BAB II
DASAR TEORI

2.1. Angin

Angin adalah udara yang bergerak dari satu tempat ke tempat lainnya. Angin akan berhembus dikarenakan ada beberapa bagian bumi yang mendapat lebih banyak panas matahari dibandingkan tempat yang lain. Permukaan tanah yang panas membuat suhu udara di atasnya naik. Akibatnya udara mengembang dan menjadi lebih ringan. Karena lebih ringan dibanding udara disekitarnya, udara akan naik. Begitu udara panas tadi naik, tempatnya segera digantikan oleh udara disekitarnya, terutama udara dari atas yang lebih dingin dan berat. Proses ini terjadi terus menerus. Akibatnya kita bisa merasakan adanya pergerakan udara atau yang kita sebut angin. Angin bertiup dari tempat yang bertekanan tinggi menuju ke tempat yang bertekanan rendah. Semakin besar perbedaan tekanan udaranya, semakin besar pula angin yang bertiup.

2.2. Sifat Angin

Ada tiga hal penting yang berkaitan dengan sifat angin yaitu:

1. Kekuatan Angin

Kekuatan angin bertiup adalah berbanding lurus dengan *gradient-barometer*nya. Jadi makin besar *gradient-barometer*nya, makin kuat angin bertiup. *Gradient-barometer* adalah angka yang menunjukkan perbedaan tekanan udara antara dua *isobar* pada tiap jarak lurus 111 km (15 meredian).

2. Arah Angin

Menurut hukum Buys Ballot, udara bergerak dari daerah yang bertekanan tinggi (maksimum) ke daerah bertekanan rendah (minimum).

Arah angin dipengaruhi oleh tiga faktor:

i Gradient barometrik

Makin besar gradient barometrik, makin besar pula kekuatannya. Angin yang besar kekuatannya makin sulit berbelok arah

ii Rotasi bumi

Rotasi bumi, dengan bentuk bumi yang bulat, menyebabkan pembelokan arah angin. Pembelokan angin di ekuator sama dengan 0 (nol). Makin ke arah kutub pembelokannya makin besar. Pembelokan angin yang mencapai 90° sehingga sejajar dengan garis isobar disebut angin geotropik. Hal ini banyak terjadi di atas samudra yang beriklim sedang.

iii Kekuatan yang menahan (rintangan)

Kekuatan yang menahan dapat membelokan arah angin. Sebagai contoh, pada saat melalui gunung, angin akan berbelok ke arah kiri, ke kanan atau ke atas.

3. Kecepatan Angin

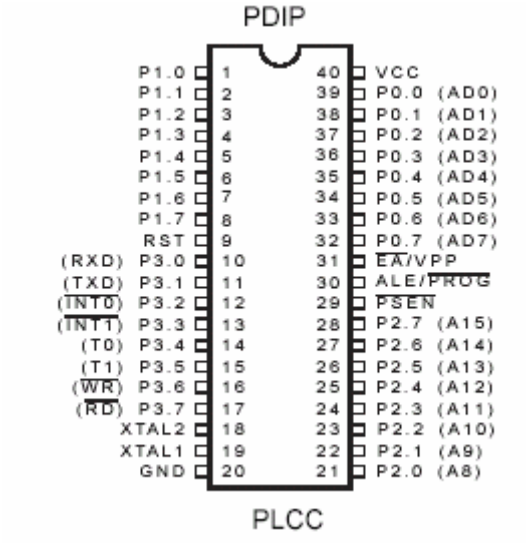
Atmosfer ikut berotasi dengan bumi. Molekul-molekul udara mempunyai kecepatan gerak ke arah timur, sesuai dengan arah rotasi bumi. Kecepatan gerak tersebut disebut kecepatan linier. Bentuk bumi yang bulat ini menyebabkan kecepatan linier makin kecil jika makin dekat ke arah kutub.

2.3. Mikrokontroler AT89C51

Mikrokontroler AT89C51 adalah mikrokontroler buatan Atmel yang termasuk keluarga mikrokontroler MCS-51. AT89C51 mempunyai 40 kaki, 32 kaki diantaranya adalah untuk keperluan port paralel. Satu port paralel terdiri dari 8 kaki, dengan demikian 32 kaki tersebut membentuk 4 buah port paralel, yang masing-masing dikenal sebagai port 0, port 1, port 2 dan port 3. Nomor dari masing-masing jalur (kaki) dari port paralel mulai dari 0 sampai 7. Jalur pertama Port0 disebut sebagai P0.0 dan jalur terakhir untuk Port3 adalah P3.7.

2.3.1. Deskripsi Pin AT89C51

Susunan pin-pin mikrokontroler AT89C51 memperlihatkan pada gambar 2.1 dibawah ini. Penjelasan dari masing-masing pin adalah sebagai berikut :



Gambar 2.1 Diagram Pin Mikrokontroler AT89C51

Tabel 2.1. Fungsi-fungsi pin

Nama Pin	Fungsi
Vcc	Suplai tegangan
GND	Ground atau pentanahan
RST	Masukan reset. Kondisi ‘1’ selama 2 siklus mesin selama osilator bekerja akan mereset mikrokontroler yang bersangkutan
ALE/PROG	ALE (*Adress Latch Enable*) menghasilkan pulsa-pulsa untuk mengancing *byte* rendah alamat selama mengakses memori eksternal. Kaki ini juga berfungsi sebagai masukan pulsa program atau PROG selama pemrograman flash. Pada operasi normal, ALE akan berpulsa dengan laju 1/6 dari frekuensi kristal dan dapat digunakan sebagai pewaktuan (*timing*) atau pendetakan (*clocking*) rangkaian eksternal
PSEN	*Program Store Enable* merupakan sinyal baca untuk memori program eksternal. Saat mikrokontroler menjalankan program dari memori eksternal, PSEN akan diaktifkan dua kali per siklus mesin, kecuali dua aktivitas PSEN dilompati (diabaikan) saat mengakses memori data eksternal
EA/Vpp	*External Access Enable* harus selalu dihubungkan ke *ground*, jika mikrokontroler akan mengeksekusi program dari memory eksternal lokasi 0000h hingga FFFFh. Selain dari itu, EA harus dihubungkan ke Vcc agar mikrokontroler mengakses program secara internal. Kaki ini juga berfungsi menerima tegangan 5 volt (Vpp) selama pemrograman *flash*

2.3.2. Ringkasan Spesifikasi Masing-masing Port

Port0

Port0 merupakan port keluaran/masukan (I/O) bertipe *open drain bidirectional*. sebagai port keluaran, masing-masing kaki dapat menyerap arus delapan masukan TTL (sekitar 3,8 mA). pada saat ‘1’ dituliskan ke kaki-kaki port0 ini, maka kaki-kaki port0 dapat digunakan sebagai masukan-masukan berimpedansi tinggi

Port0 juga dapat dikonfigurasikan sebagai bus alamat/data bagian rendah (*low byte*) selama proses pengakesan memory data dan program eksternal. Jika digunakan dalam mode ini port0 memiliki *pullup* internal

Port0 juga menerima kode-kode yang dikirimkan kepadanya selama proses pemrograman dan mengeluarkan kode-kode selama proses *verifikasi* program yang telah tersimpan dalam *flash*. Dalam hal ini dibutuhkan *pullup* eksternal selama proses *verifikasi* program.

Port1

Port1 merupakan port I/O yang dilengkapi dengan *pullup* internal. Penyangga keluaran port1 mampu memberikan/menyerap arus empat masukan TTL (sekitar 1,6 mA)

Jika ‘1’ dituliskan ke kaki-kaki port1, maka masing-masing kaki akan di *pullup high* dengan *pullup* internal sehingga dapat digunakan sebagai masukan. Sebagai masukan, jika kaki-kaki port ‘1’ dihubungkan ke *ground* (di *pullup low*), maka masing-masing kaki akan memeberikan arus (*source*) karena di *pullup high* secara internal. Port 1 juga menerima alamat bagain rendah (*low byte*) selama pemrograman dan verifikasi flash

Port2

Port2 merupakan port I/O dwi arah dengan dilengkapi *pullup* internal. Penyangga keluaran port2 mampu memberikan/menyerap arus empat masukan TTl (sekitar 1,6 mA)

Jika ‘1’ dituliskan ke kaki-kaki port2, maka masing-masing kaki akan di *pullup high* dengan pullup internal sehingga dapat digunakan sebagai masukan.

Sebagai masukan, jika kaki-kaki port2 dihubungkan ke *ground*, maka masing-masing kaki akan memberikan arus karena di *pullup high* secara internal

Port2 akan memberikan byte alamat bagian tinggi (*high byte*) selama pengambilan instruksi dari memori program eksternal dan selama pengaksesan memori data eksternal yang menggunakan perintah dengan alamat 16 bit. Dalam aplikasi ini, jika ingin mengirimkan ‘1’, maka digunakan pullup internal yang sudah disediakan. Selama pengaksesan memori dat eksternal yang menggunakn perintah dengan alamat 8 bit, port2 akan mengirimkan isi dari SFR P2. port2 juga menerima alamat bagian tinggi selama pemrograman dan *verifikasi flash*

Port3

Port3 merupakan port I/O dwi arah dengan dilengkapi *pullup* internal. Penyangga keluaran port3 mampu memberikan/menyerap arus empat masukan TTl (sekitar 1,6 mA)

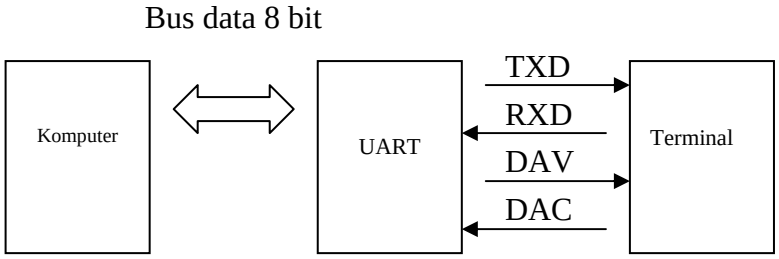
Jika ‘1’ dituliskan ke kaki-kaki port3, maka masing-masing kaki akan di *pullup high* dengan *pullup* internal sehingga dapat digunakan sebagai masukan. Sebagai masukan, jika kaki-kaki port3 dihubungkan ke *ground*, maka masing-masing kaki akan memberikan arus karena di *pullup high* secara internal

Port3, sebagimana port1, memiliki fungsi-fungsi alternatif antara lain menerima sinyal-sinyal kontrol (P3.6 dan P3.7), bersama-sama dengan port2 (P2.6 dan P2.7) selama pemrograman dan *verifikasi flash*

2.4. Pengiriman Data Serial

Di dalam sistem komputer, karakter-karakter disajikan dalam bentuk data yang terdiri dari sederetan angka biner atau bit (*binary digit*). Setiap bit hanya bernilai biner ’1’ atau biner ’0’. pemindahan, penyimpanan, dan pengolahan data di dalam komputer, atau mikroprosesor dapat dikerjakan berdasar atas operasi 8 bit, 16 bit atau 32 bit, tergantung jenis komputer yang digunakan. Setiap 8 bit disebut 1 *byte*. Diluar komputer atau mikroprosesor, data dapat dikirimkan ke periferal, terminal atau modem menggunakan cara pengiriman seri atau paralel. Dalam kesempatan ini hanya akan dibahas mengenai pengiriman seri.

Pengiriman seri biasanya digunakan untuk sambungan dengan jarak relatif lebih jauh. Gambar 2.2 menunjukkan konsep dasar pengiriman seri. Data paralel internal dimasukkan ke pengubah paralel ke seri. Pengubah paralel ke seri biasanya dengan IC dan juga melakukan sejumlah fungsi yang lain dan dikenal sebagai UART, VART, ACIA, PIA dan lain-lain. Kanal seri mengirimkan setiap karakter per elemen sehingga hanya diperlakukan dua penghantar, yaitu kirim data (TXD) dan terima data (RXD). Bit-bit dikirimkan secara berurutan dan tidak serempak, kecepatan pemindahan data lebih rendah dibanding pengiriman secara paralel. Pengiriman akan dimulai dari LSB (*least significant bit*), dan diakhiri dengan MSB (*most significant bit*). Setiap karakter yang dikirimkan, disajikan dengan suatu urutan bit tertentu sesuai dengan sandi yang digunakan. Penerima harus mencacah isyarat data yang sama, pada waktu yang tepat sebelum membentuk kembali karakter yang diterima.



Gambar 2.2. Pengiriman Serial

Pengiriman seri menimbulkan tiga masalah penyesuaian yaitu penyesuaian bit, penyesuaian karakter, dan penyesuaian blok. Ketika data seri akan dikirimkan, agar diterima dengan benar data seri tersebut, selang waktu yang digunakan oleh pengirim dan penerima harus sama satu terhadap yang lain. Untuk itu, pengirim dan penerima harus menambahkan detak. Istilah detak (*clock*) digunakan untuk menunjuk sembarang pulsa sumber pewaktuan (*timing pulse*). Detak penerima harus menunjukkan waktu yang tepat kapan isyarat harus dicacah oleh penerima untuk menentukan status logika dari setiap bit yang diterima

2.4.1. Tata Cara Komunikasi Data Serial

Dikenal dua cara komunikasi data secara serial, yaitu komunikasi data serial secara sinkron dan komunikasi data serial secara asinkron. Pada komunikasi data serial sinkron, *clock* dikirimkan bersama-sama dengan data serial, sedangkan

komunikasi data serial secara asinkron, clock tidak dikirimkan bersama data serial, tetapi dibangkitkan secara sendiri-sendiri baik pada sisi pengirim (*transmitter*) maupun pada sisi penerima (*receive*). Pada IBM PC (*personal computer*) kompatibel port serialnya termasuk jenis asinkron. komunikasi data serial ini dikerjakan oleh UART (*Universal Asynchronous Receiver/Transmitter*). IC UART dibuat khusus untuk mengubah data paralel menjadi data serial dan menerima data serial menjadi data paralel. Selain berbentuk IC mandiri, berbagai macam mikrokontroler ada yang dilengkapi UART, salah satunya adalah mikrokontroler AT89C51.

Pada UART, kecepatan pengiriman data (*baud rate*) dan *fase clock* pada sisi *transmitter* dan pada sisi *receiver* harus sinkron. Untuk itu diperlukan sinkroniasi antara *transmitter* dan *receiver*. Hal ini dilakukan oleh bit *start* dan bit *stop*. Ketika saluran transmisi dalam keadaan *idle*, output UART adalah dalam keadaan logika '1'. Ketika *transmitter* ingin mengirimkan data, output UART akan diset lebih dulu ke logika '0' untuk waktu satu bit. Sinyal ini pada *receiver* akan dikenali sebagai sinyal *start* yang digunakan untuk mensikronkan *fase clock*nya sehingga sinkron dengan *fase clock transmitter*, data akan dikirimkan secara serial dari bit paling rendah (bit 0) sampai bit tertinggi. Selanjutnya akan dikirim sinyal *stop* sebagai akhir dari pengiriman data serial.

Kecepatan transmisi (*baud rate*) dapat dipilih bebas dalam rentang tertentu. *Baud rate* yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400 dan 9600 (bit/detik). Dalam komunikasi data serial, *boud rate* dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama, selanjutnya harus ditentukan panjang data (6, 7 atau 8 bit), paritas (genap, ganjil atau tanpa paritas), dan jumlah bit *stop* (1, 1.5 atau 2 bit)

2.4.2. RS232

Standar sinyal komunikasi serial yang banyak digunakan adalah standar RS232 yang dikembangkan oleh *Electronic Industry Association and the Telecommunications Industry Association* (EIA/TIA). Standar ini hanya menyangkut komunikasi data antara komputer (*data terminal equipment*-DTE, contoh terminal di komputer) dengan alat-alat pelengkap komputer (*data circuit*

terminating equipment-DCE, contoh modem, plotter dan lain-lain). Standar RS232 inilah yang biasa digunakan pada port serial IBM PC kompatibel.

Standar sinyal serial RS232 memiliki ketentuan level tegangan sebagai berikut:

1. Logika '1' disebut *mark* terletak antara –3 Volt hingga –25 Volt
2. Logika '0' disebut *space* terletak antara +3 Volt hingga +25 Volt
3. Daerah tegangan antara –3 Volt hingga +3 Volt adalah invalid level, yaitu daerah tegangan yang tidak memiliki level logika pasti sehingga harus dihindari. Demikian juga, level tegangan lebih negatif dari –25 Volt atau lebih positif dari +25 Volt juga harus dihindari karena tegangan tersebut dapat merusak *line driver* pada saluran RS232.

2.4.3. Konfigurasi Port Serial

Konektor port serial tersedia dalam dua jenis, yaitu konektor 25 pin (DB-25) dan 9 pin (DB-9). Keduanya memiliki konektor jantan dan betina. Konektor port serial pada komputer (DTE) adalah jenis jantan. Bentuk dan ukuran DB-25 identik dengan port paralel, sedang DB-9 tampak seperti gambar 2.3 dibawah ini.



Gambar 2.3 (a) Konektor Serial DB-9 (b) Bentuk Fisik Dari DB-9

Tabel 2.2 Konfigurasi Pin dan Nama Sinyal Konektor Serial DB-9

Nomor Pin	Nama Sinyal	Direction	Keterangan
1	DCD	In	Data Carrier Detect/Received Line Signal Detect
2	RxD	In	Received data
3	TxD	Out	Transmit Data
4	DTR	Out	Data Terminal Ready
5	GND	-	Ground
6	DSR	In	Data Set Ready
7	RTS	Out	Request To Send
8	CTS	In	Clear To Send
9	RI	In	Ring Indicator

Keterangan mengenai fungsi saluran RS232 pada konektor DB-9 adalah sebagai berikut:

1. *Received Line Signal Detect*, dengan saluran ini DCE memberitahukan ke DTE bahwa pada terminal masukan ada data masuk
2. *Received Data*, digunakan DTE menerima data dari DCE
3. *Transmit Data, digunakan DTE mengirimkan data ke DCE*
4. *Data Terminal Ready*, pada saluran ini DTE memberitahukan kesiapan terminalnya
5. *Signal Ground,* saluran ground
6. *Ring Indicator*, pada saluran ini DCE memberitahukan ke DTE bahwa sebuah stasiun menghendaki hubungan dengannya
7. *Clear To Send*, dengan saluran ini DCE memberitahukan bahwa DTE boleh mulai mengirim data
8. *Request To Send,* dengan saluran ini DCE diminta mengirim data oleh DTE
9. DCE *Ready*, sinyal aktif pada saluran ini menunjukan bahwa DCE sudah siap

Pada komputer IBM PC kompatibel biasanya kita dapat menemukan dua konektor port serial DB-9 yang biasa dinamai COM1 dan COM2. *Base address* COM1 biasanya adalah 1016 (3F8h) dan COM2 biasanya 760 (2F8h). Alamat tersebut adalah alamat yang biasa digunakan, tergantung dari komputer yang digunakan.. berikut adalah tabel register-register yang digunakan untuk komunikasi port serial beserta alamatnya

Tabel 2.3. Nama Register Yang Digunakan Beserta Alamatnya

Nama Register	COM1	COM2
TX Buffer	3F8h	2F8h
RX Buffer	3F8H	2F8H
Baud Rate Divisor Latch LSB	3F8h	2F8h
Baud Rate Divisor Latch MSB	3F9h	2F9h
Interrupt Enable Register	3F9h	2F9h
Interrupt Identification Register	3Fah	2Fah
Line Control Register	3FBh	2FBh

Tabel 2.3. Lanjutan

Modem Control Register	3FCh	2FCh
Line Status Register	3FDh	2FDh
Modem Status Register	3Feh	2Feh

Keterangan mengenai fungsi register-register tersebut adalah sebagai berikut:

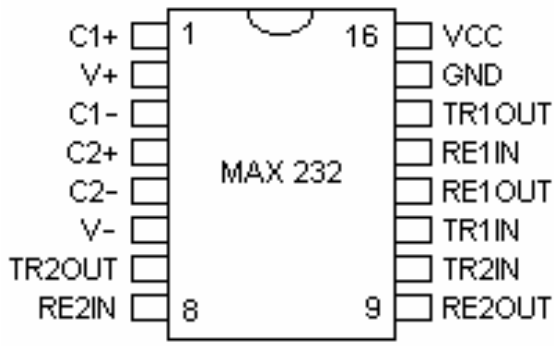
1. *RX Buffer*, digunakan untuk menampung dan menyimpan data dari DCE
2. *TX Buffer*, digunakan untuk menampung dan menyimpan data yang akan dikirim ke port serial
3. *Baud Rate Divisor Latch LSB*, digunakan untuk menampung byte bobot rendah untuk pembagi clock pada IC UART agar didapat *baud rate* yang tepat
4. *Baud Rate Divisor Latch MSB*, digunakan untuk menampung byte bobot rendah untuk pembagi clock pada IC UART sehingga total angka pembagi adalah 4 byte yang dapat dipilih dari 0001h sampai FFFFh.
5. *Interrupt Enable Register*, digunakan untuk menset interupsi apa saja yang akan dilayani komputer
6. *Interrupt Identification Register*, digunakan untuk menentukan urutan prioritas interupsi
7. *Line Control Register*, digunakan untuk menentukan jumlah bit data, jumlah bit *parity*, jumlah bit *stop*, serta untuk menentukan apakah *baud rate divisor* dapat diubah atau tidak
8. *Modem Control Register*, digunakan untuk mengatur saluran pengatur modem terutama saluran DTR dan saluran RTS.
9. *Line Status Register*, digunakan untuk menampung bit-bit yang menyatakan keadaan penerimaan atau pengiriman data dan status kesalahan operasi
10. *Modem Status Register*, digunakan untuk menampung bit-bit yang menyatakan status dari saluran hubungan dengan modem

2.4.4. MAX232

Port serial pada PC (*personal computer*) dengan standard RS232 merupakan salah satu fasilitas yang tersedia agar sebuah PC dapat dihubungkan dengan

peralatan luar. Diperlukan sebuah *chip interface* untuk mengkonversikan level tegangan yang berbeda yaitu *chip* MAX232.

Level tegangan yang berada pada tiap saluran port serial adalah sebesar ± 12 volt. Pada saat level tegangan ’1’ maka tegangan yang keluar pada port serial adalah sebesar –12 volt dan sebaliknya pada level logika ’0’ maka tegangan yang terukur pada port serial adalah sebesar + 12 volt, hal ini dikenal dengan *negatif logic* karena kondisi logika tertentu justru menghasilkan tegangan yang terbalik. MAX232 akan merubah level tegangan RS232 menjadi level tegangan TTL (5V) sekaligus mengubah logika negatif menjadi logika positif.



Gambar 2.4 IC MAX232 Sebagai Pengubah Level Tegangan

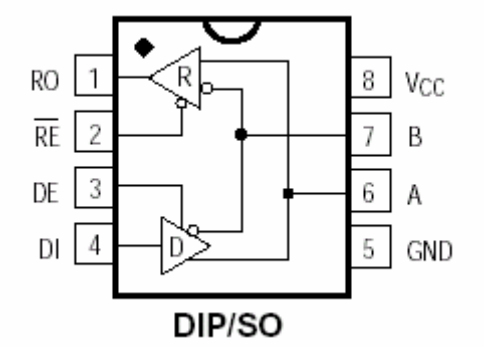
2.4.5. Komunikasi RS485

Rangkaian RS485 adalah suatu rangkaian jembatan (penghubung) antara piranti luar dengan perangkat-perangkat komputer

Standart RS485 ditetapkan oleh *Electronic Industries Association* dan *Telecomunication Industry Association* pada tahun 1983. Nama lengkapnya adalah *EIA/TIA-485*. Digunakan untuk membangun saluran transmisi sampai sejauh 4000 *feet*/ 1,2 Km dengan kecepatan maksimum lebih dari 1 Mb/detik, memiliki tegangan minimum -7V dan tegangan maksimum +12V.

Standard RS485 membicarakan karakteristik sinyal dalam transmisi data secara Balanced *Digital Multipoint System*. RS485 adalah teknik komunikasi data dapat dilakukan pada jarak yang cukup jauh yaitu 1,2 Km.

Konfigurasi pin dari RS485 ini adalah sebagai berikut :

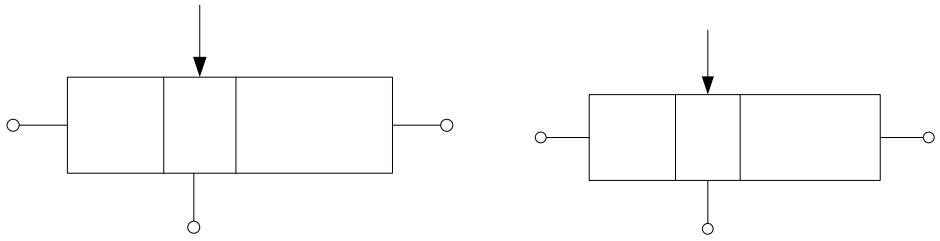


Gambar 2.5 Konfigurasi pin RS485

Berdasarkan pada gambar 2.5 diatas, output dari *Line Generator* dapat diambangkan (*High Impedance*) dengan memberikan nilai '0' pada input (DE), kemampuan ini dimaksudkan untuk menunjang keperluan dalam membentuk rangkaian saluran komunikasi *multidrop* yang menghendaki pada saluran hanya boleh satu *Line Generator* saja yang aktif. Apabila kaki DE berlogika 0 dan RE berlogika 0, maka SN75176 berfungsi sebagai penerima data sedangkan bila kaki DE berlogika 1 dan RE berlogika 1 maka SN75176 berfungsi sebagai penerima.

2.5. Transistor Sebagai Saklar

Transistor adalah komponen elektronik yang mempunyai fungsi sebagai penguat sinyal kecil. Pada umumnya transistor ini disebut juga sebagai transistor bipolar karena bekerja dengan 2 (B_1) muatan yang berbeda yaitu elektron sebagai pembawa muatan negatif dan hole sebagai pembawa muatan positif. Jenis-jenis transistor bipolar dapat dilihat pada gambar 2.6

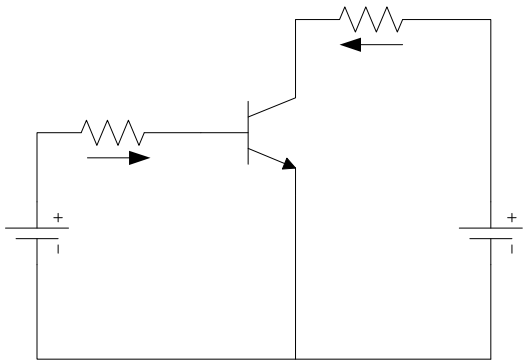


(a) (b)

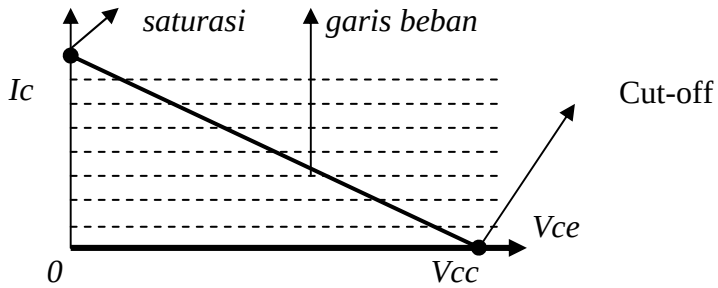
Gambar 2.6(a) Transistor npn, (b) Transistor pnp

Transistor pada gambar 2.6(a) mempunyai dua sambungan (junction), satu diantaranya emiter-basis dan basis-kolektor. Karena inilah, sebuah transistor sama seperti dua buah dioda. Transistor npn disebut juga dioda emiter basis atau singkatnya dioda emiter. Transistor pnp juga disebut dioda kolektor basis atau dioda kolektor.

Gambar 2.6(b) menunjukkan kemungkinan yang lain, yaitu sebuah transistor pnp. Transistor pnp merupakan komplemen dari transistor npn. Pembawa muatan mayoritas pada emiter adalah *hole*, sebagai pengganti dari muatan bebas. Ini berarti, pada transistor pnp dibutuhkan arus dan tegangan yang berlawanan dengan transistor npn.



(a)



(b)

Gambar 2.7 (a) Rangkaian bias basis (b) Garis beban dc

Gambar 2.7(a) memperlihatkan rangkaian bias basis. Sebuah sumber tegangan V_{BB} membias maju emiter melalui resistor R_B yang juga berfungsi membatasi arus. Penjumlahan tegangan disekitar *loop input* memberikan :

$$I_B R_B + V_{BE} - V_{BB} = 0 \quad \textbf{.....(2.1)}$$

Sehingga arus bias pada basis adalah :

$$I_B = \frac{V_{BB} - V_{BE}}{R_B} \quad \textbf{.....(2.2)}$$

Dengan V_{BE} = 0,7 V untuk transistor silikon dan 0,3 V untuk germanium. Dalam rangkaian kolektor, sumber tegangan Vcc membias balik dioda kolektor melalui Rc. Persamaan tegangan kolektor emiter dapat diperoleh melalui hukum ohm, yaitu :

$$V_{CE} = V_{CC} - I_C R_C \quad \textbf{.....(2.3)}$$

Dalam rangkaian bias basis yang diperlihatkan gambar 2.9(a), Vcc dan R_C adalah konstan. Pada persamaan 2.3 apabila disederhanakan akan dapat ditentukan besarnya arus Ic, seperti terlihat pada persamaan 2.4

$$I_C = \frac{V_{CC} - C_{CE}}{R_C} \quad \textbf{.....(2.4)}$$

Gambar 2.7(b) menunjukan grafik dari persamaan 2.4 memotong kurva dari kolektor. Perpotongan vertikal adalah Vcc/Rc dan perpotongan horizontal pada Vcc. Garis ini disebut garis *beban dc* karena garis ini menyatakan semua titik operasi yang mungkin. Perpotongan dari garis beban dc dengan arus basis adalah titik operasi kerja dari transistor. Titik di mana garis beban memotong kurva I_B = 0 disebut titik sumbat (*cut-off*). Pada titik ini, arus basis nol dan arus kolektor sangat kecil, sehingga dapat diabaikan (hanya ada arus bocor I_{CE0}). Pada titik sumbat, dioda emiter tidak lagi dibias maju dan transistor kehilangan kerja normalnya. Untuk itu digunakan suatu pendekatan, bahwa tegangan kolektor emiter adalah :

$$V_{CE(cutoff)} \cong V_{CC} \quad \textbf{.....(2.6)}$$

Perpotongan dari garis beban dan kurva $I_B = I_{B(sat)}$ disebut saturasi. Pada titik ini, arus basis sama dengan $I_{B(sat)}$ dan arus kolektor adalah maksimum. Pada

saturasi, dioda kolektor tidak lagi dibias balik dan transistor kehilangan kerja normalnya. Untuk itu digunakan suatu pendekatan bahwa arus kolektor pada saturasi adalah seperti diperlihatkan pada persamaan 2.7.

$$I_{C(sat)} \cong \frac{Vcc}{Rc} \text{ .....(2.7)}$$

dan arus basis yang tepat menimbulkan saturasi adalah seperti diperlihatkan pada persamaan 2.8.

$$I_{B(sat)} = \frac{I_{C(sat)}}{\beta dc} \text{ .....(2.8)}$$

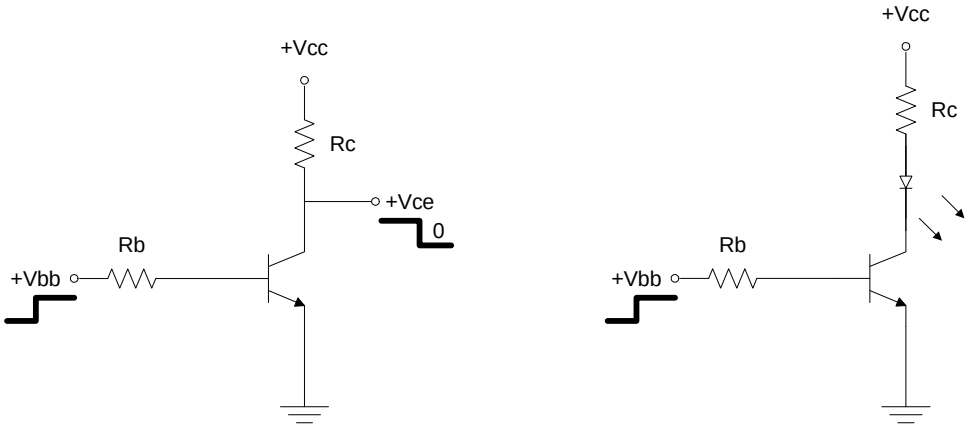
Dengan βdc merupakan penguatan arus. Salah satu penggunaan dari transistor adalah sebagai switch atau saklar, artinya bahwa mengoperasikan transistor pada salah satu dari saturasi atau titik sumbat, tetapi tidak di tempat-tempat sepanjang garis beban. Jika sebuah transisitor berada dalam keadaan saturasi, transistor seperti sebuah switch yang tertutup dari kolektor ke emiter. Jika transistor tersumbat (*cut-off*), transistor seperti sebuah switch yang terbuka. Dalam transistor dikenal istilah aturan disain *soft saturation* dan *hard saturation*. *Soft saturation* berarti kita membuat transistor hampir saturasi, dimana arus basis hanya cukup untuk mangoperasikan transistor pada ujung atas dari garis beban. *Soft saturation* tidak dapat diandalkan pada produksi massal karena adanya perubahan-perubahan pada βdc dan $I_{B(sat)}$. *Soft saturation* akan mengacu pada rancangan di mana transistor akan jenuh secara terbatas, dalam hal ini penguatan arus hanya sedikit lebih kecil dari penguatan arus aktif.

Pada kondisi *hard saturation*, berarti terdapat arus basis yang cukup kuat untuk membuat transistor saturasi pada semua harga dari βdc. Untuk keadaan yang paling buruk dari temperatur dan arus, hampir semua transistor silikon sinyal kecil mempunyai βdc lebih besar daripada 10. karena itu, semua pedoman desain untuk *hard saturation* adalah mempunyai arus basis kira-kira sepersepuluh dari harga saturasi arus kolektor, ini menjamin *hard saturation* pada semua kondisi kerja. Sebagai contoh, jika ujung atas garis beban mempunyai arus kolektor sebesar 10 mA, maka akan didapatkan arus basis sebesar 1 mA. Hal ini menjamin keadaan saturasi untuk semua transistor, arus, temperatur dan sebagainya. Digunakan

aturan 10:1 dalam proses mendesain rangkaian *switching* transistor, kecuali jika ditentukan lain. Jika nilai tahanan standar menghasilkan perbandingan I_C/I_B sedikit lebih besar daripada 10, hampir setiap transistor sinyal kecil akan menuju keadaan *hard saturation*.

Gambar 2.8(a) menunjukkan sebuah rangkaian *switching* transistor yang digerakkan oleh tegangan step. Jika tegangan *input* nol, transistor tersumbat (*cut-off*). Dalam hal ini, transistor kelihatannnya sebuah switch yang terbuka. Dengan tidak adanya arus yang mengalir melalui tahanan kolektor, maka tegangan *output* sama dengan $+V_{BB}$.

Gambar 2.8(b) menunjukkan rangkaian *switching* transistor dengan sedikit variasi. Rangkaian ini disebut *LED driver*, karena transistor mengendalikan LED. Jika tegangan *input* rendah (*low*), transistor akan tersumbat (*cut-off*) dan LED padam. Jika tegangan *input* tinggi (*high*), transistor saturasi dan LED menyala.



(a) Switching transistor step

(b) Switching transistor sebagai LED driver

Gambar 2.8 Contoh Transistor yang Digunakan Sebagai Switch

2.6. Optocoupler

Optocoupler merupakan salah satu jenis komponen yang memanfaatkan sinar sebagai pemicu *on/off*-nya. Bisa diartikan bahwa optocoupler merupakan suatu komponen yang bekerja berdasarkan picu cahaya optic.

Optocoupler merupakan bagian dari *photo detector* yaitu rangkaian elektronika yang dapat mengubah besaran cahaya menjadi besaran listrik. Rangkaian ini terdiri dari LED (*Light Emiting Diode*) yang terhubung secara optik dengan NPN *silicon photo transistor*

Arus input melalui LED akan menimbulkan cahaya yang akan mempengaruhi besar kecilnya arus yang mengalir dari kolektor ke emiter. Keuntungan dari komponen ini adalah kopling yang dilakukan melalui cahaya yang menyebabkan adanya isolasi listrik antara *input* dan *output*, akibatnya rangkaian *input output* ini dapat memiliki VCC dan *ground* yang berbeda

Sensor *optocoupler* terdiri dari *transceiver* (pemancar) dan *receiver* (penerima), dimana ada *infrared* sebagai *transceivernya* yang terhubung ke VCC dan *fhoto transistor* sebagai *receiver*. Pada saat ada cahaya yang mengenai basis atau tidak ada penghalang maka *optocoupler* akan bernilai *low* ‘0’, tapi kalau ada penghalang maka cahaya dari *infra red* tidak mengenai basis maka *outputnya* bernilai *high* ‘1’. Dalam rangkaian sensor terpasang transistor yang digunakan sebagai saklar dimana *output* yang berasal dari sensor semula bernilai *high*, maka oleh transistor dirubah menjadi *low*.



Gambar 2.9 Optocuopler OPB 703

2.7. Perangkat Lunak (*Software*)

2.7.1. Bahasa Assembly

Assembler adalah program komputer yang men-*translitrasi* program dari bahasa *assembly* ke bahasa mesin. Sedangkan bahasa *assembly* adalah ekuivalensi bahasa mesin dalam bentuk alpanumerik. *Mnemonics* alpanumerik digunakan sebagai alat bantu bagi *programmer* untuk memprogram mesin komputer daripada menggunakan serangkaian 0 dan 1 (bahasa mesin) yang panjang dan rumit. Program sumber *assembly* terdiri dari kumpulan baris-baris perintah dan biasanya disimpan dengan *extension* .ASM dengan satu baris untuk satu perintah, setiap baris perintah tersebut bisa terdiri atas beberapa bagian, yakni bagian label, bagian *mnemonic,* bagian operand yang bisa lebih dari satu dan terakhir bagian komentar.

Program sumber (*source code*) dibuat dengan program *pinnacle* 52. Hasil kerja program yang telah dikompile dalam bahasa *assembler* ini adalah "*assembly listing*" dan juga "file dengan *extention* .HEX". File dengan *extention* HEX ini adalah yang akan diisikan ke dalam chip mikrokontroler. Ketentuan penulisan *source code* adalah sebagai berikut :

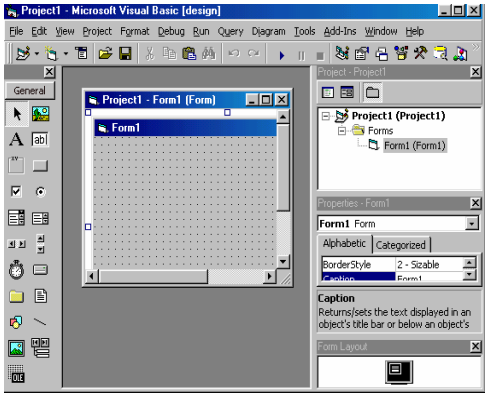
1 Masing-masing bagian dipisahkan dengan spasi atau TAB, khusus untuk *operand* yang lebih dari satu masing-masing *operand* dipisahkan dengan koma.
2 Bagian-bagian tersebut tidak harus semuanya ada dalam satu baris, jika ada satu bagian yang tidak ada maka spasi atau TAB sebagai pemisah bagian tetap hasus ditulis.
3 Bagian label ditulis mulai huruf pertama dari baris, jika baris bersangkutan tidak mengandung label maka label tersebut digantikan dengan spasi atau TAB, yakni sebagai tanda pemisah antara bagian label dengan bagian *mnemonic*.

2.7.2. *Visual Basic* 6.0

Visual basic atau sering disingkat VB adalah perangkat lunak untuk menyusun program aplikasi yang bekerja dalam lingkungan sistem operasi windows. Seperti program berbasis windows lainnya, *Visual basic* terdiri dari banyak windows.

Ketika akan memulai *Visual basic* sekumpulan windows yang paling berkaitan inilah yang disebut dengan *Integrated Development Environtment* (IDE). Program yang berbasis windows bersifat *event driven*, artinya program bekerja berdasarkan *event* dalam objek tersebut, misalnya jika seorang user mengklik sebuah tombol maka program akan memberikan ’reaksi’ terhadap *event* klik tersebut. Program akan memberikan reaksi sesuai dengan kode-kode program yang dibuat untuk suatu *event* pada objek tertentu.

Pada waktu memulai *Visual basic*, beberapa windows kecil berada disebuah windows besar (windows induk), bentuk inilah yang dikenal dengan format *Multiple Document Interface* (MDI). Berikut ini adalah contoh tampilan IDE pada sebuah *project Visual basic* dengan sebuah *form* dan sebuah *command button*



Gambar 2.10. Tampilan IDE Visual Basic

Windows - windows *Visual basic*

1. Menu/Toolbar

Windows ini adalah satu-satunya elemen dari IDE yang selalu tampak berguna untuk memilih ataupun menambah elemen IDE lain, menambah *form* dan menambah kontrol pada suatu *project*



Gambar 2.11. Menu/Toolbar

2. Toolbox

Toolbox adalah sekumpulan kontrol yang dapat diletakkan pada *form* program aplikasi yang dibuat. *Toolbox* dapat disembunyikan untuk memberikan ruangan

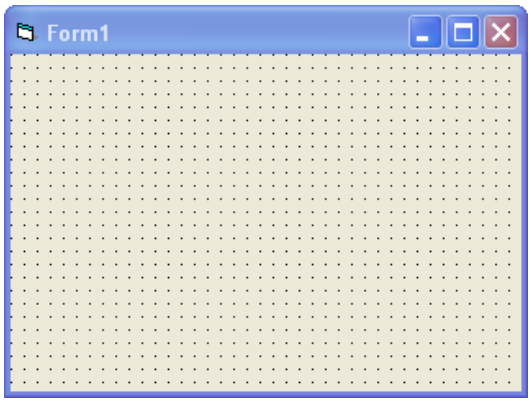
bagi elemen IDE lainnya, sehingga lebih mempermudah proses desain maupun penulisan programnya.



Gambar 2.12. Toolbox

3. Form

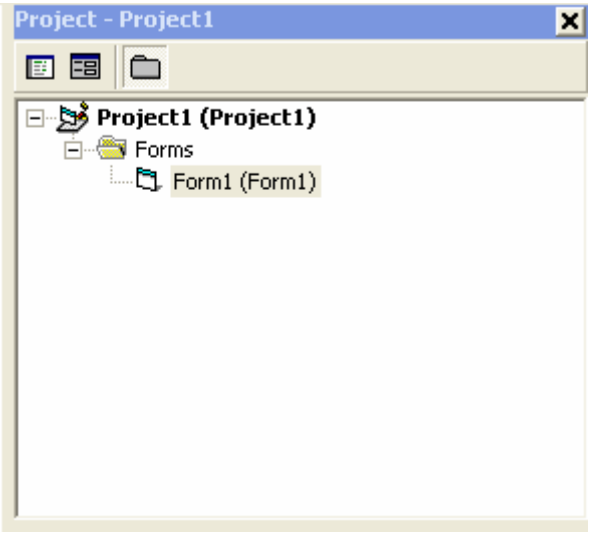
Form adalah sebuah atau beberapa windows untuk pembuatan program aplikasi. *Form* ini dapat memuat berbagai macam kontrol baik tombol-tombol, maupun teks yang diperlukan dalam desain program yang sesuai dengan kebutuhan program.



Gambar 2.13. Form

4. *Project Windows*

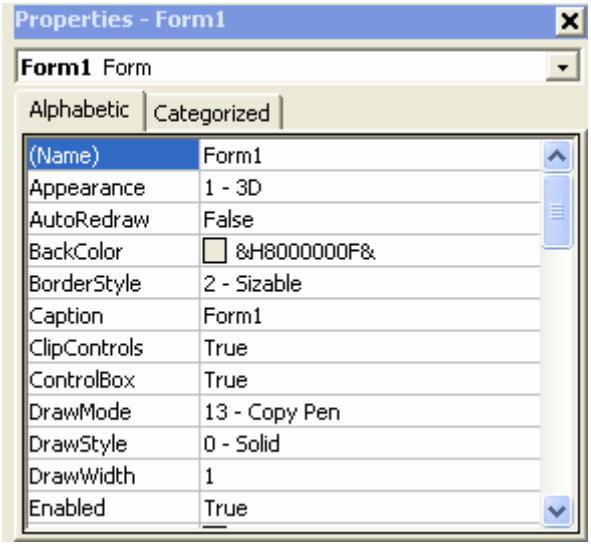
Windows ini menampilkan seluruh *form, class, class module,* dan komponen lain yang ada pda sebuah *project*



Gambar 2.14. Project Windows

5. *Property Windows*

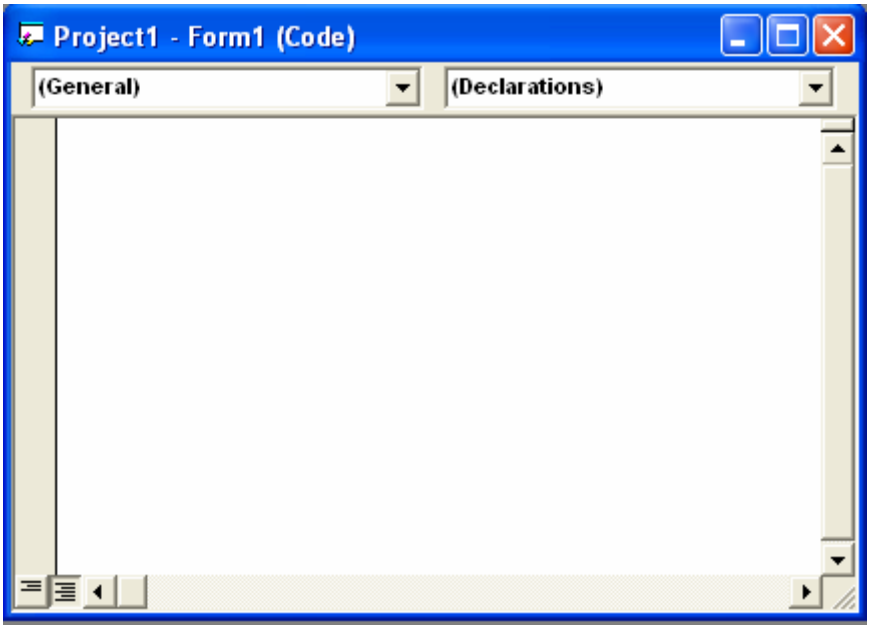
Windows ini berisi seluruh *property* dari masing-masing objek pada sebuah *project* yang meliputi *property form* dan kontrol-kontol yang ada pada *form* tersebut. Beberapa *property* dapat diisikan pada tahap desain dan ada pula *property* yang harus diisikan dengan menuliskan kode selama program dijalankan.



Gambar 2.15. Property Windows

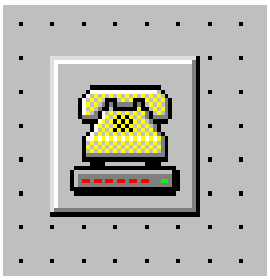
6. *Code Windows*

Pada windows inilah semua kode atau perintah tentang program dituliskan dengan memperhatikan *event* apa saja yang diperlukan



Gambar 2.16. Code Windows

Untuk pengksesan port serial kita dapat mengaksesnya secara langsung dengan menggunakan kontrol MSComm yang telah disediakan *Visual basic*. Simbol dari MSComm itu sendiri seperti yang ditunjukkan oleh gambar 2.17 dibawah ini.



Gambar 2.17. Simbol MSComm

Kontrol MSComm menyediakan fasilitas komuniksi antara program aplikasi yang kita buat dengan port serial untuk mengirim atau menerima data melalui port serial. Setiap MSComm hanya menangani satu port serial sehingga jika kita ingin menggunkan lebih dari satu port serial, kita juga harus menggunakan MSComm sebanyak port serial yang kita pakai. Property MSComm yang sering dipakai adalah sebagai berikut:

CommPort :Digunakan untuk menentukan nomor port serial yang akan dipakai

Setting :Digunakan untuk menset nilai *baud rate*, *parity*, jumlah bit data, dan jumlah bit *stop*

PortOpen :Digunakan untuk membuka ataupun menutup port serial yang dihubungkan dengan MSComm ini

Input :Digunakan untuk mengambil data *string* yang ada pada *buffer* penerima

Output :Digunakan untuk menulis data *string* pada *buffer* kirim