

2.4. Teknologi Pendukung

Pada bagian ini dijelaskan mengenai teknologi pendukung yang digunakan untuk merealisasikan sistem yang dibuat dalam proyek akhir ini.

2.4.1. Wifi

Istilah "hotspot" sudah merupakan ungkapan umum di dunia global untuk lokasi layanan akses WLAN bagi publik. Hot Spot Area adalah salah satu bentuk pemanfaatan teknologi Wireless LAN (WLAN IEEE 802.11b) pada lokasi publik seperti Bandara, Lobby Hotel, Ruang konferensi, Perguruan Tinggi dan Kafe. Teknologi WLAN ini mampu memberikan kecepatan akses kecepatan tinggi hingga 11 Mbps pada jangkauan hingga 100 meter dari Access Point (AP) tergantung struktur bangunan atau penghalang yang ada diantara AP dengan terminal pengguna.

Wi-Fi merupakan singkatan dari Wireless Fidelity, memiliki pengertian yaitu sekumpulan standar yang digunakan untuk Jaringan Lokal Nirkabel (Wireless Local Area Networks - WLAN) yang didasari pada spesifikasi IEEE 802.11. Standar terbaru dari spesifikasi 802.11a atau b, seperti 802.16 g, saat ini sedang dalam penyusunan, spesifikasi terbaru tersebut menawarkan banyak peningkatan mulai dari luas cakupan yang lebih jauh hingga kecepatan transfernya.

Awalnya Wi-Fi ditujukan untuk penggunaan perangkat nirkabel dan Jaringan Lokal (LAN), namun saat ini lebih banyak digunakan untuk mengakses internet. Hal ini memungkinkan seseorang dengan komputer dengan kartu nirkabel (wireless card) atau personal digital assistant (PDA) untuk terhubung dengan internet dengan menggunakan access point (atau dikenal dengan hotspot) terdekat.

Wi-Fi (Wireless Fidelity) adalah koneksi tanpa kabel seperti handphone dengan mempergunakan teknologi radio sehingga pemakainya dapat mentransfer data dengan cepat dan aman. Wi-Fi tidak hanya dapat digunakan untuk mengakses internet, Wi-Fi juga dapat digunakan untuk membuat jaringan tanpa kabel di perusahaan. Karena itu banyak orang mengasosiasikan Wi-Fi dengan kebebasan, karena teknologi Wi-Fi memberikan kebebasan kepada pemakainya untuk mengakses internet atau mentransfer data dari ruang meeting, kamar hotel, kampus, dan café-café yang bertanda Wi-Fi Hot Spot.

a. Standard WIFI

Standarisasi jaringan Wireless LAN adalah IEEE 802.11 = IEEE (Institute of Electrical and Electronic Engineers) merupakan institusi yang melakukan diskusi, riset dan pengembangan terhadap perangkat jaringan yang kemudian menjadi Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer 2010 Fakultas Ilmu Komputer, Universitas Brawijaya standarisasi untuk digunakan sebagai perangkat jaringan.(Zamidra,2014).

b. Spesifikasi WIFI

Wi-Fi dirancang berdasarkan spesifikasi IEEE 802.11. Sekarang ini ada empat variasi dari 802.11, yaitu: 802.11a, 802.11b, 802.11g, and 802.11n. Spesifikasi b merupakan produk pertama Wi-Fi. [13]

2.4.2. Komunikasi Serial

Komunikasi serial adalah komunikasi yang pengiriman datanya per-bit secara berurutan dan bergantian. Komunikasi ini mempunyai suatu kelebihan yaitu hanya membutuhkan satu jalur dan kabel yang sedikit dibandingkan dengan komunikasi paralel. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit sehingga lebih lambat dibandingkan komunikasi paralel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data di mana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Pada dasarnya komunikasi serial adalah kasus khusus komunikasi paralel dengan nilai $n = 1$, atau dengan kata lain adalah suatu bentuk komunikasi paralel dengan jumlah kabel hanya satu dan hanya mengirimkan satu bit data secara simultan. Hal ini dapat disandingkan dengan komunikasi paralel yang sesungguhnya di mana n -bit data dikirimkan bersamaan, dengan nilai umumnya $8 \leq n \leq 128$.

Komunikasi serial ada dua macam, *asynchronous serial* dan *synchronous serial*. *Synchronous serial* adalah komunikasi dimana hanya ada satu pihak (pengirim atau penerima) yang menghasilkan clock dan mengirimkan clock tersebut bersama-sama dengan data. Contoh penggunaan *synchronous serial* terdapat pada transmisi data keyboard. *Asynchronous serial* adalah komunikasi dimana kedua pihak (pengirim dan penerima) masing-masing menghasilkan clock namun hanya data yang ditransmisikan, tanpa clock. Agar data yang dikirim sama dengan data yang diterima, maka kedua frekuensi clock harus sama dan harus terdapat sinkronisasi. Setelah adanya sinkronisasi, pengirim akan mengirimkan datanya sesuai dengan frekuensi clock pengirim dan penerima akan membaca data sesuai dengan frekuensi clock penerima. Contoh penggunaan asynchronous serial adalah pada Universal Asynchronous Receiver Transmitter (UART) yang digunakan pada serial port (COM) komputer.

Antarmuka Kanal serial lebih kompleks/sulit dibandingkan dengan antarmuka melalui kanal paralel, hal ini disebabkan karena:

1. Dari Segi perangkat keras: adanya proses konversi data paralel menjadi serial atau sebaliknya menggunakan piranti tambahan yang disebut UART (Universal Asynchronous Receiver/Transmitter) dan
2. Dari Segi perangkat lunak: lebih banyak register yang digunakan atau terlibat. Namun di sisi lain antarmuka kanal serial menawarkan berapa kelebihan dibandingkan secara paralel, antara lain:

1. Kabel untuk komunikasi serial bisa lebih panjang dibandingkan dengan paralel; data-data dalam komunikasi serial dikirim-kan untuk logika '1' sebagai tegangan -3 s/d -25 volt dan untuk logika '0' sebagai tegangan +3 s/d +25 volt, dengan demikian tegangan dalam komunikasi serial memiliki ayunan tegangan maksimum 50 volt, sedangkan pada komunikasi paralel hanya 5 volt. Hal ini menyebabkan gangguan pada kabel-kabel panjang lebih mudah diatasi dibandingkan pada paralel.
2. Jumlah kabel serial lebih sedikit; Anda bisa menghubungkan dua perangkat komputer yang berjauhan dengan hanya 3 kabel untuk konfigurasi null modem, yaitu TXD (saluran kirim), RXD(saluran terima) dan Ground, bayangkan jika digunakan teknik paralel akan terdapat 20 – 25 kabel. Namun pada masing-masing komputer dengan komunikasi serial harus dibayar “biaya” antarmuka serial yang agak lebih mahal.
3. Banyaknya piranti saat ini (palmtop, organizer, hand-phone dan lainlain) menggunakan teknologi infra merah untuk komunikasi data, dalam hal ini pengiriman datanya dilakukan secara serial. IrDA-1 (spesifikasi infra merah pertama) mampu mengirimkan data dengan laju 115,2 kbps dan Konsep Komunikasi Serial 2 dibantu dengan piranti UART, hanya panjang pulsa berkurang menjadi 3/16 dari standar RS-232 untuk menghemat daya.
4. Untuk teknologi embedded system, banyak mikrokontroler yang dilengkapi dengan komunikasi serial (baik seri RISC maupun CISC) atau Serial Communication Interface (SCI); dengan adanya SCI yang terpadu pada 1C mikrokontroler akan mengurangi jumlah pin keluaran, sehingga hanya dibutuhkan 2 pin utama TxD dan RxD (di luar acuan ground).

2.4.2.2. Komunikasi Serial Pada Arduino

Komunikasi data serial digunakan untuk komunikasi antara board arduino dengan komputer atau perangkat lain. Semua board arduino mempunyai sedikitnya 1 buah port serial yang juga dikenal dengan nama UART atau USART. Komunikasi data serial menggunakan 2 buah pin yaitu pin RX untuk menerima data dan pin TX untuk mengirimkan data. Pada board arduino pin RX terletak pada pin0 dan pin TX terletak pada pin1. Ketika board arduino dikonfigurasi untuk berkomunikasi secara serial, maka kedua pin0 dan pin1 tidak dapat digunakan sebagai pin input/output digital.

Pada sistem operasi windows XP dan sebelumnya terdapat program HyperTerminal yang dapat digunakan untuk berkomunikasi secara serial dengan hardware, Pada Windows yang lebih baru, seperti Win7, Win8 dan Vista program hyperterminal sudah tidak tersedia. Tetapi hal ini tidak menjadi masalah, program Arduino telah menyediakan serial monitor yang dapat dibuka dengan memilih tool

– serial monitor pada menu program arduino atau dengan mengetik shift-ctrl-M bersamaan.

Board Arduino Nano dilengkapi dengan 1 buah serial port yang dapat diakses melalui mini-usb port dengan membuat virtual comport atau melalui pin0 dan pin1.

Program Arduino telah dilengkapi dengan serial port library yang memudahkan programmer untuk membuat program.

1. Serial.available()

Digunakan untuk menyatakan angka, bytes atau karakter yang sudah siap dibaca dari serial port. Data ini adalah data yang telah diterima dan disimpan dalam serial receive buffer. Serial receive buffer dapat menampung 64 bytes data. Berikut ini contoh penulisan instruksi available.

```
int dataterkirim = 0;
void setup()
{ Serial.begin(9600); }
void loop()
{
  if (Serial.available() > 0) {
    dataterkirim = Serial.read();
    Serial.print("saya menerima data: ");
    Serial.println(dataterkirim, DEC);
  }
}
```

2. Serial.begin()

“begin()” digunakan untuk mengatur baudrate / kecepatan transmisi data. Beberapa pilihan kecepatan komunikasi data yang dapat digunakan pada board arduino adalah 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 atau 115200. Pengaturan baudrate dilakukan pada bagian setup(). Berikut ini format penulisan dan contoh program “begin()”:

```
void setup() { Serial.begin(9600); }
```

3. Serial.end()

Perintah “serial.end()” digunakan untuk menutup komunikasi serial port. Berikut format penulisan dan contoh program. Untuk membuka kembali komunikasi serial port dapat menggunakan perintah “Serial.begin()”.

4. Serial.find()

Perintah “Serial.find()” digunakan untuk membaca data dari serial port buffer hingga target yang ditentukan dalam perintah “Serial.find()” terpenuhi. Fungsi akan

bernilai benar jika target tercapai dan bernilai salah jika target tidak tercapai. Berikut ini contoh penulisan program “Serial.find()”.

5. Serial.print()

Perintah “Serial.print” digunakan untuk menampilkan data ke serial monitor. Data yang ditampilkan dapat berupa karakter, bytes, atau angka. Berikut ini beberapa contoh perintah “serial.print()”.

```
Serial.print(78)           // mencetak "78"
Serial.print(1.23456)      // mencetak "1.23"
Serial.print('N')         // mencetak "N"
Serial.print("Hello world.") // mencetak "Hello world."
Serial.print(78, BIN)      // mencetak "1001110"
Serial.print(78, OCT)      // mencetak "116"
Serial.print(78, DEC)      // mencetak "78"
Serial.print(78, HEX)      // mencetak "4E"
Serial.println(1.23456, 0) // mencetak "1"
Serial.println(1.23456, 2) // mencetak "1.23"
Serial.println(1.23456, 4) // mencetak "1.2346"
```

Terdapat 2 macam yaitu Serial.print dan Serial.println. perbedaannya adalah jika “Serial.print” akan menampilkan data di serial monitor dalam satu baris, tetapi perintah “Serial.println” akan menampilkan data di serial monitor pada baris baru.

6. Serial.read()

Perintah “Serial.read()” digunakan untuk membaca data dari serial port. Berikut contoh penulisan perintah “Serial.read()”

```
void loop() {
  if (Serial.available() > 0) {
    dataterkirim = Serial.read();
    Serial.print("saya menerima data: ");
    Serial.println(dataterkirim, DEC);
  }
}
```

7. Serial.write()

Perintah “Serial.write” digunakan untuk membaca data biner dari serial port. Data ini dikirim dalam bentuk byte atau deretan data byte. Contoh penulisan perintah “Serial.write”.

```
void setup() {
  Serial.begin(9600); }
void loop() {
```

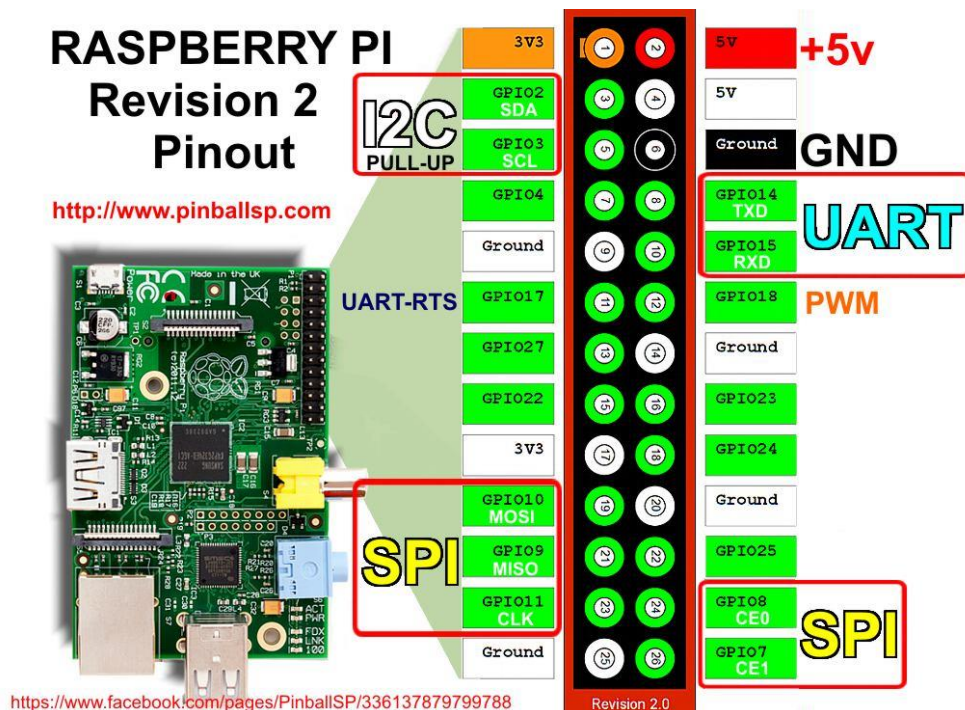
```
Serial.write(45);
int bytesSent = Serial.write("hello"); }
```

2.4.2.3. Komunikasi Serial Pada Raspberry

Komunikasi serial sering digunakan untuk mengirim data melalui dua board yang berbeda ataupun sama. Sebagai contohnya pengiriman data string dari raspberry ke arduino ataupun dari arduino ke avr board. Komunikasi serial ini tidak lepas dari penggunaan pin tx rx. Kebanyakan board sudah dilengkapi fasilitas tersebut.

TX merupakan singkatan dari Transmitter (pengirim), sedangkan RX merupakan singkatan dari Receiver (penerima). Semisal terdapat contoh kasus, Arduino akan mengirim data yang telah diperoleh dari sensor. Data-data sensor tersebut akan diproses dan dimonitoring didalam Raspberry. Pengiriman dari Arduino ke Raspberry ini menggunakan komunikasi serial. Maka yang harus dilakukan adalah pin TX pada Arduino disambungkan ke pin RX pada Raspberry, karena Arduino merupakan pengirim data dan Raspberry merupakan penerima.

Gambar dibawah ini merupakan pin GPIO yang ada pada RaspberryPi Revision 2 :



Gambar 2.5. TX (GPIO14) pin 8 dan RX (GPIO15) pin 10. [14]