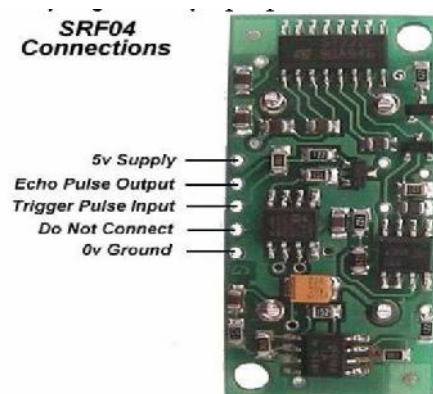


## BAB II

### TINJAUAN PUSTAKA

Pada bab ini diuraikan beberapa landasan teori yang dipergunakan untuk menyelesaikan masalah dan pengerjaan pada penelitian ini yang mencakup, prinsip kerja sensor ultrasonic, mikrokontroler ATmega16, komunikasi serial, aplikasi program *Borland Delphi* dan *CodeVision AVR*.

#### 2. 1. Sensor Ultrasonik



Gambar 2.1. Sensor Ultrasonik

( sumber : [www.parallax.com](http://www.parallax.com) )

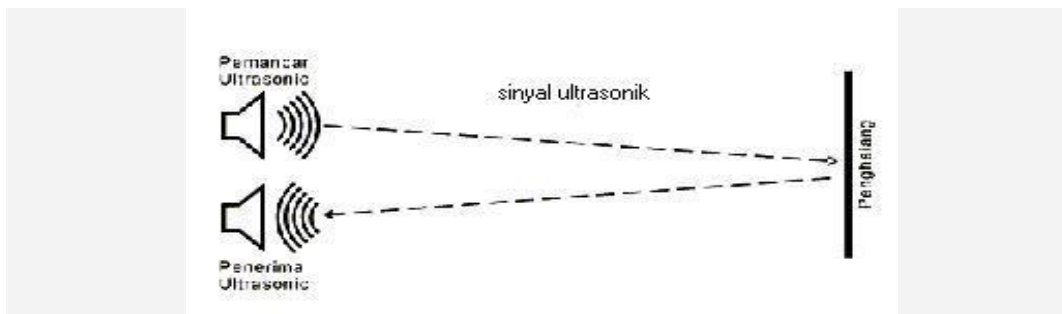
Berdasarkan gambar 2.1 Sensor ultrasonik adalah salah satu piranti elektronika yang dapat mengukur jarak didepannya dengan menggunakan sifat dari gelombang suara yang dipantulkan. Sensor (jarak) ultrasonik menggunakan media suara untuk melakukan pengukuran jarak. Sensor ultrasonik terdiri dari 2 bagian, yaitu bagian pemancar dan bagian penerima, bagian pemancar ultrasonik akan memancarkan gelombang ultrasonik pada frekwensi lebih dari 20kHz,

biasanya menggunakan frekwensi 40kHz. Kemudian sinyal tersebut akan dipantulkan kembali setelah mencapai suatu benda di depannya. Sinyal tersebut akan di terima kembali oleh bagian penerima sinyal. Jarak antara sensor dengan bidang pantul nya dapat diketahui dengan menggunakan rumus :  $S = V.t/2$

Dimana S adalah jarak sensor dengan bidang pantul, V adalah kecepatan suara (340m/s), dan t adalah selisih waktu antara pemancaran sinyal sampai diterima kembali oleh penerima ultrasonik. Sensor ultrasonik tersebut di hubungkan ke mikrokontroler untuk melakukan penghitungan waktu beserta jarak pantulnya.( [www.parallax.com](http://www.parallax.com) )

### **2.1.1 Prinsip Kerja Rangkaian 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 rangkaian pemancar ultrasonik yang disebut *transmitter* dan rangkaian 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 rangkaian *receiver* dikirimkan ke rangkaian mikrokontroler untuk selanjutnya diolah untuk menghitung jarak terhadap benda di depannya (bidang pantul).Prinsip kerja dari sensor ultrasonik dapat ditunjukkan dalam gambar dibawah ini :



Gambar 2.2. Prinsip kerja sensor ultrasonik

( sumber : [www.parallax.com](http://www.parallax.com) )

Dari gambar 2.2 Prinsip kerja dari sensor ultrasonik adalah sebagai berikut :

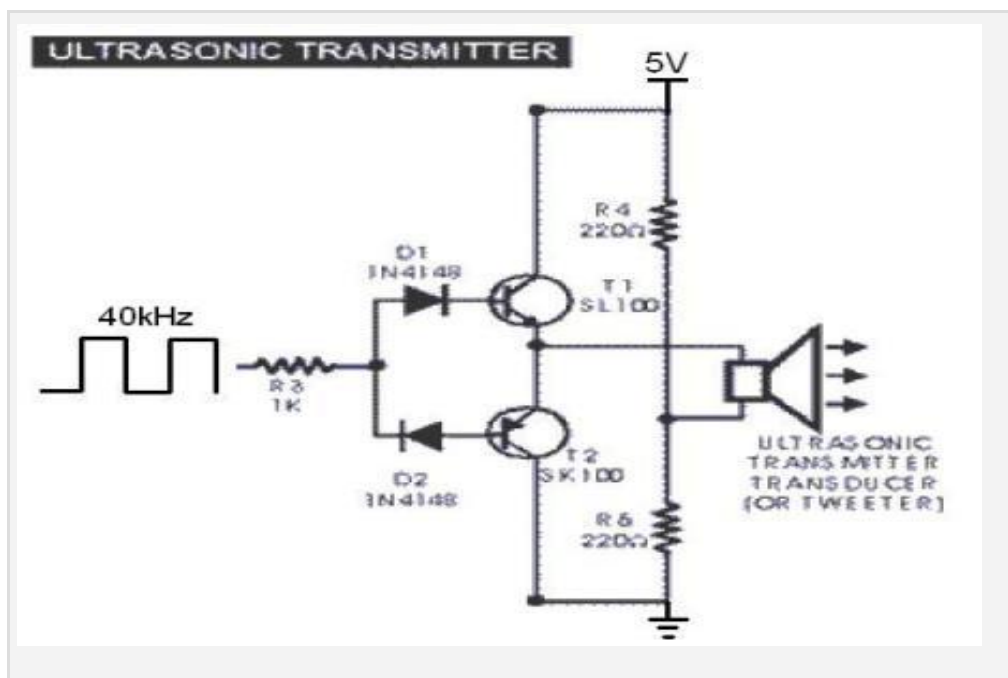
1. Sinyal dipancarkan oleh pemancar ultrasonik. Sinyal tersebut berfrekuensi diatas 20kHz, biasanya yang digunakan untuk mengukur jarak benda adalah 40kHz. Sinyal tersebut di bangkitkan oleh rangkaian pemancar ultrasonik.
2. Sinyal yang dipancarkan tersebut kemudian akan merambat sebagai sinyal / gelombang bunyi dengan kecepatan bunyi yang berkisar 340 m/s. Sinyal tersebut kemudian akan dipantulkan dan akan diterima kembali oleh bagian penerima Ultrasonik.
3. Setelah sinyal tersebut sampai di penerima ultrasonik, kemudian sinyal tersebut akan diproses untuk menghitung jaraknya. Jarak dihitung berdasarkan rumus :

$$S = 340.t/2$$

dimana S adalah jarak antara sensor ultrasonik dengan bidang pantul, dan t adalah selisih waktu antara pemancaran gelombang ultrasonik sampai diterima kembali oleh bagian penerima ultrasonik

### 2.1.2 Pemancar Ultrasonik (*Transmitter*)

Pemancar Ultrasonik ini berupa rangkaian yang memancarkan sinyal sinusoidal berfrekuensi di atas 20 KHz menggunakan sebuah *transducer transmitter* ultrasonik



Gambar 2.3. Rangkaian Pemancar Gelombang Ultrasonik  
( sumber : [www.parallax.com](http://www.parallax.com) )

Dari gambar 2.3 Prinsip kerja dari rangkaian pemancar gelombang ultrasonik tersebut adalah sebagai berikut :

1. Sinyal 40 kHz dibangkitkan melalui mikrokontroler.
2. Sinyal tersebut dilewatkan pada sebuah resistor sebesar 3kOhm untuk pengaman ketika sinyal tersebut membias maju rangkaian dioda dan transistor.

3. Kemudian sinyal tersebut dimasukkan ke rangkaian penguat arus yang merupakan kombinasi dari 2 buah dioda dan 2 buah transistor.
4. Ketika sinyal dari masukan berlogika tinggi (+5V) maka arus akan melewati dioda D1 (D1 on), kemudian arus tersebut akan membias transistor T1, sehingga arus yang akan mengalir pada kolektor T1 akan besar sesuai dari penguatan dari transistor.
5. Ketika sinyal dari masukan berlogika tinggi (0V) maka arus akan melewati dioda D2 (D2 on), kemudian arus tersebut akan membias transistor T2, sehingga arus yang akan mengalir pada kolektor T2 akan besar sesuai dari penguatan dari transistor.
6. Resistor R4 dan R6 berfungsi untuk membagi tegangan menjadi 2,5 V. Sehingga pemancar ultrasonik akan menerima tegangan bolak – balik dengan  $V_{\text{peak-peak}}$  adalah 5V (+2,5 V s.d -2,5 V).

### **2.1.3 Penerima Ultrasonik (*Receiver*)**

Penerima Ultrasonik ini akan menerima sinyal ultrasonik yang dipancarkan oleh pemancar ultrasonik dengan karakteristik frekuensi yang sesuai. Sinyal yang diterima tersebut akan melalui proses filterisasi frekuensi dengan menggunakan rangkaian *band pass filter* (penyaring pelewat pita), dengan nilai frekuensi yang dilewatkan telah ditentukan. Kemudian sinyal keluarannya akan dikuatkan dan dilewatkan ke rangkaian komparator (pembanding) dengan tegangan referensi

ditentukan berdasarkan tegangan keluaran penguat pada saat jarak antara sensor kendaraan mini dengan sekat/dinding pembatas mencapai jarak minimum untuk berbelok arah. Dapat dianggap keluaran komparator pada kondisi ini adalah *high* (logika '1') sedangkan jarak yang lebih jauh adalah *low* (logika '0'). Logika-logika biner ini kemudian diteruskan ke rangkaian pengendali mikrokontroler.

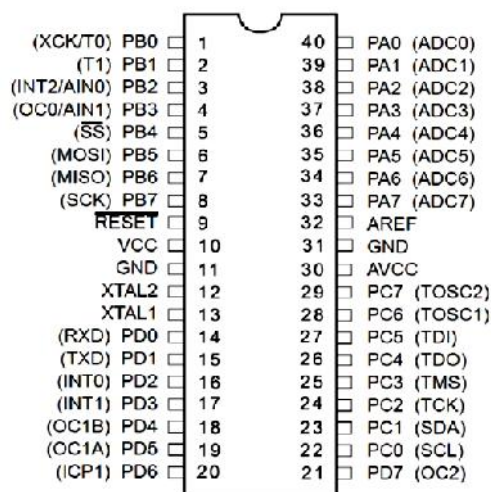
## 2.2 Mikrokontroler ATmega 16

Mikrokontroler ATmega16 merupakan sebuah *chip* mikrokomputer yang didalamnya terdapat mikroprosesor (CPU), *memory* RAM, ROM, EEPROM, I/O Port dan bahkan ADC serta beberapa fasilitas penunjang lainnya seperti misalnya Timer, USART, dan PWM yang sudah terintegrasi dalam satu IC.

Mikrokontroler ATmega16 ini memiliki arsitektur Harvard, yaitu memisahkan memori untuk kode program dan memori untuk data, sehingga dapat memaksimalkan unjuk kerja. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil (*pre-fetched*) dari memori program. Konsep inilah yang memungkinkan instruksi-instruksi dapat dieksekusi setiap satu *clock cycle*. CPU terdiri dari 32x8-bit *general purpose register* yang dapat diakses dengan cepat dalam satu *clock cycle*, yang mengakibatkan operasi *Arithmetic Logic Unit* (ALU) dapat dilakukan dalam satu *cycle*. Pada operasi ALU, dua *operand* berasal dari *register*, kemudian operasi dieksekusi dan hasilnya disimpan kembali pada *register* dalam satu *clock cycle*. Operasi aritmatik dan *logic* pada

ALU akan mengubah *bit-bit* yang terdapat pada *Status Register* (SREG). Proses pengambilan instruksi dan pengeksekusian instruksi berjalan secara *paralell*.

Konfigurasi *pin* ATmega16 dengan kemasan 40 *pin* DIP (*Dual In-line Package*) dapat dilihat pada gambar dibawah ini :



Gambar 2.4. Konfigurasi pin ATmega16

( sumber : [www.atmel.com](http://www.atmel.com) )

Dari gambar 2.4. dapat dijelaskan fungsi dari masing-masing *pin* ATmega16 sebagai berikut:

1. VCC merupakan *pin* yang berfungsi sebagai masukan catu daya.
2. GND merupakan *pin* *Ground*.
3. *Port* A(PA0..PA7) merupakan *pin* *input/output* dua arah dan *pin* masukan ADC.

4. Port B(PB0..PB7) merupakan *pin input/output* dua arah dan *pin* memiliki fungsi khusus (**lihat *datasheet ATMega16* [www.atmel.com](http://www.atmel.com)**).

### 2.2.1 Pemrograman Mikrokontroler ATMega 16

Pengembangan sebuah sistem menggunakan mikrokontroler AVR buatan ATMEL menggunakan *software* CodeVisionAVR. CodeVisionAVR merupakan *software C-cross compiler*, dimana program dapat ditulis dalam bahasa C, CodeVisionAVR memiliki IDE (*Integrated Development Environment*) yang lengkap, dimana penulisan program, *compile*, *link*, pembuatan kode mesin (*assembler*) dan *download* program ke *chip* AVR dapat dilakukan pada *CodeVision*. Selain itu terdapat fasilitas terminal untuk melakukan komunikasi serial dengan mikrokontroler. Proses *download* program ke IC mikrokontroler AVR dapat menggunakan *system download* secara ISP (*In-System Programming*). *In-System Programming Flash on-chip* mengizinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI.

Bentuk dasar program dalam bahasa C setidaknya harus memiliki sebuah fungsi. Fungsi dasar ini disebut dengan fungsi utama (fungsi *main*) dan memiliki kerangka program sebagai berikut:

```
void main (void)

{

// pernyataan-pernyataan }
```

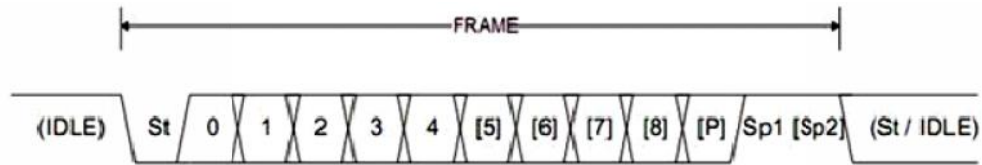


Jika terdapat beberapa fungsi yang lain maka fungsi inilah yang memiliki kedudukan paling tinggi dibandingkan fungsi-fungsi lain sehingga setiap kali program dijalankan akan selalu dimulai dari memanggil fungsi utama terlebih dahulu. Fungsi-fungsi yang lain dapat dipanggil setelah fungsi utama dijalankan melalui pernyataan-pernyataan yang berada di dalam fungsi utama.

### **2.2.2 USART (*Universal Synchronous and Asynchronous Receiver-Transmitter*)**

USART merupakan protokol komunikasi yang umum digunakan dalam pengiriman data serial antara *device* satu dengan yang lainnya. USART menjadi bagian perangkat keras komputer yang menerjemahkan antara bit-bit paralel data dan bit-bit serial. USART biasanya berupa sirkuit terintegrasi yang digunakan untuk komunikasi serial pada komputer atau port serial perangkat periperal. USART ini telah terdapat sebagai fitur di dalam mikrokontroler ATmega16.

USART memungkinkan transmisi data baik secara *synchronous* maupun *asynchronous* sehingga dengan demikian USART pasti kompatibel dengan UART. Pada ATmega16, untuk mengatur mode dan prosedur komunikasi baik *synchronous* atau *asynchronous* dilakukan melalui register UCSRA, UCSRB, UCSRC, UBRRH, UBRRL, dan UDR.



Gambar 2.5. Transmisi data USART

( sumber : [www.atmel.com](http://www.atmel.com) )

Gambar 2.5 di atas merupakan ilustrasi pengiriman data secara serial. Pada pengiriman data USART terdapat beberapa parameter yang dapat diatur yaitu start bit, parity bit, dan stop bit. Pengaturan ini harus sama antara pengirim dan penerima karena jika tidak maka data tidak akan diterima. Data yang dikirim adalah data berukuran 8 bit atau 1 byte. Jika ditambah dengan 3 parameter diatas maka total bit data yang dikirim adalah 11 bit. Dari format data inilah setiap data yang terbaca dapat diterjemahkan menjadi bit-bit yang merepresentasikan data tertentu.

Karena protokol USART ini universal, maka *baudrate* yang ada adalah nilai-nilai tetap yang tidak bisa diubah ubah dari kisaran nilai 110 sampai 11059200 bps (bit/second) atau lebih. Semakin cepat clock yang digunakan mikrokontroler maka *baudrate* akan semakin cepat juga. Umumnya *baudrate* yang digunakan dalam pengiriman data antar mikrokontroler adalah 9600 bps.

### 2.2.3 ADC (*Analog to Digital Converter*)

ADC adalah pengubah *input* analog menjadi kode-kode digital. ADC banyak digunakan sebagai Pengatur proses industri, komunikasi digital dan rangkaian pengukuran/pengujian. Umumnya ADC digunakan sebagai perantara antara sensor yang kebanyakan analog dengan sistim komputer seperti sensor suhu, cahaya, tekanan/ berat, aliran dan sebagainya kemudian diukur dengan menggunakan sistim digital (komputer). ADC memiliki 2 karakter prinsip, yaitu kecepatan sampling dan resolusi . Kecepatan sampling suatu ADC menyatakan seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu. Kecepatan sampling biasanya dinyatakan dalam *sample per second* (SPS).

Resolusi ADC menentukan ketelitian nilai hasil konversi ADC. Sebagai contoh: ADC 8 bit akan memiliki *output* 8 bit data digital, ini berarti sinyal *input* dapat dinyatakan dalam 255 ( $2^n-1$ ) nilai diskrit. ADC 12 bit memiliki 12 bit *output* data digital, ini berarti sinyal *input* dapat dinyatakan dalam 4096 nilai diskrit. Dari contoh diatas ADC 12 bit akan memberikan ketelitian nilai hasil konversi yang jauh lebih baik daripada ADC 8 bit.

Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal *input* dan tegangan referensi. Sebagai contoh, bila tegangan referensi 5 volt, tegangan *input* 3 volt, rasio *input* terhadap referensi adalah 60%. Jadi, jika menggunakan ADC 8 bit dengan skala

maksimum 255, akan didapatkan sinyal digital sebesar  $60\% \times 255 = 153$  (bentuk decimal) atau 10011001 (bentuk biner). Contoh perhitungannya sebagai berikut.

$$\begin{aligned}\text{signal} &= (\text{sample}/\text{max\_value}) * \text{reference\_voltage} \\ &= (153/255) * 5 \\ &= 3 \text{ Volts}\end{aligned}$$

Mikrokontroler ATmega16 memiliki fasilitas Analog to Digital Converter yang sudah *built-in* dalam *chip*. ATmega16 memiliki resolusi ADC 10-bit dengan 8 *channel input* dan mendukung 16 macam penguat beda. ADC ini bekerja dengan teknik *successive approximation*. Rangkaian internal ADC ini memiliki catu daya tersendiri yaitu pin AVCC. Tegangan AVCC harus sama dengan VCC  $\pm$  0,3 V.

Data hasil konversi ADC dirumuskan sebagai berikut:

$$ADC = \frac{V_{in} \cdot 1024}{V_{REF}}$$

Dimana:

$V_{in}$  : Tegangan masukan pada pin yang dipilih.

$V_{ref}$  : Tegangan referensi yang dipilih.

Untuk mengatur mode dan cara kerja ADC pada ATmega16 dilakukan melalui register ADMUX, ADCSRA, ADCL, ADCH, dan SFIOR.

## 2.3 Komunikasi Serial

Komunikasi Serial adalah pengiriman data satu persatu secara berurutan dalam waktu tertentu. Oleh karena itu komunikasi serial lebih lambat daripada komunikasi paralel. Dalam komunikasi serial dikenal 2 cara pengiriman yaitu secara sinkron dan asinkron. Pada transmisi data secara asinkron, sinyal *clock* diperlukan oleh penerima data untuk mengetahui adanya pengiriman setiap *bit* data. Data akan dikirim dan diterima dengan kecepatan yang sama. Sedangkan pada transmisi data secara sinkron tidak memerlukan sinyal *clock* sebagai sinkronisasi, namun pengiriman data harus diawali dengan *start bit* dan diakhiri dengan *stop bit*. Jadi data bisa dikirimkan kapan saja. Penerima hanya perlu mendeteksi adanya *start bit* sebagai awal pengiriman data, dan menunggu adanya *stop bit* sebagai tanda bahwa data telah dikirim (Yatna Supriyatna dkk, 2009).

Komunikasi serial lebih sulit ditangani karena peralatan yang dihubungkan ke serial *port* harus berkomunikasi dengan transmisi serial, sedangkan data yang diterima komputer harus diproses secara paralel. Oleh karena itu data yang diproses dari dan ke serial *port* harus diubah terlebih dahulu ke dan dari *paralel port* agar bisa diolah oleh komputer. Dengan menggunakan *hardware*, hal ini bisa dilakukan dengan *Universal Asynchronous Receiver Transmitter (UART)*.

Serial *port* menggunakan logika '1' atau *mark* dengan kisaran tegangan -3 Volt hingga -25 Volt dan menggunakan logika '0' atau *space* dengan kisaran tegangan +3 Volt hingga +25 Volt. Sehingga, permasalahan turunnya tegangan karena panjangnya kabel atau disebut dengan *cable loss* bukan menjadi masalah. Sinyal komunikasi serial memiliki beberapa elemen penting diantaranya :

### **1. Kecepatan Data (*Baud Rate*)**

Kecepatan Data atau dikenal dengan istilah *baud rate* merupakan kecepatan transmisi data per bit yang dinyatakan dalam satuan Bps (Bit perSecond). Komunikasi serial antara satu modul perangkat elektronik dengan modul lain dalam penetapan *baud rate* antara keduanya harus sama, sehingga keduanya bisa berkomunikasi.

### **2. Jumlah bit data per karakter (*data length*)**

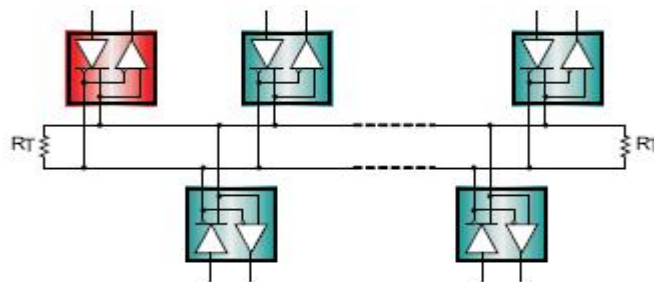
Dalam komunikasi data serial mode *asynchronous* biasanya berlangsung transmisi data yang dikemas dalam bentuk karakter. Dalam satu karakter diperbolehkan terdiri dari beberapa variasi jumlah bit. Dari sekian variasi yang diperbolehkan diantaranya adalah terdiri dari 7 bit dan 8 bit (panjang data karakternya saja). Kedua variasi ini adalah yang paling sering digunakan dalam komunikasi data serial.

### **3. Bit Parity**

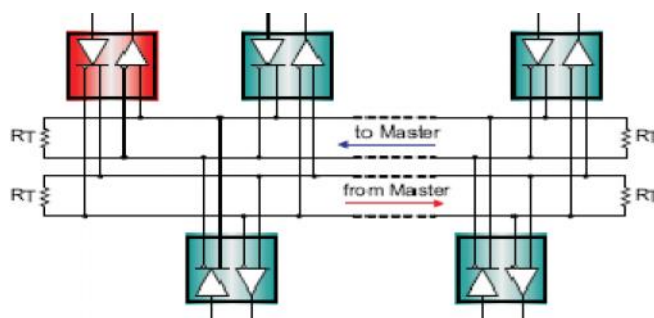
Bit *parity* adalah bit yang digunakan sebagai alat pemeriksaan kesalahan sederhana dalam proses transmisi data *digital*. Bit *parity* ini akan diletakkan setelah susunan bit data. Bit *parity* ini ada tiga macam, yaitu : *parity* ganjil, *parity* genap dan tanpa *parity* (tidak diikuti dalam pemeriksaan kesalahan).

### 2.3.1 Komunikasi Serial RS-232

RS 232 merupakan salah satu perangkat yang berfungsi sebagai antarmuka (*interfacing*) komputer ke peralatan luar dengan pengiriman data secara serial (per bit). RS-232 didesain untuk komunikasi dua peralatan elektronik dengan jarak maksimal 100 feet (Axelson, 1998). Komunikasi serial RS-232 menggunakan logika '1' atau *mark* dengan kisaran tegangan -3 Volt hingga -25 Volt dan menggunakan logika '0' atau *space* dengan kisaran tegangan +3 Volt hingga +25 Volt.



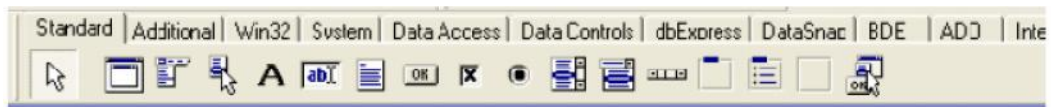
Gambar 2.6. Komunikasi 2 *wire*



Gambar 2.7. Komunikasi 4 *wire*

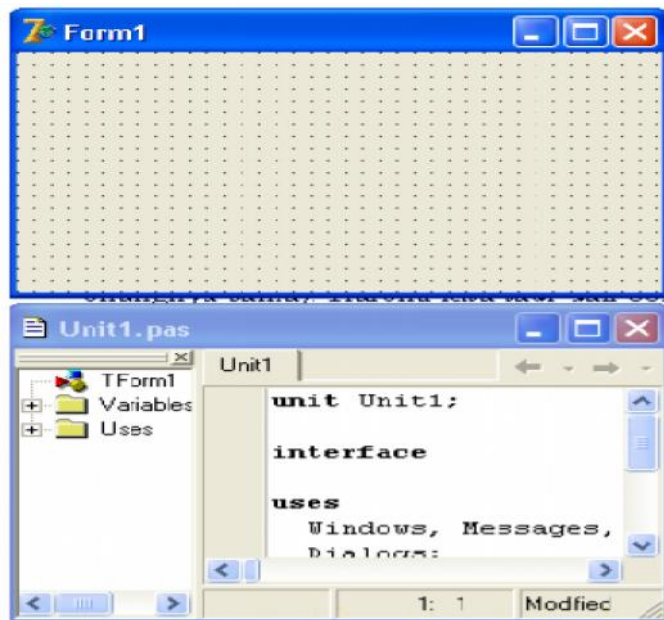
## 2.4 Pemrograman Borland Delphi

Dalam membuat program, Borland Delphi telah menyediakan banyak kemudahan, yaitu dengan disediakannya komponen-komponen. Komponen ini merupakan sebuah prosedur atau program yang sudah dikompilasi dan langsung dapat digunakan, sesuai dengan fungsinya masing-masing. Berikut merupakan tampilan komponen-komponen yang dapat digunakan dalam Borland Delphi.



Gambar 2.8. Komponen *Borland Delphi*

Komponen-komponen tersebut dapat digunakan dengan cara mengklik lalu diklik pada *form* sehingga komponen akan muncul pada *form*.



Gambar 2.9. *form*



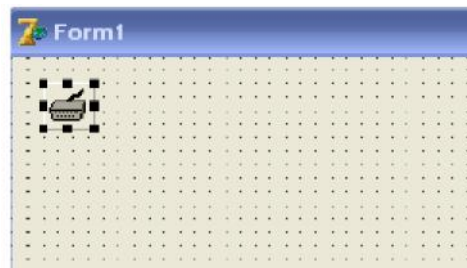
### 2.4.1 Komponen TComport

Delphi 7 tidak mempunyai *package* serial sehingga perlu diinstal terlebih dahulu. TComport merupakan komponen *freeware* untuk pemrograman serial karya Dejan Crnila. Komponen ini bisa diunduh secara gratis melalui situs internet. Setelah menginstal serial *package* maka akan muncul CportLib *tab* dengan komponen seperti dibawah ini:



Gambar 2.10. Komponen *Tcomport*

Untuk memulai membuat program, komponen maka komponen diseret dan diletakkan pada form, seperti pada gambar:



Gambar 2.11. *Form* dengan komponen *Tcomport*

langkah selanjutnya, dibuat program untuk memanggil *dialog setting* dari parameter serial yang akan digunakan. Programnya adalah cukup dengan menuliskan *syntax* yang berada di kotak merah program dibawah ini:

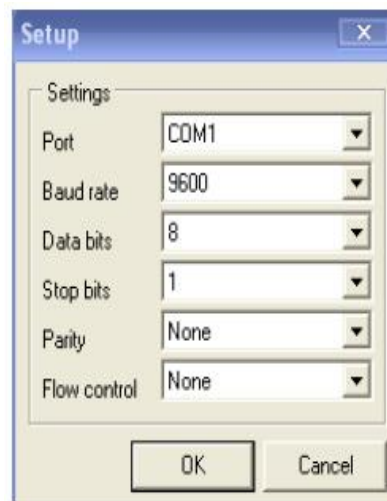
```

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  comport1.ShowSetupDialog;
end;

```

Gambar 2.12. Penulisan *syntax*

Jika program diatas dieksekusi, maka akan tampil *dialog box setting serial* seperti pada gambar dibawah ini:



Gambar 2.13. Dialog *Box Setting Serial*

#### 2.4.2. Program Mengirim Data

Untuk membuat program mengirim data maka perlu dipersiapkan sebuah variabel biasa bertipe *string* atau *integer*. Untuk memulai proses pengiriman data maka *com* serial harus dibuka atau dikoneksikan terlebih dahulu dengan delphinya.

Adapun program transmit data seperti berikut:

```
Comport1.open;
```

```
Comport1.write(str);
```

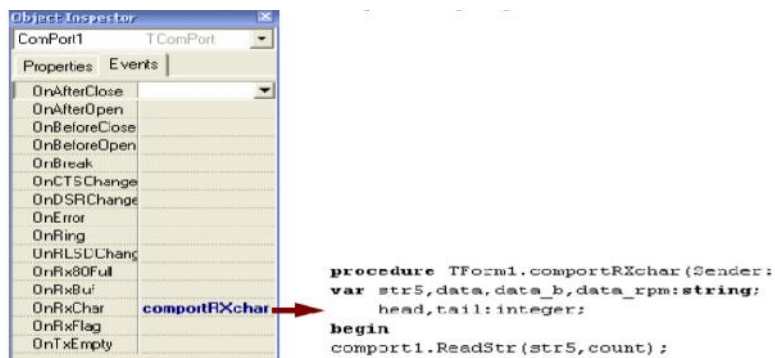
Maksud program:

comport1.Open → *open* koneksi delphi dengan *com* serial komputer

comport1.WriteStr(str) → transfer data *string* (data yang ditransfer bertipe *string*).

### 2.4.3. Program Menerima Data

Untuk membuat program menerima data seperti berikut:



Gambar 2.14. *Syntax* Menerima Data

maksud Program:

comport1.ReadStr(str5,count) → menerima data dari luar dan dipindahkan ke variabel str5 yang bertipe *string*.

## 2.5 CodeVision AVR

*CodevisionAVR* merupakan *compiler* bahasa C, dengan lingkungan pemrograman terpadu dan program generator otomatis didesain khusus untuk mikrokontroler keluarga Atmel AVR. Dapat digunakan pada Windows 98, Me, NT 4, XP, dan Vista. Fungsi utama CodevisionAVR ialah untuk menuliskan program yang akan dimasukkan kedalam mikrokontroller (Haiduc, 2008)

*Integrated Development Environment (IDE)* sudah dibangun dalam AVR *Chip In-System Programmer software* yang memungkinkan untuk secara otomatis mengirim program ke mikrokontroler setelah sukses di compile. *In-System Programmer software* didesain untuk dapat bekerja dengan Atmel STK500, AVRISP, AVRISP MkII, AVR Dragon, AVRProg (AVR910 *application note*), Kanda Systems STK200+, STK300, Dontronics DT006, Vogel Elektronik VTEC-ISP, Futurlec JRAVR and MicroTronics' ATCPU, Mega2000 *development boards*. Selain *library* standar C, CodeVisionAVR C compiler juga memiliki *library* lainnya yaitu

1. *Alphanumeric LCD modules*
2. *Philips I2C bus*
3. *National Semiconductor LM75 Temperature Sensor*
4. *Philips PCF8563, PCF8583, Maxim/Dallas Semiconductor DS1302 and DS1307 Real Time Clocks*
5. *Maxim/Dallas Semiconductor 1 Wire protocol*

6. Maxim/Dallas Semiconductor DS1820, DS18S20 and DS18B20  
*Temperature Sensors*

7. Maxim/Dallas Semiconductor DS1621 Thermometer/Thermostat

8. Maxim/Dallas Semiconductor DS2430 and DS2433 EEPROMs

9. SPI

10. Power management

11. Delays

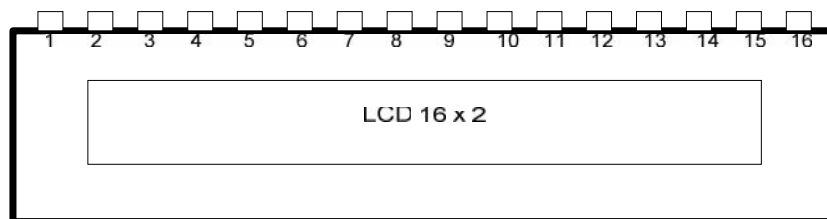
12. Gray code converse

( sumber : [www.atmel.com](http://www.atmel.com) )

## 2.6. LCD (Liquid Crystal Display) M1632 16x2 Char.

Pada LCD, adalah sebuah peraga kristal cair. Prinsip kerja LCD adalah mengatur cahaya yang ada, atau nyala LED. Dibandingkan dengan seven segment, memang LCD lebih dianggap rumit oleh sebagian orang, akan tetapi ada pula orang yang lebih suka memakai LCD karena pemakaian daya yang sangat rendah, selain itu juga karena jumlah karakter yang ditampilkan semakin banyak.

LCD yang akan penulis gunakan dalam pembuatan Tugas Akhir adalah LCD M1632 buatan *Seiko Instrument Inc.* Meskipun harganya termasuk dalam kategori mahal akan tetapi lebih praktis dalam segi pembuatan modul beserta pembuatan programnya, selain itu juga karena mudah didapat. Berikut ini gambar 2.15 diperlihatkan konfigurasi penyemat LCD.



Gambar 2.15. Konfigurasi penyemat LCD

(Sumber : *Seiko Instrument Inc., Japan, 1987*)

Pada gambar 2.15 diperlihatkan konfigurasi penyemat LCD yang terdiri dari 16 penyemat yang masing-masing penyemat mempunyai fungsi yang berbeda-beda. Berikut tabel 1. diperlihatkan fungsi-fungsi tersebut.

Tabel 1. Fungsi penyemat LCD M1632

Penyemat	Simbol	Logika	Keterangan
1	Vss	-	Catu Daya 0 Volt (Ground)
2	Vcc	-	Catu Daya 5 Volt
3	Vee	-	Catu daya untuk LCD
4	RS	H/L	H: Masukan Data, L: Masukan Instruksi
5	R/W	H/L	H: Baca (Read), L: Tulis (Write)
6	E	H/L (L)	Enable Signal
7	DB0	H/L	Data Bit 0
8	DB1	H/L	Data Bit 1
9	DB2	H/L	Data Bit 2
10	DB3	H/L	Data Bit 3
11	DB4	H/L	Data Bit 4
12	DB5	H/L	Data Bit 5
13	DB6	H/L	Data Bit 6
14	DB7	H/L	Data Bit 7
15	V+ BL	-	Backlight 4-4,2 Volt ; 50-200 mA
16	V- BL	-	Backlight 0 Volt (ground)

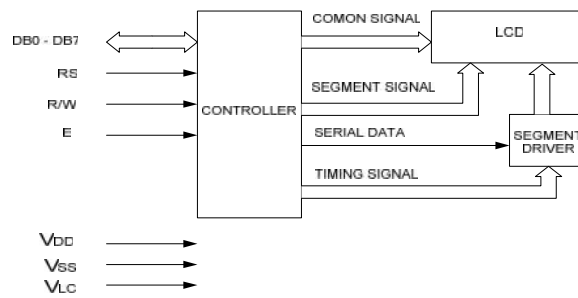
(Sumber : Seiko Instrument Inc., Japan, 1987)

LCD M1632 mempunyai karakteristik sebagai berikut :

1. 16 karakter, dua baris tampilan kristal cair (LCD) dari matriks titik.
2. Duty Ratio : 1/16. ROM pembangkit karakter untuk 192 tipe karakter (bentuk karakter 5 x 7 matriks titik).
3. Mempunyai dua jenis RAM yaitu, RAM pembangkit karakter dan RAM data tampilan.
4. RAM pembangkit karakter untuk 8 tipe karakter program tulis dengan bentuk 5 x 7 matrik titik.
5. RAM data tampilan dengan bentuk 80 x 8 matrik titik (maksimum 80 karakter).
6. Mempunyai pembangkit clock internal.
7. Sumber tegangan tunggal +5 Volt.
8. Rangkaian otomatis reset saat daya dinyalakan.
9. Jangkauan suhu pengoperasian 0 sampai 50 derajat.

LCD M1632 terdiri dari dua bagian utama. Bagian pertama merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf / angka dua baris, masing-masing baris bisa menampung 16 huruf/angka. Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroler yang ditempelkan dibalik panel LCD, yang berfungsi mengatur tampilan informasi serta mengatur komunikasi LCD M1632 dengan mikrokontroler. Gambar 2-16 diperlihatkan diagram blok pengendali LCD.





Gambar 2.16. Diagram blok pengendali LCD

(Sumber : Seiko Instrument Inc., Japan, 1987)

Dari gambar 2.16 diatas dijelaskan bahwa data inputan pada LCD yang berupa 8 bit data (D0-D7) diterima terlebih dahulu di dalam mikrokontroler dalam LCD yang berguna untuk mengatur data inputan sebelum ditampilkan dalam LCD. Selain itu juga dilengkapi dengan inputan E, R/W, dan RS yang digunakan sebagai pengendali mikrokontroler. Pada proses pengiriman data R/W=1 dan proses pengambilan data R/W=0.

Penyemat RS dipakai untuk membedakan jenis data yang dikirim, jika RS=0 data yang dikirim adalah perintah untuk mengatur kerja modul LCD, sedangkan jika RS=1 data yang dikirim adalah kode ASCII yang ditampilkan. Demikian pula saat pengambilan data, jika RS=0 data yang diambil dari modul merupakan data status yang mewakili aktivitas modul LCD, sedangkan saat RS=1 maka data yang diambil merupakan kode ASCII dari data yang ditampilkan.