *Master Transmitter Slave Receiver* dan *Master Receiver Slave Transmitter*, dan di setiap mode memiliki fungsi tertentu untuk menginisialisasi *library Wired*. (Jack, 2017)

Pengalamatan pada I2C terdiri dari 7 bit dan 10 bit, dalam kesehariannya yang biasa digunakan adalah 7 bit. Saat modul terhubung dengan jalur I2C 7 bit dapat menampung 128 (2^7) perangkat. Saat *master* atau *slave* mengirimkan 7 bit alamat, juga menyertakan 1 bit sebagai penginformasian perangkat apakah *master* menulis (*write*) data ke *slave* atau membaca (*read*) data dari *slave*. Jika bit tersebut 0, maka *master* menulis data ke *slave*, dan jika 1 maka *master* membaca data dari *slave*. Gambar 2.3 akan terlihat pengalamatan I2C dengan menyertakan *address* dan perintah *read/write*.

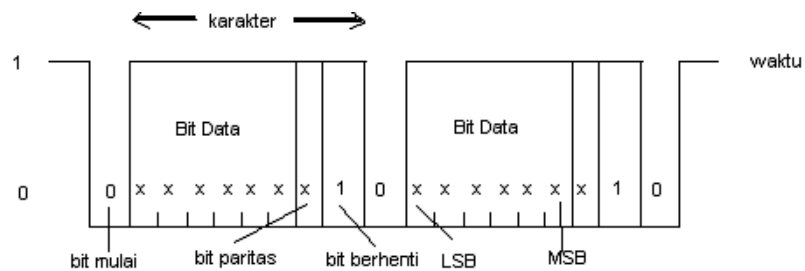


Gambar 0.3 Penglamatan I2C

Sumber: (<http://www.I2C-bus.org>, 2015)

1.2.4 Universal Asynchronous Receiver Transmitter (UART)

Universal Asynchronous Receiver-Transmitter merupakan metode dalam komunikasi dengan menggunakan transmisi serial. Komunikasi serial merupakan komunikasi data dengan pengiriman data secara satu per satu berurutan dimulai dari LSB (*Least Significant Bit*) sampai dengan MSB (*Most Significant Bit*), dengan menggunakan satu jalur kabel data dalam satu waktu. UART merupakan komunikasi serial dengan mode *asincron* (*no clock*). Untuk melihat lebih jelas tentang komunikasi UART dapat memperhatikan Gambar 2.8.



Gambar 0.4 Ilustrasi Komunikasi dengan UART

Sumber: (learn.sparkfun.com, 2017)

Transmisi serial asinkron memiliki kecepatan data yang bervariasi, oleh karena itu membutuhkan parameter-parameter yang harus dimengerti saat akan berkomunikasi. Parameter tersebut adalah:

1. Jumlah bit tiap karakter adalah 5 sampai 8 bit.
2. Parity bit yang digunakan untuk mendeteksi kesalahan (*error*) yang berbentuk *odd* (*ganjil*), *even* (*genap*).

3. Jumlah stop bit (1 bit, 1,5 bit, atau 2 bit) dan 1 start bit.
4. Baud rate atau kecepatan data (bps).

Arduino UNO memiliki satu buah serial UART yang berada pada pin digital 0 (Rx) dan 1 (Tx).