

BAB II

KAJIAN PUSTAKA

2.1 Teori Tentang Tubuh Ideal

Sebagian besar dari masyarakat umum mengabaikan atau tidak memperhatikan perkembangan postur tubuh, namun bagi wanita ingin tampil sempurna di hadapan orang lain. Banyak cara orang untuk mendapatkan postur tubuh yang ideal baik secara sehat seperti, rutin berolah raga, minum susu, menjaga pola makan, tidur teratur menjaga pola hidup serta masih banyak lagi yang lainnya. Adapula dengan cara yang tidak sehat/tidak baik seperti, banyak alat yang bermunculan menarik badan secara manual karena dengan metode itu bisa membahayakan tulang balakang kita, makan-makanan berlebih sehingga muncul lemak jahat, pola tidur yang tidak teratur. Sehingga bisa dijelaskan bahwa menjaga kesehatan itu penting agar mendapatkan tubuh yang ideal.

Postur tubuh ideal, pada umumnya usia sekitar 18 tahun, dengan tinggi badan untuk wanita 160 cm dengan suhu badan 36,8°C sedangkan tinggi badan untuk laki-laki sekitar 165-175 cm 36,8 °C pada usia 21 tahun.

2.1.1. Perancangan Alat Pengukur Tinggi Badan Yang Sudah Ada

Dari referensi yang pertama didapat yaitu hasil Tugas Akhir dengan judul “Alat Ukur Tinggi Dan Berat Badan Manusia Secara Digital Berbasis Mikrokontroler AT89S51 Menggunakan Ultrasonik (Sensor Jarak) dan *Flexiforce* (Sensor Tekanan)”. Pada alat yang

dibuat, menggunakan 2 mikrokontroler AT89S51 dan sebagai penampil yaitu *LCD* 16x2 dan *Seven Segmen*.

Dan referensi kedua “Rancang Bangun Alat Pengukur Tinggi Badan Otomatis Dengan Keluaran Suara Berbasis Mikrokontroler AT89S51”. Pada alat ini motor *Stepper* sebagai media pengontrolan untuk pengukuran ketinggian obyek. Bentuk tampilan menggunakan keluaran suara.[14]

Tabel 2.1. Perbandingan Alat pengukur tinggi badan manusia

No	Nama Alat	Kelebihan	Kekurangan	Ket
1	Alat ukur tinggi badan manusia secara digital berbasis mikrokontroler AT89S51 menggunakan sensor ultrasonik dan <i>Flexiforce</i>	Tampilan <i>seven segmen</i> dan <i>LCD</i>	<ul style="list-style-type: none"> • Bentuk komunikasi yang digunakan untuk memantau hasil dari pengukuran pada alat dengan jarak jauh berupa <i>Bluetooth</i> sebagai media pemantauan hasil pengukuran • Menggunakan 2 tampilan dan mikrokontroler 	Tidak ada Kurang efisien
2	Rancang Bangun Alat Pengukur Tinggi Badan Otomatis Dengan Keluaran Suara Berbasis Mikrokontroler AT89S51	<ul style="list-style-type: none"> • Menggunakan motor <i>Stepper</i> sebagai pengukur ketinggian obyek • Tampilan menggunakan keluaran suara 	<ul style="list-style-type: none"> • Kemungkinan untuk menggunakan motor <i>Stepper</i> sistem kinerjanya agak lebih lama • Tampilan berupa <i>LCD</i> akan lebih baik 	Kurang efisien

Sehingga dari latang belakang alat yang didapat, penulis merancang sistem pada alat kali ini menggunakan komunikasi *Bluetooth* sebagai media untuk memantau hasil yang diperoleh dari alat, menggunakan *LCD 4x20* untuk menampilkan informasi dari tinggi badan, berat badan, suhu badan (dalam proses) dan detak jantung (sementara ini belum ada yang fokus pada sistem ini) sistem untuk mengukur ketinggian obyek masih manual, sehingga dokter/perawat dapat lebih mudah untuk memantau dan mengambil data dari pasien untuk menentukan status pada pasien.

2.2 Mikrokontroler

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah *chip*. Di dalamnya terkandung sebuah inti *processor*, *memory (RAM, memory program* atau keduanya) dan perlengkapan *input output*. Mikrokontroler merupakan sebuah prosesor yang digunakan untuk kepentingan kontrol. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan *computer mainframe*, mikrokontroler dibangun dari elemen-elemen dasar yang sama. 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.[6]

Beberapa fitur yang umumnya ada di dalam mikrokontroler adalah sebagai berikut :

1. *RAM (Random Access Memory)*

RAM digunakan oleh mikrokontroler untuk tempat penyimpanan variabel. Memori ini bersifat *volatile* (mudah berubah) yang berarti akan kehilangan semua datanya jika tidak mendapatkan *catu daya*. [6]

2. *ROM (Read Only Memory)*

ROM seringkali disebut sebagai kode memori karena berfungsi untuk tempat penyimpanan program yang akan diberikan oleh *user*. [6]

3. *Register*

Merupakan tempat penyimpanan nilai-nilai yang akan digunakan dalam proses yang telah disediakan oleh mikrokontroler. [6]

4. *Special Function Register*

Merupakan *register* khusus yang berfungsi untuk mengatur jalannya mikrokontroler. Register ini terletak pada *RAM*. [6]

5. *Input dan Output*

Pin input adalah bagian yang berfungsi sebagai penerima signal dari luar, pin ini dapat dihubungkan ke berbagai media *inputan* seperti *keypad*, sensor, dan sebagainya. *Pin output* adalah bagian yang berfungsi untuk mengeluarkan sinyal dari hasil proses algoritma mikrokontroler.

6. *Interrupt*

Interrupt bagian dari mikrokontroler yang berfungsi sebagai bagian yang dapat melakukan interupsi, sehingga ketika program utama sedang berjalan, program utama tersebut dapat diinterupsi dan menjalankan program interupsi terlebih dahulu. [6]

Beberapa *interrupt* pada umumnya adalah sebagai berikut :

- *Interrupt Eksternal*

Interrupt akan terjadi bila ada inputan dari pin *interrupt*.

- *Interrupt timer*

Interrupt akan terjadi bila waktu tertentu telah tercapai.

- *Interrupt serial*

Interrupt yang terjadi ketika ada penerimaan data dari komunikasi serial.[6]

2.2.1 Fitur ATMega 328

ATMega 328 adalah mikrokontroler keluaran dari ATMEL yang mempunyai arsitektur *RISC (Reduce Instruction Set Computer)* dimana setiap proses eksekusi data lebih cepat dari pada arsitektur *CISC (Completed Instruction Set Computer)*. Mikrokontroler ini memiliki beberapa fitur antara lain :[6]

- 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.
- 32 x 8-bit *register* serba guna.
- Kecepatan mencapai 16 MIPS dengan *clock* 16 MHz.
- 32 KB *Flash memory* dan pada *Arduino* memiliki *bootloader* yang digunakan 2 KB dari *flash memory* sebagai *bootloader*.
- Memiliki *EEPROM (Electrically Erasable Programmable Read Only Memory)* sebesar 1 KB sebagai tempat penyimpanan data semi permanent karena *EEPROM* tetap dapat menyimpan data meskipun *catu daya* dimatikan.

- Memiliki *SRAM (Static Random Access Memory)* sebesar 2 KB.
- Memiliki *pin I/O* digital sebanyak 14 *pin* 6 diantaranya *PWM (Pulse Width Modulation) output*.
- *Master / Slave SPI Serial interface*.

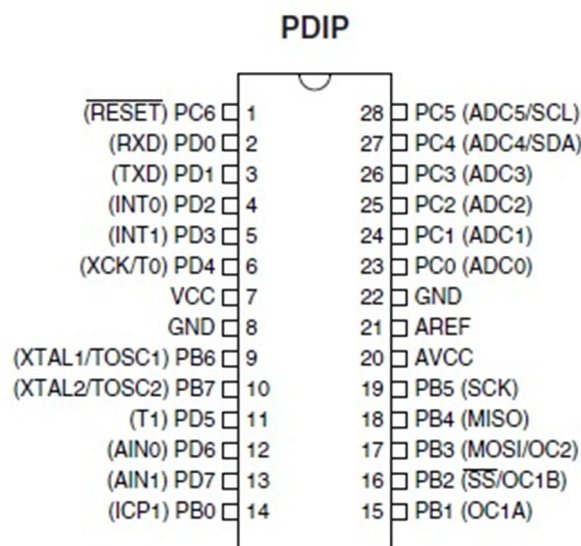
Mikrokontroler ATmega 328 memiliki arsitektur *Harvard*, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan kerja dan *parallelism*. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program. Konsep inilah yang memungkinkan instruksi-instruksi dapat dieksekusi dalam setiap satu siklus *clock*. [6]

32x8-bit *register* serba guna digunakan untuk mendukung operasi pada *ALU (Arithmetic Logic unit)* yang dapat dilakukan dalam satu siklus. 6 dari *register* serbaguna ini dapat digunakan sebagai 3 buah *register pointer 16-bit* pada *mode* pengalamatan tak langsung untuk mengambil data pada ruang memori data. Ketiga *register pointer 16-bit* ini disebut dengan *register X* (gabungan R26 dan R27), *register Y* (gabungan R28 dan R29), dan *register Z* (gabungan R30 dan R31). [6]

Hampir semua instruksi *AVR* memiliki format 16-bit. Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit. Selain *register* serba guna di atas, terdapat *register* lain yang

terpetakan dengan teknik *memory mapped I/O* selebar 64 *byte*. Beberapa *register* ini digunakan untuk fungsi khusus antara lain sebagai *register control Timer/Counter, Interupsi, ADC, USART, SPI, EEPROM*, dan fungsi *I/O* lainnya. *Register-register* ini menempati *memory* pada alamat 0x20h – 0x5Fh.[6]

2.2.2 Konfigurasi *PIN* ATMega 328



Gambar 2.2. Konfigurasi *pin* ATMega 328

1) **PORTB (PB7..PB0)**

PORTB adalah *port I/O* dua arah (*bidirectional*) 8-bit dengan *resistor pull-up internal* yang dapat dipilih. *Buffer* keluaran *port* ini memiliki karakteristik yang simetrik ketika digunakan sebagai *source* ataupun *sink*. Ketika digunakan sebagai *input*, *pin* yang di *pull-low* secara *eksternal* akan memancarkan arus jika *resistor pull-up*-nya diaktifkan. *Pin-pin*

PORTB akan berada pada kondisi *tri-state* ketika *Reset* aktif, meskipun *clock* tidak *running*. [6]

Tabel 2.2. Konfigurasi *Port B*

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	\overline{SS} (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

2) *PORTC (PC5..PC0)*

PORTC adalah *port I/O* dua arah (*bidirectional*) 7-bit dengan *resistor pull-up internal* yang dapat dipilih. *Buffer* keluaran *port* ini memiliki karakteristik yang simetrik ketika digunakan sebagai *source* ataupun *sink*. Ketika digunakan sebagai *input*, *pin* yang di *pull-low* secara *eksternal* akan memancarkan arus jika *resistor pull-up*-nya diaktifkan. *Pin-pin*

PORTC akan berada pada kondisi *tri-state* ketika *Reset* aktif, meskipun *clock* tidak *running*. [6]

3) *PC6/RESET*

Jika *Fuse RSTDISBL* diprogram, maka *PC6* berfungsi sebagai *pin I/O* akan tetapi dengan karakteristik yang berbeda dengan *PC5..PC0*. Jika *Fuse RSTDISBL* tidak diprogram, maka *PC6* berfungsi sebagai masukan *Reset*. Sinyal *LOW* pada pin ini dengan lebar minimum 1,5 mikrodetik akan membawa mikrokontroler ke kondisi *Reset*, meskipun *clock* tidak *running*.

Tabel 2.3. Konfigurasi *Port C*

Port Pin	Alternate Function
PC6	RESET (Reset pin) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

4) **PORTD (PD7..PD0)**

PORTD adalah *port I/O* dua arah (*bidirectional*) 8-bit dengan *resistor pull-up internal* yang dapat dipilih. *Buffer* keluaran *port* ini memiliki karakteristik yang simetrik ketika digunakan sebagai *source* ataupun *sink*. Ketika digunakan sebagai *input*, *pin* yang di *pull-low* secara *eksternal* akan memancarkan arus jika *resistor pull-upnya* diaktifkan. *Pin-pin PORTD* akan berada pada kondisi *tri-state* ketika *Reset* aktif, meskipun *clock* tidak *running*. [6]

Tabel 2.4. Konfigurasi *Port D*

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

5) **RESET**

Pin masukan *Reset*. Sinyal *LOW* pada *pin* ini dengan lebar minimum 1,5 mikrodetik akan membawa mikrokontroler ke kondisi *Reset*, meskipun *clock* tidak *running*. Sinyal dengan lebar kurang dari 1,5 mikrodetik tidak menjamin terjadinya kondisi *Reset*. [6]

6) **AVCC**

AVCC adalah *pin* suplai tegangan untuk *ADC*, *PC3..PC0*, dan *ADC7..ADC6*. *Pin* ini harus dihubungkan dengan *VCC*, meskipun *ADC* tidak digunakan. Jika *ADC* digunakan, *VCC* harus dihubungkan ke *AVCC* melalui *low-pass filter* untuk mengurangi *noise*. [6]

7) **AREF** *Pin* Analog Reference untuk *ADC*. [6]

8) **ADC7..ADC6** Analog *input* *ADC*. Hanya ada pada ATmega8 dengan *package* *TQFP* dan *QFP/MLF*. [6]

2.2.3 **Arduino**

Arduino didefinisikan sebagai sebuah *platform* elektronik yang *open source*, berbasis pada *software* dan *hardware* yang fleksibel dan mudah digunakan, yang ditujukan untuk seniman, *desainer*, *hobbies* dan setiap orang yang tertarik dalam membuat objek atau lingkungan yang interaktif. [4]

Arduino sebagai sebuah *platform* komputasi fisik (*Physical Computing*) yang *open source* pada *board input output* sederhana, yang dimaksud dengan *platform* komputasi fisik disini adalah sebuah sistem fisik yang interaktif dengan penggunaan *software* dan *hardware* yang dapat mendeteksi dan merespon situasi dan kondisi.[4]

Kelebihan *Arduino* dari *platform hardware* mikrokontroler lain adalah :

- *IDE Arduino* merupakan *multiplatform*, yang dapat dijalankan di berbagai sistem operasi, seperti *Windows*, *Macintosh* dan *Linux*.
- *IDE Arduino* dibuat berdasarkan pada *IDE Processing*, yang sederhana sehingga mudah digunakan.[4]

Pemrograman *arduino* menggunakan kabel yang terhubung dengan *port USB*, bukan *port serial*. *Fitur* ini berguna karena banyak komputer yang sekarang ini tidak memiliki *port serial*. *Arduino* adalah *hardware* dan *software open source* pembaca bisa *download software* dan gambar rangkaian *Arduino* tanpa harus membayar ke pembuat *Arduino*.[4]

Biaya *hardware* cukup murah, sehingga tidak terlalu menakutkan untuk membuat kesalahan. Proyek *Arduino* ini dikembangkan dalam lingkungan pendidikan sehingga bagi pemula akan lebih cepat dan mudah mempelajarinya.[4]

➤ Sejarah *Arduino*

Proyek *Arduino* dimulai pertama kali di *Ovre, Italy* pada tahun 2005. Tujuan proyek ini awalnya untuk membuat peralatan kontrol interaktif dan modul pembelajaran bagi siswa yang lebih murah dibandingkan dengan *prototype* yang lain. Pada tahun 2010 telah terjual dari 120 unit *Arduino*. *Arduino* yang berbasis *open source* melibatkan tim pengembang. Pendiri *Arduino* itu *Massimo Banzi* dan *David Cuartielles*, awalnya mereka memberi nama proyek itu dengan sebutan *arduin* dari *ivrea* tetapi seturut perkembangan zaman nama proyek itu diubah menjadi *Arduino*. [4]

Arduino dikembangkan dari *thesis hernando Barragan* di desain interaksi *institute Ivrea*. *Arduino* dapat menerima masukan dari berbagai macam sensor dan juga dapat mengontrol lampu, motor dan *aktuator* lainnya. Mikrokontroler pada *board arduino* di program dengan menggunakan bahasa pemrograman *Arduino (based on wiring)* dan *IDE Arduino (based on processing)*. Proyek *Arduino* dapat berjalan sendiri atau juga bisa berkomunikasi dengan *software* yang berjalan pada komputer. [4]

2.2.3.1 *Arduino Uno*

Arduino Uno merupakan *board Arduino* revisi terbaru yang merupakan penerus dari *Arduino Duemilanove*. Yang membedakan antara *Arduino Uno* dan *Arduino duemilanove* yaitu

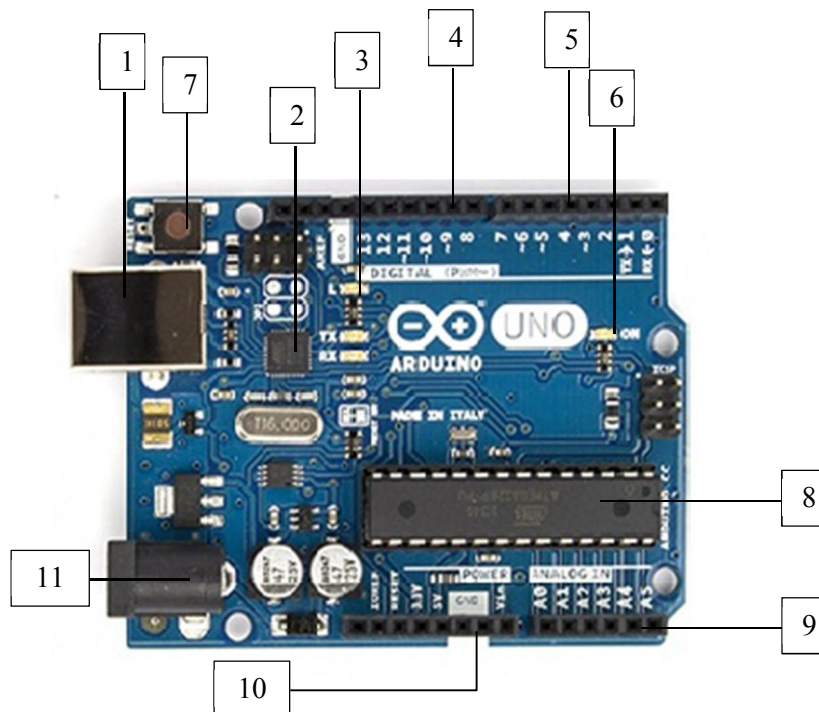
tidak lagi digunakannya *chip FTDI (USB to Serial driver)* dan sebagai gantinya menggunakan ATmega 8U2 yang diprogramkan untuk berfungsi sebagai *konverter USB to Serial*. Perubahan ini cukup membantu dalam instalasi *software Arduino*, terutama bagi anda yang memakai sistem operasi *windows*, karena tidak perlu menginstal *driver FTDI* untuk menghubungkan *board Arduino Uno* dengan *windows*. [4]

Arduino Uno merupakan *board* mikrokontroler berbasis ATmega 328 yang memiliki 14 *pin input* dari *output* digital dimana 6 *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 mikrokontroler agar dapat digunakan cukup hanya menghubungkan *board Arduino Uno* ke komputer dengan menggunakan kabel *USB* atau listrik dengan *AC (Alternative Current)* yang ke adaptor *DC (Direct Current)* atau baterai untuk menjalankannya. [4]

(1) Hardware

Arduino Uno terdiri dari mikrokontroler *MegaAVR* seperti ATmega 8, ATmega 168, ATmega 328, ATmega 1280 dan ATmega 2560 dengan menggunakan kristal *osilator* 16 MHz, namun ada juga *Arduino* yang menggunakan kristal *osilator* 8 MHz. *Catu daya* yang dibutuhkan untuk men-*supply minimum system Arduino* cukup dengan tegangan 5 *VDC*. *Port*

Arduino ATmega terdiri dari 20 *pin* yang meliputi 14 *pin I/O* digital dengan 6 *pin* dapat berfungsi sebagai *output PWM* (*Pulse Width Modulation*) dan 6 *pin I/O* analog. Kelebihan *Arduino* ini tidak membutuhkan *flash programmer external* karena di dalam *chip* mikrokontroler *Arduino* telah diisi dengan *bootloader* yang membuat proses *upload* menjadi lebih sederhana. Untuk koneksi terhadap komputer dapat menggunakan *RS232 to TTL Converter* atau menggunakan *chip USB* ke serial converter seperti *FTDI FT232*. Berikut ini menunjukkan gambar dari *Arduino Uno* dan bagian-bagian komponen beserta keterangannya.[4]



Gambar 2.3 *Arduino Uno*

Tabel 2.5. Keterangan dari *Arduino Uno*

No	Keterangan
1	<i>Port USB</i>
2	<i>IC konverter Serial-USB</i>
3	<i>LED untuk tes output kaki</i>
4	Kaki-Kaki <i>Input Output</i> Digital (D8-D13)
5	Kaki-Kaki <i>Input Output</i> Digital (D0-D7)
6	<i>LED Indikator catu daya</i>
7	Tombol <i>Reset</i>
8	Mikrokontroler ATmega 328
9	Kaki-Kaki <i>Input</i> Analog (A0-A5)
10	Kaki-Kaki <i>catu daya</i> (5 volt, GND)
11	Terminal <i>catu daya</i> (6-9 volt)

Tabel 2.6. Ringkasan *Arduino Uno*

Mikrokontroler	ATmega 328
Tegangan pengoperasian	5V
Tegangan <i>input</i> yang disarankan	7-12V
Batas tegangan <i>input</i>	6-20V
Jumlah <i>pin I/O</i> digital	14 (6 di antaranya menyediakan keluaran <i>PWM</i>)
Jumlah <i>pin input</i> analog	6
Arus <i>DC</i> tiap <i>pin I/O</i>	40 mA
Arus <i>DC</i> untuk <i>pin 3.3V</i>	50 mA
<i>Memory Flash</i>	32 KB (ATmega328), sekitar 0.5 KB digunakan oleh <i>bootloader</i>
<i>SRAM</i>	2 KB (ATmega 328)
<i>EEPROM</i>	1 KB (ATmega 328)
<i>Clock Speed</i>	16 MHz

(2) *Software*

Software arduino yang digunakan adalah *driver* dan *IDE*, walaupun masih ada beberapa *software* lain yang sangat berguna selama pengembangan *Arduino*. *IDE Arduino* adalah *software* yang sangat canggih ditulis dengan menggunakan *java*. *IDE Arduino* terdiri dari :[4]

➤ Editor Program

Sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.

➤ *Compiler*

Sebuah modul yang mengubah kode program menjadi kode biner bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa *processing*.

➤ *Uploader*

Sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam papan *Arduino*.

Dalam bahasa pemrograman *Arduino* ada tiga bagian utama yaitu struktur, variabel dan fungsi :

1. Struktur Program *Arduino*

a. Kerangka Program

Kerangka program *Arduino* sangat sederhana, yaitu terdiri atas dua blok. Blok pertama

adalah *void setup()* dan blok kedua adalah *void loop()*.^[3]

- *Blok void setup()*

Berisi kode program yang hanya dijalankan sekali sesaat setelah *Arduino* dihidupkan atau di-*reset*. Merupakan bagian persiapan atau instalasi program.^[4]

- *Blok void loop()*

Berisi kode program yang akan dijalankan terus menerus. Merupakan tempat untuk program utama.^[4]

b. Sintaks Program

Baik blok *void setup loop()* maupun blok *function* harus diberi tanda kurung kurawal buka “{” sebagai tanda awal program di blok itu dan kurung kurawal tutup “}” sebagai tanda akhir program.^[3]

```
void setup() {
    // perintah-perintah untuk
    konfigurasi dan inisialisasi
    arduino board
}

void loop() {
    // perintah - perintah utama
    arduino board
}
```

Gambar 2.4. Struktur program *Arduino*

2. Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas dengan menggunakan sebuah variabel.[3]

```
Type variableName = 0;
```

Gambar 2.5. Contoh variabel *Arduino*

1. Fungsi

Pada bagian ini meliputi fungsi *input output* digital, *input output* analog, *advanced I/O*, fungsi waktu, fungsi matematika serta fungsi komunikasi.[4]

```
Type functionName (parameters)
    {
        // Statement;
    }
```

Gambar 2.6. Contoh fungsi *Arduino*

- Digital *I/O*

Digital *input* memberikan dua kondisi sinyal masukan yaitu tombol tertekan atau tombol tidak tertekan. Pada saat tombol tertekan akan memberikan tegangan 5 atau 9 *volt* pada masukan sedangkan sebaliknya pada saat tombol dilepas hanya memberikan tegangan 0 *volt*. Kondisi ini disebut dengan digital *input* dengan logika 1 dan 0,

dimana 1 itu untuk tegangan 5 *volt* dan 0 untuk tegangan 0 *volt*. Begitu juga halnya pada sisi *output*, jika hanya melibatkan dua kondisi keluaran seperti menghidupkan dan mematikan suatu sensor jika diberi tegangan ini disebut dengan digital *output* dimana digital 1 untuk yang diberi tegangan dan digital 0 dengan *output* tegangan 0 *volt*. [3]

a. *pinMode(pin, mode)*

Digunakan dalam *void setup()* untuk mengkonfigurasi *pin* apakah sebagai *Input* atau *Output*. *Arduino* digital *pins* secara *default* di konfigurasi sebagai *input* sehingga untuk merubahnya harus menggunakan operator *pinMode(pin, mode)*. [3]

Parameter: *pin* = angka dari *pin* digital yang akan dikonfigurasi.

mode = konfigurasi yang diinginkan (*Input*, *Input_Pullup* dan *Output*).

```
pinMode (pin, Output);    // mengset pin
sebagai output
digitalWrite (pin, High); // pin sebagai
source voltage
```

Gambar 2.7. Contoh *pin Mode*

b. *digitalRead(pin)*

Membaca nilai dari *pin* yang kita kehendaki dengan hasil *High* atau *Low*.

Parameter: *pin* = angka dari *pin* digital yang akan dibaca. Berikut contoh penggunaan fungsi masukan dan keluaran digital dalam sebuah program : [4]

```
int ledPin = 13; // LED terhubung ke pin digital 13
int inPin = 7; // pushbutton terhubung ke pin digital 7
int val = 0; // variable untuk menyimpan sebuah nilai
void setup() {
pinMode(ledPin, OUTPUT); // set pin digital 13 sebagai
keluaran
pinMode(inPin, INPUT); // set pin digital 13 sebagai masukan
}
void loop(){
val = digitalRead(inPin); // baca nilai pin input
digitalWrite(ledPin, val); // set LED sesuai dengan nilai val
}
```

Gambar 2.8. Contoh *Input Output digital Read*

c. *digitalWrite (pin, value)*

Digunakan untuk mengset *pin* digital.

Pin digital *Arduino* mempunyai 14 (0 – 13).[3]

```
digitalWrite (pin, High); // set
pin to High
```

Gambar 2.9. Contoh *Digital Write*

- *Analog I/O*

Analog input output merupakan pengolahan *input* dan *output* secara digital mungkin sudah memenuhi suatu kebutuhan, akan tetapi pada kondisi tertentu ada kemungkinan yang dihadapi pada kondisi *input* dan *output* yang membutuhkan besaran yang berubah-ubah dengan nilai yang kontinyu dan tidak lagi hanya dengan dua keadaan seperti sinyal digital. *Arduino* dalam analog *I/O* digunakan sebagai kontroler yang mampu mengolah semua variasi tegangan keluaran dari sensor yang dihubungkan pada *pin inputnya*. Perintah yang digunakan dalam analog *I/O* ini adalah : [4]

a. *analog Read (pin)*

Membaca nilai *pin* analog yang memiliki resolusi 10-bit. Fungsi ini hanya dapat bekerja

pada analog *pin* (0-5). Hasil dari pembacaan berupa nilai *integer* dengan *range* 0 sampai 1023.[4]

```
Value = analogRead(pin); // mengset
'value' sama dengan nilai analog pin
```

Gambar 2.10. Contoh analog *Read*

b. analog *Write* (*pin*, *value*)

Mengirimkan nilai analog pada *pin* analog.[4]

```
analogWrite(pin, value); // menulis
ke pin analog
```

Gambar 2.11. Contoh analog *Write*

➤ **Power**

Arduino dapat diberikan *power* melalui koneksi *USB* atau *power supply*. *Powernya* *diselect* secara otomatis. *Power supply* dapat menggunakan adaptor *DC* atau baterai. Adaptor dapat dikoneksikan dengan mencolok *jack adaptor* pada koneksi *port input supply*. *Board Arduino* dapat dioperasikan menggunakan *supply* dari luar sebesar 6-20 *volt*. Jika *supply* kurang dari 7 *volt*, kadangkala *pin 5 volt* akan menyuplai kurang dari 5 *volt* dan *board* bisa menjadi tidak stabil. Jika menggunakan

lebih dari 12 *volt*, tegangan di regulator bisa menjadi sangat panas dan menyebabkan kerusakan pada *board*. Rekomendasi tegangan ada pada 7 *volt* sampai 12 *volt*. [4]

Penjelasan pada pin power adalah sebagai berikut :

- ***Vin***

Tegangan *input* ke *board Arduino* ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 *volt* dari koneksi *USB* atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui *pin* ini, atau jika tegangan suplai menggunakan *power jack*, aksesnya menggunakan *pin* ini. [4]

- **5 *volt***

Regulasi power *supply* digunakan untuk power mikrokontroler dan komponen lainnya pada *board*. 5 *volt* dapat melalui *Vin* menggunakan *regulator* pada *board*, atau *supply* oleh *USB* atau *supply* regulasi 5 *volt* lainnya. [4]

- **3V3**

Suplai 3.3 *volt* didapat oleh *FTDI chip* yang ada di *board*, arus maximumnya adalah 50mA. [4]

- ***Pin Ground*** berfungsi sebagai jalur *ground* pada *Arduino*. [4]

➤ **Memori**

ATMega328 memiliki 32 *KB flash* memori untuk menyimpan kode, juga 2 *KB* yang digunakan untuk *bootloader*. ATMega328 memiliki 2 *KB* untuk *SRAM* dan 1 *KB* untuk *EEPROM*. [4]

➤ **Input dan Output**

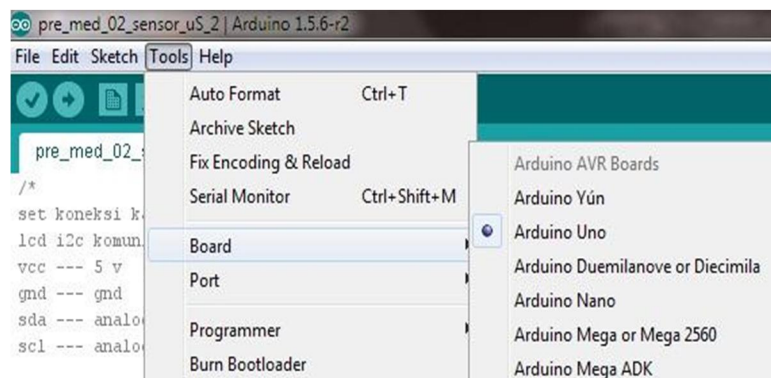
Setiap 14 *pin* digital pada *Arduino* dapat digunakan sebagai *input* atau *output*, menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. *Input/output* dioperasikan pada 5 *volt*. Setiap *pin* dapat menghasilkan atau menerima maximum 40mA dan memiliki *internal pull-up resistor (disconnected oleh default)* 20-50 *KOhms*. Beberapa *pin* memiliki fungsi sebagai berikut : [4]

- Serial : 0 (*RX*) dan 1 (*TX*). Digunakan untuk menerima (*RX*) dan mengirim (*TX*) *TTL* data serial. *Pin* ini terhubung pada *pin* yang *koresponding* dari *USB FTDI* ke *TTL chip* serial. [4]
- *Interrupt eksternal* : 2 dan 3. *Pin* ini dapat dikonfigurasi untuk *trigger* sebuah *interap* pada *low value*, *rising* atau *falling edge*, atau perubahan nilai. [4]
- *PWM* : 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit keluaran *PWM* dengan fungsi *analogWrite()*.

- *SPI* : 10 (*SS*), 11 (*MOSI*), 12 (*MISO*), 13 (*SCK*). *Pin* ini *men-support* komunikasi *SPI* (*Serial Pheriperal Interface*), yang mana masih mendukung *hardware*, yang tidak termasuk pada bahasa *Arduino*. [4]
- *LED* : 13. Ini adalah dibuat untuk koneksi *LED* ke digital *pin* 13. Ketika *pin* bernilai *HIGH*, *LED* hidup, ketika *pin* *LOW*, *LED* mati. [4]

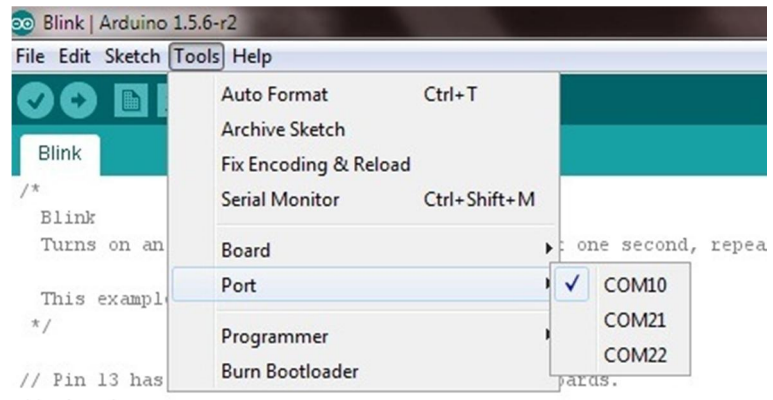
➤ **Cara upload Programnya**

Setelah program yang disusun benar dan berhasil *compiling*, sebelumnya *Arduino Uno* tetap dalam kondisi tersambung/*connect* pada *USB Arduino*. Kemudian pilih *Tools* selanjutnya pilih *Board* kemudian *Arduino Uno* untuk yang *ATMega 328*, sedangkan untuk yang menggunakan *ATMega 8* pilih *Tools* selanjutnya pilih *Borad* kemudian *Arduino NG or older* setelah itu kembali ke menu *Tools* pilih *Processor ATMega 8*, ini untuk menyesuaikan dengan yang digunakan oleh pengguna.



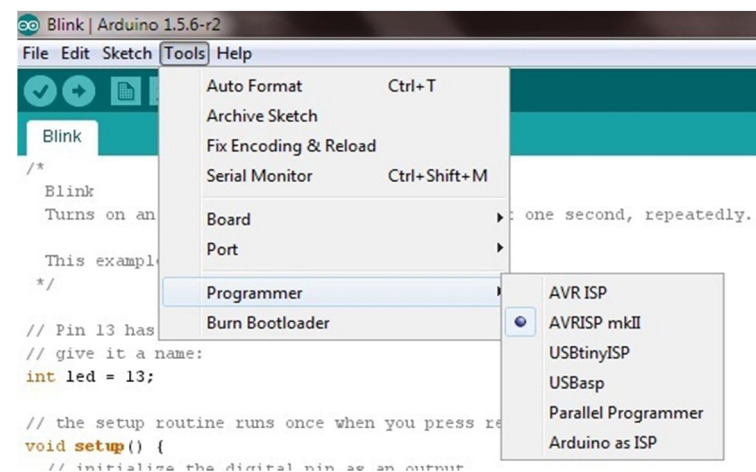
Gambar 2.12. Tampilan *Board* yang digunakan

Setelah menentukan *Board*-nya, selanjutnya untuk menentukan *Port*. Ini berfungsi untuk menentukan *COM* berapa yang telah diinstall *driver USB* sebelumnya untuk menghubungkan *USB* dengan *Board Arduino Uno*.



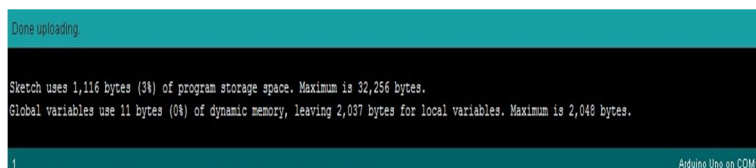
Gambar 2.13. Tampilan untuk menentukan *Port*

Kemudian tentukan *Programmer*, ini sebagai media untuk meng-*upload* sebuah program agar cocok dengan *Serial Programmer*.



Gambar 2.14. Tampilan untuk menentukan *Programmer*

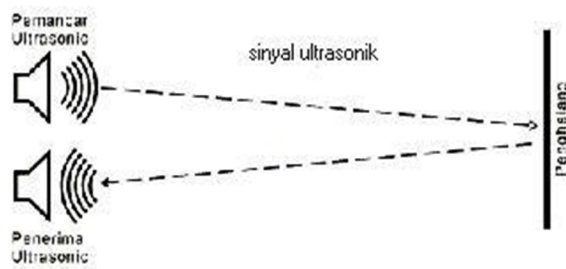
Selanjutnya untuk meng-*upload* programnya pilih menu *File* kemudian *Upload* atau pada *menu bar* terdapat simbol panah kanan kemudian klik, tunggu sampai *Done uploading*. Setelah sukses, maka program yang telah dibuat sudah terisi pada mikrokontroler yang digunakan.



Gambar 2.15. Tampilan setelah mengupload Program

2.3 Sensor Ultrasonik

Gelombang ultrasonik adalah gelombang dengan besar frekuensi diatas frekuensi gelombang suara yaitu lebih dari 20 KHz. Seperti telah disebutkan bahwa sensor ultrasonik terdiri dari pemancar ultrasonik yang disebut *transmitter* dan penerima ultrasonik yang disebut *receiver*. Sinyal ultrasonik yang dibangkitkan akan dipancarkan dari *transmitter* ultrasonik. Ketika sinyal mengenai benda penghalang, maka sinyal ini dipantulkan dan diterima oleh *receiver* ultrasonik. Sinyal yang diterima oleh *receiver* dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap objek yang di depannya (bidang pantul).[8]



Gambar 2.3. Prinsip kerja Sensor Ultrasonik

Prinsip kerja dari sensor ultrasonik adalah sebagai berikut :

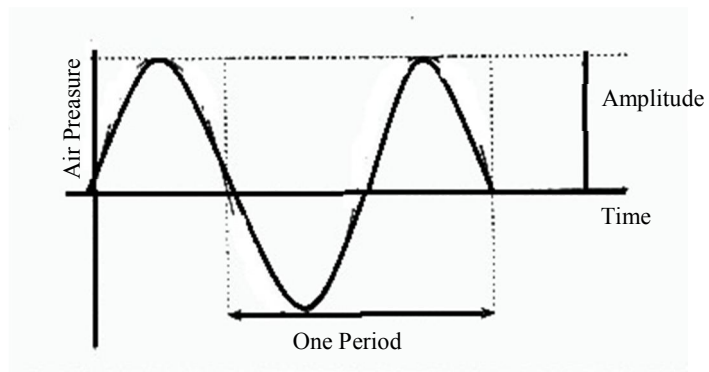
1. Sinyal dipancarkan oleh pemancar ultrasonik yaitu *transmitter*. Sinyal tersebut berfrekuensi diatas 20 kHz , biasanya yang digunakan untuk mengukur jarak benda/objek adalah 40 kHz . [8]
2. Sinyal yang dipancarkan tersebut kemudian akan merambat sebagai sinyal/gelombang bunyi. Sinyal tersebut kemudian akan dipantulkan dan akan diterima kembali oleh bagian penerima ultrasonik yaitu *reciever*. [8]
3. Setelah sinyal tersebut sampai di penerima ultrasonik, kemudian sinyal tersebut akan diproses untuk menghitung jaraknya. Jarak dihitung berdasarkan rumus : $((\text{durasi}/2.00)/29.10)$ dimana durasi adalah jarak antara sensor ultrasonik dengan bidang pantul.

Sensor ultrasonik yang digunakan dalam pengujian ini adalah jenis sensor ultrasonik *HC-SR04*. Sensor ini memiliki banyak kesamaan dengan sensor *HY-SRF05*. [8]

2.3.1. Kecepatan Suara

Kecepatan suara adalah istilah yang digunakan untuk menyebut kecepatan gelombang suara yang merambat pada medium elastis. Kecepatan rambatan gelombang suara ini dapat berbeda tergantung medium yang dilewati (misalnya suara lebih cepat melalui air dari pada udara), sifat-sifat medium tersebut, masa jenis air dan suhu. Namun, istilah ini lebih banyak dipakai untuk kecepatan suara di udara. Pada ketinggian air laut, dengan suhu 21 °C dan kondisi atmosfer normal, kecepatan suara adalah **344 m/detik (1238 km/jam)**. Kecepatan suara akan lebih cepat melaju di air dan di benda padat. Kecepatan suara di air adalah 4.3 kali lipat kecepatan di udara, yaitu 1.484 m/detik. Kecepatan suara di besi adalah 15 kali lipat kecepatan di udara, yaitu 5.120 m/detik.[12]

Suara berhubungan erat dengan rasa “mendengar”. Suara/bunyi biasanya merambat melalui udara. Suara/bunyi tidak bisa merambat melalui ruang hampa. Gelombang suara bervariasi dalam tingkatan tekanan suara (amplitudo) dan dalam frekuensi. Jumlah waktu yang diperlukan untuk terjadinya suatu getaran atau gelombang dinamakan perioda (T). Sedangkan jumlah gelombang yang terjadi setiap detik dinamakan frekuensi (f) dengan satuan m/dt (Hz).[13]



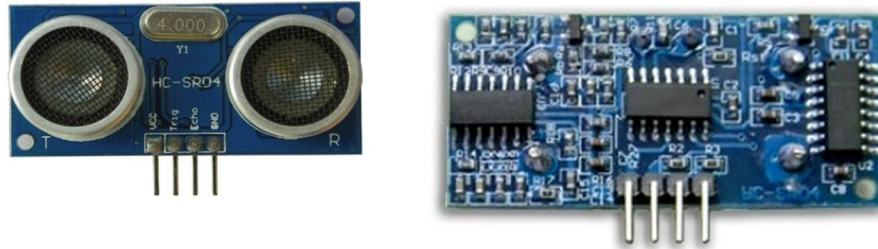
Gambar 2.4. Gelombang Suara

Tabel 2.3. Spesifikasi Sensor Ultrasonik

	HC-SR04	HY-SRF05
<i>Working Voltage</i>	5 VDC	5 VDC
<i>Static current</i>	< 2mA	< 2 mA
<i>Output signal</i>	<i>Electric frequency signal, high level 5V, low level 0V</i>	<i>Electric frequency signal, high level 5V, low level 0V</i>
<i>Sensor angle</i>	< 15 degrees	< 15 degrees
<i>Detection distance (claimed)</i>	2cm-450cm	2cm-450cm
<i>precision</i>	~3 mm	~2 mm
<i>Input trigger signal</i>	10us TTL impulse	10us TTL impulse
<i>Echo signal</i>	output TTL PWL signal	output TTL PWL signal
<i>Pins</i>	<ol style="list-style-type: none"> 1. VCC 2. trig(T) 3. echo(R) 4. GND 	<ol style="list-style-type: none"> 1. VCC 2. trig(T) 3. echo(R) 4. OUT 5. GND

Untuk menghitung nilai jarak, dengan menggunakan informasi di atas, yaitu

$$\text{Jarak} = ((\text{durasi}/2.00)/29.10)$$



Gambar 2.5 Bentuk fisik dari sensor ultrasonik

2.3.2 Gelombang Ultrasonik

Gelombang ultrasonik adalah gelombang *sinusoidal* dengan besar frekuensi di atas frekuensi gelombang suara yaitu lebih dari 20 *KHz* sampai dengan 500 *KHz* seperti yang dijelaskan di atas. Sensor ultrasonik bekerja berdasarkan prinsip pantulan gelombang suara, dimana sensor ini menghasilkan gelombang *sinusoidal* yang kemudian menangkapnya kembali dengan perbedaan waktu sebagai dasar pengindraanya. Perbedaan waktu antara gelombang ultrasonik yang dipancarkan dengan ditangkapnya kembali gelombang tersebut merupakan representasi jarak. Kecepatan rambat gelombang ultrasonik dipengaruhi oleh beberapa faktor antara lain suhu, tekanan, kelembapan, dan *signal* radio.

Beberapa hal yang penting dalam pemantulan gelombang sensor ultrasonik adalah sebagai berikut :

- **Bahan atau jenis material**

Objek padat akan memantulkan gelombang yang jauh

lebih baik dari pada objek cair. Objek padat terbuat dari besi baja akan memberikan pemantulan *echo* yang lebih baik dibandingkan objek padat kayu atau gabus, dikarenakan kayu sedikit-banyak dapat menyerap gelombang yang mengenai dan dipantulkan kembali. Dalam hal ini objek manusia cukup baik dalam proses pemantulan gelombang ultrasonik.

- **Pola permukaan objek**

Objek dengan permukaan datar, halus dan tegak lurus dengan sinyal ultrasonik akan memberikan pemantulan (*echo*) yang lebih baik dibanding dengan permukaan yang tidak rata dan posisi objek tidak mendukung pemantulan gelombang secara sempurna.

- **Ukuran dan bentuk objek**

Objek dengan ukuran yang lebih besar akan memantulkan lebih banyak gelombang dibandingkan dengan objek yang lebih kecil sehingga memberikan hasil yang tidak baik bahkan sampai tidak terdeteksi adanya suatu objek.

Objek dengan bentuk bola dapat memantulkan gelombang ultrasonik ke segala arah sehingga pemantulan gelombang yang diterima oleh sensor ultrasonik *HC-SR04* menjadi lemah atau tidak baik dan sebaliknya jika objek dengan bentuk datar akan memantulkan gelombang yang lebih baik atau sempurna.

2.3.3 Efek *Doppler*

Radar *Bistatic* adalah jenis sistem radar yang mempunyai komponen pemancar sinyal (*transmitter*) dan penerima sinyal (*receiver*) dipisahkan oleh suatu jarak yang dapat dibanding dengan jarak target/objek. Objek dideteksi berdasarkan pantulan sinyal dari objek tersebut ke pusat antena. Contoh Radar *Bistatic* yaitu *Passive* radar.

Pada alat sebelumnya yang sudah ada dibuat, menggunakan sistem kontrol otomatis untuk mengetahui jarak obyek. Pada alat yang dibuat kali ini menggunakan papan sebagai bidang pantul untuk mengetahui jarak obyek. Jadi pada alat tersebut, papan yang dibuat di atas kepala obyek secara manual mengukur tinggi badan obyek dengan meletakkan sensor disepanjang lintasan papan.

Jadi pada alat yang dibuat kali ini adalah menggunakan sensor ultrasonik sebagai pendeteksi tinggi badannya.

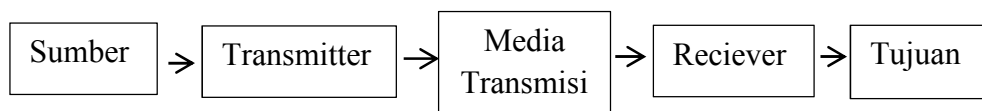
2.3.4 Perhitungan Tinggi Badan

Gelombang ultrasonik ini melalui udara dengan kecepatan 200 cm per detik, mengenai obyek dan memantul kembali ke sensor. Pada sensor bagian 1 ultrasonik *HC-SR04* ini akan mengeluarkan pulsa *output low* pada *trigger pin* dan *delay* yang diberikan 2 milidetik, setelah memancarkan gelombang ultrasonik dan gelombang pantulan terdeteksi, sensor *HC-SR04* akan membuat *output High* pada *trigger pin* dengan

memberikan *delay 5 milidetik* , begitu sebaliknya pada sensor ultrasonik bagian 2 tetapi *delay* yang diberikan 5 milidetik dan 2 milidetik. Durasi *High (tIN)* akan sesuai dengan lama waktu tempuh gelombang ultrasonik untuk 2x jarak ukur dengan obyek. Maka jarak yang diukur adalah $[(\text{durasi pulsa masuk}/2.00) / 29.10]$ cm.

2.4 Komunikasi Data

Komunikasi data merupakan proses pentransmisian data secara elektronik melalui media berupa kabel maupun tanpa kabel. Transmisi data adalah proses yang terjadi antara *transmitter* dan *receiver* melalui suatu media transmisi yang diklasifikasikan sebagai *guided* (pemandu) media, mengacu pada media transmisi fisik berupa kabel, dan *unguided* (tidak terpandu) media, mengacu pada media transmisi tanpa kabel. Tujuan fundamental dari sebuah sistem komunikasi data adalah pertukaran data secara elektronik antara dua pihak. Secara umum, model komunikasi data dapat diilustrasikan seperti gambar di bawah ini, yang merupakan sebuah model sistem komunikasi data yang telah disederhanakan.



Gambar 2.4 Model Komunikasi Data

Elemen sumber pada gambar di atas merupakan suatu perangkat yang membangkitkan data untuk ditransmisikan. Informasi yang diterima

akan diubah kedalam bentuk data-data digital dengan tingkat ketelitian tertentu sesuai dengan kebutuhan. *Transmitter* akan melakukan perubahan dan pengkodean informasi untuk membangkitkan suatu gelombang elektromagnetik yang dapat ditransmisikan melalui suatu media transmisi. Media transmisi dapat berupa jalur - jalur kabel (*wire*) atau tanpa kabel (*wireless*) sebagai penghubung antara sumber dan tujuan. *Receiver* akan menerima sinyal dari sistem transmisi data dan mengubah sinyal tersebut kedalam bentuk yang dapat ditangani oleh perangkat tujuan. Bentuk informasi yang diterima pada perangkat tujuan bisa jadi tidak sama dengan bentuk informasi pada sumber. Sistem penerimaan dimungkinkan untuk melakukan perubahan terhadap data pada penerima kedalam bentuk yang berbeda dengan data dari sumber.

Berikut komunikasi data dengan transmisi kabel maupun tanpa kabel :

1. Menggunakan *Wire* (kabel) merupakan media transmisi *guided* (terpandu), disebut *guided* karena dapat mentransmisikan serta memandu arah gelombang kepada penerima data. Media ini biasanya digunakan jika jarak sumber dan penerima tidak terlalu jauh.

Media transmisi data yang berupa kabel ada tiga jenis, yaitu :

- Kabel Koaksial (*Coaxial Cable*) : terdiri dari dua buah konduktor yang terpisah oleh *isolator*.
- Kabel Dua Kawat (*Twisted Pair Cable*) : Terdiri beberapa pasang kawat yang digulung dalam satu bendel.

- Kabel Serat Optik (*Fiber Optik Cable*) : Kabel yang mengirimkan informasi dengan cara menghantarkan informasi dalam gelombang cahaya.
2. *Wireless* adalah teknologi tanpa kabel, dalam hal ini adalah melakukan komunikasi data/*transfer* data dengan menggunakan gelombang elektromagnetik sebagai pengganti kabel. *Wireless* (tanpa kabel) menggunakan gelombang elektromagnetik (radio dan infra merah) untuk melakukan komunikasi data menyalurkan data dari satu poin ke poin yang lain tanpa melalui fasilitas fisik. Koneksi ini menggunakan frekuensi tertentu untuk menyalurkan data tersebut, kebanyakan *Wireless* (tanpa kabel) menggunakan frekuensi 2,4 GHz. Frekuensi inilah yang disebut dengan *Industrial, Scientific* dan *Medical Band* atau sering disebut *ISM Band*, seperti telah dijelaskan di atas. Penggunaannya dapat anda lihat dalam penggunaan modem *GSM, CDMA*.

Berikut hal yang dilakukan dengan media tanpa kabel :

1. *InfraRed* merupakan salah satu jenis gelombang elektromagnetik yang berbentuk cahaya. Implementasi *Infra* merah untuk keperluan komunikasi antara lain :
 - Komunikasi data nirkabel jarak dekat menggunakan *IrDA* (*Infrared Data Association*).
 - Komunikasi data yang menggunakan kabel *fiber optic*.

- *Free space optical communication*, yakni telekomunikasi antar 2 titik menggunakan sinar laser.
2. Gelombang radio merupakan bagian dari spektrum gelombang elektromagnetik. Implementasi gelombang radio untuk keperluan komunikasi antara lain :
 - Komunikasi radio, Komunikasi satelit, dan
 - *WiFi*.
 3. *Bluetooth* adalah protokol komunikasi *wireless* (tanpa kabel) yang dapat digunakan untuk pertukaran data dalam area yang terbatas. *Wireless network* yang dibentuk oleh *Bluetooth* masuk ke dalam kategori *WPAN*. *Bluetooth* menggunakan gelombang radio untuk mentransmisikan data dengan membagi data menjadi beberapa bagian dan mengirimkannya dalam beberapa kanal frekuensi yang berbeda. *Bluetooth* digunakan pada *peripheral PC*, seperti *mouse*, *keyboard*, dan perangkat *mobile*, seperti ponsel, dan *PDA*. *WiFi* adalah standar untuk *WLAN* yang dibuat oleh *WiFi Alliance* berdasarkan *IEEE 802.11*. Tujuan *WiFi* adalah memungkinkan komunikasi yang terjalin dengan baik pada berbagai perangkat *wireless* dari beragam pabrik pembuatnya. Jadi, buatan manapun perangkat *wireless* tersebut dapat saling berkomunikasi dengan baik selama mengikuti tata cara yang telah distandarkan oleh *WiFi*. *WiFi* diimplementasikan pada *PC*, Laptop, *UMPC*, *PDA*, dan berbagai perangkat *mobile* lainnya.

2.4.1 Komunikasi Paralel

Pada transmisi paralel, sejumlah *bit* dikirimkan per waktu. Masing-masing *bit* mempunyai jalurnya tersendiri. Dikarenakan oleh sifatnya yang demikian, maka data yang mengalir pada transmisi paralel jauh lebih cepat pada transmisi serial. Model transmisi paralel biasanya digunakan untuk melakukan komunikasi jarak pendek atau biasa disebut menggunakan kabel (*wire*). Contohnya, transmisi ke *printer* atau untuk komunikasi data dua buah komputer.

Pada transmisi paralel, beberapa *bit* (biasanya 8 *bit* atau satu *byte*/karakter) akan dikirim secara bersamaan pada saluran yang berbeda (kabel, saluran frekuensi) dalam kabel yang sama, atau radio jalan, dan disinkronisasi untuk sebuah jam. Perangkat paralel memiliki *bus* data yang lebih luas dari pada perangkat serial sehingga dapat mentransfer data dalam kata-kata dari satu atau lebih *byte* pada suatu waktu. Akibatnya, ada percepatan dalam transmisi paralel *bit rate* lebih dari laju bit transmisi serial. Namun, percepatan ini adalah biaya *versus tradeoff* sejak beberapa kabel biaya lebih dari satu kawat, dan sebagai kabel paralel mendapatkan lagi, sinkronisasi waktu antara beberapa saluran menjadi lebih sensitif terhadap jarak. Waktu untuk transmisi paralel disediakan oleh sinyal *clocking* konstan dikirim melalui kawat terpisah dalam kabel, sehingga transmisi paralel dianggap sinkron.

2.4.2 Komunikasi Serial

Komunikasi serial adalah komunikasi yang mengantarkan data digital secara *bit per bit* secara bergantian melalui media antarmuka serial, contoh: modem, *mouse* dan masih banyak lagi yang lainnya.

Pengiriman data melalui *interface* (antarmuka) serial dapat dilakukan secara *bit per bit* (setiap satu *step* waktu 1 *bit*) atau juga dalam satuan *baud* dimana 1 *baud* tidak mesti senilai dengan 1 *bit* persecond, tergantung besaran data untuk setiap kali *clock* transfer.

➤ Kerugian Komunikasi Paralel

- Penggunaan kabel yang lebih pendek, sebab keterbatasan proses transfer.
- Membutuhkan banyak kabel penghantar.

➤ Konsekuensi terhadap Komunikasi Serial

- Tingginya tingkat keamanan terhadap gangguan karena tingginya ayunan tegangan (dengan jangkauan max. 50 *volt*) sehingga dapat direalisasikan dengan kabel yang lebih panjang.
- Membutuhkan sedikit kabel penghantar (misalkan dengan tiga utas kabel: *Tx*, *Rx* dan *Ground*).

- Membutuhkan penyesuaian protokol komunikasi data terutama untuk sinkronisasi antara pengirim dan penerima.

➤ **Perbedaan diantara Komunikasi Serial**

- Perbedaan data *rate* (jumlah data per waktu)
- Jumlah dan jenis penghantar (min. 1 kabel *koaxial*)
- Penggunaan protokol komunikasi

➤ **Metode Sinkronisasi**

- Masalah utama komunikasi serial adalah metode sinkronisasi, yakni pengendalian *clock* pengirim dan penerima. Kedua *clock* seharusnya berada pada frekuensi yang sama, agar penerima dapat mengambil data tepat pada waktunya. Tujuan sinkronisasi adalah menghindari keterlambatan dan kesalahan pengambilan data sehingga perlu dilakukan penyesuaian *clock* penerima dengan *clock* pengirim.

➤ **Komunikasi Sinkron**

- Ditandai dengan *clock* penerima *disetting* hanya pada awal komunikasi *clock* pengirim.
- Terdapat dua bentuk realisasi:
 1. Menyediakan 3 penghantar (untuk data yang dikirim, data yang diterima dan *external clock*). Dengan bantuan

penghantar *clock*, penerima dapat mengendalikan proses pengambilan data (*sampling data*).

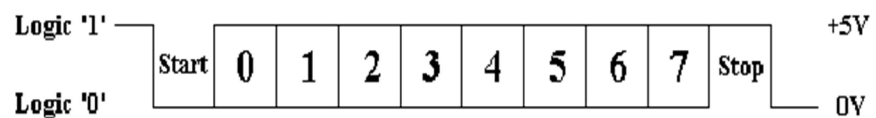
2. *Interface* serial terdiri hanya satu penghantar atau pasangan penghantar, dimana diawal paket data dikirimkan *bit preamble* sebagai *bit* sinkronisasi. *Clock* penerima akan mengalami *settingan* selama *bit preamble* berjalan.

➤ **Komunikasi Asinkron (Tidak Sinkron)**

- Ditandai dengan dimana sinkronisasi *clock* pengirim dan penerima terjadi pada awal dari setiap simbol data yang dikirim.
- Realisasinya: sebelum *bits* data terdapat satu atau dua *startbit*. *Starbit* ini menentukan kapan penerima mengambil data, dan ini berjalan dalam sebagian dari periode *clock*.
- Komunikasi *Asinkron* mengirimkan data secara simbol per simbol, dimana disini ditandai *acknowledge* untuk setiap penyelesaian masing-masing simbol.
- Format Data Komunikasi *Asinkron* tidak standar, bervariasi tergantung pada :

1. Genap atau ganjilnya **parity** (*parity* menandakan genap atau ganjilnya jumlah dari *bit* '1'). *Parity* disini yaitu pada *bit* ke delapan, sebagai pelengkap pengiriman.
2. Satu atau dua **stopbit**.

Komunikasi *Asynchrone* pada *RS-232* (mis. 1 simbol = 1 *Byte*)



Gambar 2.5. Bentuk gelombang komunikasi serial

Gambar diatas memperlihatkan bentuk gelombang komunikasi serial dengan format 8N1, yaitu 8-bit data, tanpa *parity*, 1 *stop bit*.

- Pada keadaan *idle* atau menganggur (*idle*), jalur *RS-232* ditandai dengan *mark state* atau Logika *HIGH*.
- Pengiriman data diawali dengan *start bit* yang berlogika 0 atau *LOW*, berikutnya data dikirimkan *bit* demi *bit* mulai dari *LSB* (*Least Significant bit*) atau *bit* ke-0.
- Pengiriman setiap *byte* diakhiri dengan *stop bit* yang berlogika *HIGH*.

Gambar di atas memperlihatkan kondisi *LOW* setelah *stop bit*, ini adalah *start bit* yang menandakan data berikutnya

akan dikirimkan. Jika tidak ada lagi data yang ingin dikirim, maka jalur transmisi ini akan dibiarkan dalam keadaan *HIGH*.

Ada yang disebut '*Break Signal*', yaitu keadaan *LOW* yang lamanya cukup untuk mengirimkan 8-bit data. Jika pengirim menyebabkan jalur komunikasi dalam keadaan seperti ini, penerima akan menganggap ini adalah '*break signal*' atau sinyal rusak. Data yang dikirimkan dengan cara seperti pada gambar diatas disebut data yang terbingkai (*to be framed*) oleh *start* dan *stop bit*. Jika *stop bit* dalam keadaan *LOW*, berarti telah terjadi *framing error*. Biasanya hal ini terjadi karena perbedaan kecepatan komunikasi antara pengirim dengan penerima.

2.4.3 Komunikasi Tanpa Kabel

Untuk transmisi *guided* atau dengan menggunakan kabel, pengiriman dan penerimaan signal dilakukan tanpa menggunakan antena. Untuk transmisi tanpa kabel antena memancarkan gelombang elektromagnetik ke medium transmisi udara dan kemudian diterima oleh antena penerima melalui medium yang ada disekitarnya (udara). Terdapat dua jenis konfigurasi dari *wireless transmission* yaitu satu arah (*directional*) dan segala arah (*omnidirectional*). Untuk transmisi satu arah, antena memfokuskan pancaran-pancaran elektromagnetik dimana antena pemancar dan antena penerima harus saling berhadapan dalam satu garis lurus.

Sedangkan untuk transmisi ke segala arah, signal yang dipancarkan menyebar ke segala arah dan dapat diterima oleh banyak antena dari berbagai arah. Pada umumnya, semakin tinggi frekuensi dari signal, semakin memungkinkan untuk memfokuskannya pada pemancaran ke segala arah.

Penggunaan media kabel untuk berkomunikasi dirasakan kurang efektif *fleksibel* dan *efisien*. Adapun beberapa media komunikasi *wireless* adalah sebagai berikut :

1) Infrared

Teknologi *infrared* merupakan teknologi pertama dan bersifat umum, seperti pada *remote* televisi. Prinsip kerja dari *remote* televisi berupa prosesor kecil pada *remote* yang akan melakukan penerjemahan pada saat penekanan tombol. Penekanan pada tombol tersebut merupakan instruksi bahasa mesin (*assembly*) yang dikirimkan melalui *infrared* ke TV, lalu data diubah kembali menjadi instruksi yang dikenali oleh TV tersebut.

Media transmisi yang sering terdapat dalam telepon selular yaitu *infrared*. Terutama bagi beberapa telepon selular seri lama yang belum memiliki media transmisi jenis baru.

IrDA(*Infrared Data Association*) memiliki panjang gelombang sekitar 875 Nm. Sinar yang dihasilkan dan dipancarkan dari sebuah lampu *LED*. Terdapat dua versi yang

telah dikembangkan pada *infrared*, yaitu versi 1.0 memiliki kecepatan dari 0,576 *Kbps* hingga 115,2 *Kbps* dan versi 2.0 memiliki kecepatan 0,576 *Mbps* hingga 1,152 *Mbps*.

2) *Wireless Fidelity (Wi-Fi)*

Wi-Fi (Wireless Fidelity) bekerja pada frekuensi sama dengan *Bluetooth*, yaitu pada 2.4 GHz. Tetapi terdapat perbedaan, *Bluetooth* menggunakan *Spread Spectrum Frequency Hopping (SSFH)*, sedangkan *Wi-Fi* menggunakan *Direct Sequence Spread Spectrum (DSSS)*. Jadi, *spread* pada *Wi-Fi* akan lebih stabil dan tentunya lebih cepat dibandingkan dengan *Bluetooth*. *Wi-Fi* memiliki kelemahan dalam hal keamanan dan rentan terhadap konflik dengan perangkat lain dalam waktu yang bersamaan. *Wi-Fi* yang dikenal dengan standar *IEEE 802.11b* pengoperasiannya mulai meluas dan beberapa operator di Amerika Serikat mengoperasikannya secara *Hot Spot* di lokasi strategis. Teknologi *Wi-Fi* pada telepon selular dewasa ini sudah mulai ramai di pasaran, baik itu telepon selular yang berjenis *GSM* maupun *CDMA*. [11]

3) *Bluetooth*

Bluetooth adalah sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz *unlicensed ISM* dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data

dan suara secara *real-time* antara *host-host Bluetooth* dengan jarak jangkauan layanan yang terbatas (sekitar 10 meter).[10]

Bluetooth bertujuan mengganti spesifik *IrDA* dari *InfraRed* pada ponsel dan peralatan *Mobile* lainnya.[3]

Karakteristik sistem *Bluetooth* adalah :

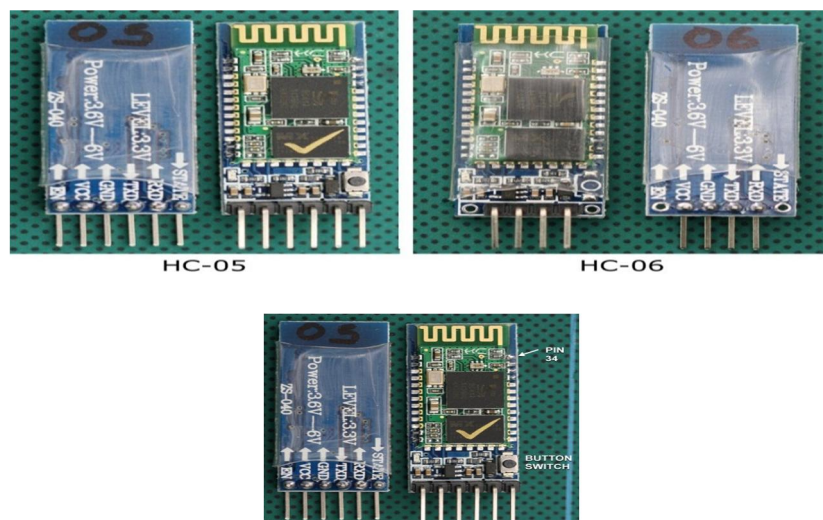
1. Beroperasi pada 2.4 Ghz *Industrial Scientific Medical (ISM) band*.
2. Memiliki range antara 10 m – 100 m.
3. Menggunakan *frequence Hop (FH) spread spectrum*, yang membagi frekuensi *band* ke beberapa *hop channels*.
4. Pada sebuah koneksi, radio *trancievers hop* dari satu *channel* ke *channel* lainnya.
5. Mendukung 8 peralatan dalam suatu *piconet* (lebih dari dua unit *Bluetooth* dalam *channel* secara bersamaan).
6. *Built in* dari segi keamanan.
7. *Omnidirectional*.
8. *Isochronous and asynchronous*, integrasi mudah dengan *TCP/IP networking*. [3]

ISM adalah *band (Industrial Scientific and Medical)* bebas lisensi yang ditetapkan oleh *Federal Comition Comunicaton (FCC)*. *FCC* menetapkan undang-undang yang mengatur pengoperasian piranti *LAN* nirkabel (tanpa kabel). *Band ISM* berada pada lokasi mulai dari 902 MHz, 2.4 GHz dan

5.8 GHz dengan lebar yang bervariasi dari sekitar 26 MHz hingga 150 MHz.[10]

Spesifikasi *Bluetooth* menyediakan definisi *link layer* dan *application layer* sehingga mendukung aplikasi data dan suara. Teknologi *Bluetooth* juga dapat menembus benda padat dan bersifat *omni directional* sehingga tidak memerlukan posisi *line of sight* seperti pada *infrared*. Keamanan merupakan prioritas utama dalam pengembangan spesifikasi *Bluetooth*. [10]

Bluetooth sendiri dapat berupa *card* yang bentuk dan fungsinya hampir sama dengan *card* yang digunakan untuk *wireless local area network (WLAN)* dimana menggunakan frekuensi radio standar *IEEE 802.11*, hanya saja pada *Bluetooth* mempunyai jangkauan jarak layanan yang lebih pendek dan kemampuan transfer data yang lebih rendah. Berikut gambaran secara fisik dari *Bluetooth* : [10]



Gambar 2.6. *Bluetooth* secara fisik

Pada dasarnya *bluetooth* diciptakan bukan hanya untuk menggantikan atau menghilangkan penggunaan kabel didalam melakukan pertukaran informasi, tetapi juga mampu menawarkan fitur yang baik untuk teknologi *mobile wireless* dengan biaya yang relatif rendah, konsumsi daya yang rendah, *interoperability* yang menjanjikan, mudah dalam pengoperasian dan mampu menyediakan layanan yang bermacam-macam.[10]

➤ **Keuntungan Teknologi *Bluetooth***

Teknologi *Bluetooth* memiliki fitur-fitur utama ketangguhan (*robustness*), kebutuhan daya yang kecil (*low power consumption*) dan biaya yang murah. Perangkat *Bluetooth* menghindari *interferensi* dengan sistem lainnya dengan cara mengirimkan signal yang sangat lemah, berkisar 1 *milliwatt*. Sebagai perbandingan, kekuatan telepon selular dapat memancarkan suatu *signal 3 watt*. [10]

Teknologi *Bluetooth* saat ini telah mengglobal dan menjadi standar untuk menghubungkan berbagai *device* elektronik, mulai dari *mobile phone*, *portable computer*, *mobil*, *handsfree headset*, sampai *MP3 player*. [3]

Bluetooth juga memiliki konsep unik "*profile*", sehingga memungkinkan produk untuk dapat bekerja tanpa menginstall *driver*, selain itu *Bluetooth* juga menyediakan layanan sekuritas *built-in*. Terdapat 3 mode sekuritas pada

spesifikasi *bluetooth*, yaitu :

1. *Mode Keamanan 1 : non-secure.*
2. *Mode Keamanan 2 : service level enforce security.*
3. *Mode Keamanan 3 : link level enforced security*

Device security levelnya terbagi menjadi 2, yaitu : *trusted device* dan *untrusted device*, sedangkan *service security* levelnya terbagi menjadi 3, yaitu :

1. Servis yang membutuhkan *authorization* dan *authentication*
2. Servis yang hanya membutuhkan *authentication*.
3. Servis yang terbuka untuk semua *device*. [10]

Selain layanan sekuritas yang memadai, teknologi *Bluetooth* relatif mudah digunakan karena *Bluetooth* tidak memerlukan kita untuk melakukan apapun yang khusus untuk membuatnya bekerja. [10]

Teknologi *bluetooth* juga sangat mendukung aplikasi *synchronous* dan *asynchronous*, sehingga memudahkan untuk mengimplementasikan *devices* (perangkat-perangkat) yang saling berbeda untuk berbagai macam layanan, misalnya seperti suara dan *internet*. Selain itu, karena *signalnya* dapat menembus benda padat dan bersifat *omni-directional*, maka tidak memerlukan posisi *line-of-sight*

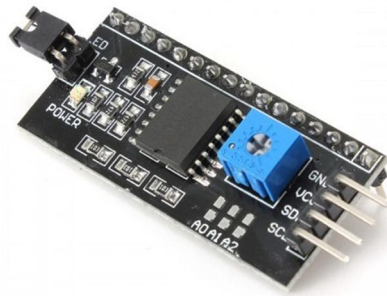
seperti pada *infrared*. [10]

Teknologi *Bluetooth* menyediakan layanan komunikasi *point to point* maupun komunikasi *point to multipoint*. Selain itu, sistem *Bluetooth* juga mendukung *bridging* jaringan, misalnya *mobile phone* yang menggunakan *chip bluetooth* dapat menggabungkan *piconet* lokal dengan jaringan *GSM* global. [10]

2.4.4 *I²C (Inter Integrated Circuit)*

Bus adalah sistem pengantar yang dilengkapi dengan komponen pengendali untuk melayani pertukaran data antara komponen *Hardware* satu dengan komponen *Hardware* lainnya. Pada sistem mikrokontroler terdapat *bus* data, *bus* alamat, dan beberapa pengantar pengendali. Semakin tinggi frekuensi *clock processor*, maka semakin lebih cermat pengembang untuk memperhatikan *timing* dari seluruh komponen yang terlibat, agar tidak terjadi kesalahan dalam transaksi data. *Bus* yang cukup sering digunakan adalah *bus* bersifat paralel. Transaksi data dilakukan secara paralel sehingga transaksi data lebih cepat, akan tetapi disisi lain mahal. Jika sistem relatif tidak membutuhkan transaksi yang cepat, maka penggunaan Serial *Bus* menjadi pilihan. Salah satu pilihan sistem data *bus* yang sering digunakan adalah *I²C (Inter Integrated Circuit)*. Sistem *Bus I²C* pertamakali diperkenalkan oleh *Firma Philips*. [2]

I^2C merupakan *bus* standar yang didesain oleh *Firma Philips* pada awal tahun 1980-an untuk memudahkan komunikasi antar komponen yang tersebar pada papan rangkaian. I^2C merupakan singkatan dari *Inter IC* atau komunikasi antar *IC*, sering disebut juga *IIC* atau I^2C . Pada awalnya, kecepatan komunikasi maksimumnya diset pada 100kbps karena pada awalnya kecepatan tinggi belum dibutuhkan pada transmisi data. Untuk yang membutuhkan kecepatan tinggi, ada *mode* 400kbps dan sejak 1998 ada *mode* kecepatan tinggi 3,4Mbps. I^2C tidak hanya digunakan pada komponen yang terletak pada satu *board*, tetapi juga digunakan untuk mengkoneksikan komponen yang dihubungkan melalui kabel.[2]



Gambar 2.7. Bentuk I^2C secara fisik

Karakter I^2C :

1. Serial *Bus*

Data dikirim serial secara per *bit*.

2. Menggunakan dua penghantar koneksi dengan *ground* bersama. I^2C terdiri dari dua penghantar :

- *SCL (Serial Clock Line)* untuk menghantarkan sinyal *clock*.
- *SDA (Serial Data)* untuk mentransaksikan data.

3. Jumlah Peserta *Bus* maksimal 127

Peserta dialamatkan melalui *7-bit* alamat. Alamat ditetapkan kebanyakan secara *hardware* dan hanya sebagian kecil dapat dirubah.[2]

4. Pengirim dan Penerima

Setiap transaksi data terjadi antara pengirim (*Transmitter*) dan penerima (*Receiver*). Pengirim dan penerima adalah peserta *bus*. [2]

5. *Master and Slave*

Device yang mengendalikan operasi transfer disebut *Master*, sementara *device* yang dikendalikan oleh *Master* disebut *Slave*. [2]

2.5 Sistem Informasi

Secara garis besar, sistem informasi merupakan gabungan teknologi informasi serta kegiatan orang yang memakai teknologi itu untuk mensupport operasi serta manajemen. Dalam makna yang amat luas, arti sistem informasi yang kerap dipakai mengacu pada hubungan antara

orang, sistem algoritmik, data serta teknologi. Dalam pengertian ini, istilah tersebut dipakai untuk mengacu bukan cuma pada pemakaian organisasi teknologi informasi dan komunikasi (TIK), namun juga untuk cara berinteraksi dengan teknologi ini dalam mendukung proses proyek.

➤ **Pengertian Sistem Informasi**

Sistem informasi adalah paduan yang terorganisasi dari manusia, piranti lunak, piranti keras, jaringan komunikasi serta sumber data dalam menghimpun, merubah serta menyebarkan informasi dalam organisasi.

2.5.1 *LCD (Liquid Crystal Display)*

LCD adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. Dipasaran tampilan *LCD* sudah tersedia dalam bentuk modul yaitu tampilan *LCD* beserta rangkaian pendukungnya termasuk *ROM* dan masih banyak lain lagi. *LCD* mempunyai *pin* data, kontrol *catu daya*, dan pengatur kontras tampilan.[8]

Sekarang *LCD* lebih dipilih dari pada *dot* matriks, *seven segment LED* atau *Multi segment LED* untuk tampilan, disebabkan oleh selain harganya murah, *LCD* sudah mampu menampilkan huruf, angka bahkan grafik sekalipun serta dalam memprogramnya lebih mudah.[8]

LCD juga merupakan perangkat *display* yang paling umum dipasangkan di Mikrokontroler, Mengingat ukurannya yang kecil

dan kemampuannya menampilkan karakter atau grafik yang lebih dibandingkan tampilan *seven segment*. Pada pengembangan sistem *embedded LCD* mutlak diperlukan sebagai sumber pemberi informasi utama, misalnya alat pengukur kadar gula darah, penampil jam, penampil *counter* putaran motor industri dan lainnya.[8]

Berdasarkan jenis tampilan, *LCD* dapat dikelompokkan menjadi beberapa jenis, yaitu :

- *Segment LCD*

LCD ini berbentuk dari beberapa *Sevent Segment display* atau *Sixteen Segment display*, namun ada juga yang menggabungkan keduanya. *LCD* ini sering dipakai untuk jam digital.[8]

- *Dot Matrix character LCD*

LCD ini terbentuk dari beberapa *Dot Matrix Display* berukuran 5x7 atau 5x9 yang membentuk sebuah matriks yang lebih besar dengan berbagai kombinasi jumlah baris dan kolom. Kombinasi ini yang menentukan karakter yang dapat ditampilkan *LCD* tersebut. Seperti 2 baris x 20 karakter atau 4 baris 20 karakter.[8]

- *Grafik LCD*

LCD jenis ini masih berkembang saat ini. Resolusi *LCD* ini bervariasi, diantaranya 128 x 64, 128 x 128. Sekarang ini

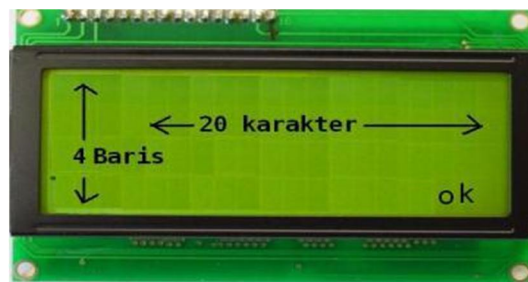
Graphic LCD banyak dipakai pada *Handycam*, laptop, telepon seluler (*cellphone*), monitor komputer dan lain sebagainya.[8]

Register yang terdapat di *LCD* adalah sebagai berikut :

- *IR (Intruction Register)* Digunakan untuk menentukan fungsi yang harus dikerjakan oleh *LCD* serta pengalamatan *DRAM* atau *CGRAM*. [8]
- *DR (Data Register)* Digunakan sebagai tempat data *DDRAM* atau *CGRAM* yang akan ditulis atau dibaca oleh komputer atau sistem minimum. Saat dibaca, *DR* menyimpan data *DDRAM* atau *CGRAM*, setelah itu data alamatnya secara otomatis masuk ke *DR*. Pada waktu menulis, cukup lakukan inisialisasi *DDRAM* atau *CGRAM*, kemudian untuk selanjutnya data dituliskan ke *DDRAM* atau *CGRAM* sejak awal alamat tersebut. [8]
- *BF (Busy Flag)* Digunakan untuk bahwa *LCD* dalam keadaan siap atau sibuk. Apabila *LCD* sedang melakukan operasi *internal*, *BF* diset menjadi 1, sehingga tidak akan menerima perintah dari luar. Jadi, *BF* harus dicek apakah telah diriset menjadi 0 ketika akan menulis *LCD* (memberi data pada *LCD*). Cara untuk menulis *LCD* adalah dengan mengeset *RS* menjadi 0 dan mengeset *R/W* menjadi 1. [8]
- *AC (Address Counter)* Digunakan untuk menunjukkan alamat pada *DDRAM* atau *CGRAM* dibaca atau ditulis, maka *AC* secara otomatis menunjukkan alamat berikutnya. Alamat yang disimpan

AC dapat dibaca bersamaan dengan *BF*. [8]

- *DDRAM (Display Data Random Access Memory)* Digunakan sebagai tempat penyimpanan data yang sebesar 80 *byte* atau 80 karakter. *AC* menunjukkan alamat karakter yang sedang ditampilkan. [8]
- *CGROM (Character Generator Read Only Memory)* Pada *LCD* terdapat *ROM* untuk menyimpan karakter-karakter *ASCII (American Standart Code for Interchage Intruction)*, sehingga cukup memasukan kode *ASCII* untuk menampilkanya. [8]
- *CGRAM (Character Generator Random Access Memory)* Sebagai data *storage* untuk merancang karakter yang dikehendaki. Untuk *CGRAM* terdapat kode *ASCII* dari 00h sampai 0Fh, tetapi hanya 8 karakter yang disediakan. Alamat *CGRAM* hanya 6 *bit*, 3 *bit* untuk mengatur tinggi karakter dan 3 *bit* tinggi menjadi 3 *bit* rendah *DDRAM* yang menunjukkan karakter, sedangkan 3 *bit* rendah sebagai posisi data *CGRAM* untuk membuat tampilan baris dalam dot matriks 5x7 karakter tersebut, dimulai dari atas. Sehingga karakter untuk kode *ASCII* 00h sama dengan 09h sampai 07h dengan 0Fh. Dengan demikian untuk perancangan 1 karakter memerlukan penulisan data ke *CGRAM* samapai 8 kali. [8]
- *Cursor and Blink Control circuit* Merupakan rangkaian yang menghasilkan tampilan kursor dan kondisi blink (berkedip). [8]



Gambar 2.5. Karakter LCD 4 x 20

Tabel 2.5. Deskripsi pin pada LCD

Pin	Simbol	I / O	Deskripsi
1	VSS	--	Ground
2	VCC	--	+ 5 V power suplay
3	VEE	--	Power suplay source to control contrast
4	RS	I	Register select: RS = 0 to select instruksi. Command register; RS = 1 to selsct data reg.
5	R / W	I	Read/Write: R/W = 0 for write, R/W = 1 for read
6	E	I	Enable
7	DB0	I/O	The 8-bit data bus
8	DB1	I/O	The 8-bit data bus
9	DB2	I/O	The 8-bit data bus
10	DB3	I/O	The 8-bit data bus
11	DB4	I/O	The 8-bit data bus
12	DB5	I/O	The 8-bit data bus
13	DB6	I/O	The 8-bit data bus
14	DB7	I/O	The 8-bit data bus

1.5.2 Perangkat Berbasis *Android*

Pada alat kali ini yaitu sistem *pre medical checkup* pengukuran tinggi badan berbasis ATmega 328 dengan penampil LCD dan sistem berbasis *Android*. Pada alat ini dimana *Bluetooth*

sebagai alat komunikasinya untuk memberikan/menampilkan informasi berupa data kepada pasien/dokter melalui *gadget* berbasis *Android*.