

BAB II

DASAR TEORI

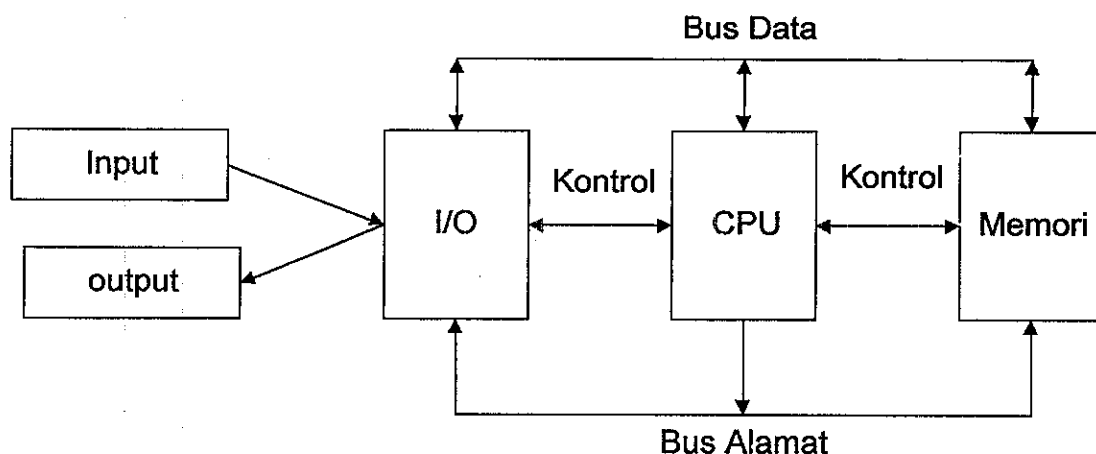
2.1. Umum

Pemanfaatan komputer secara maksimal tidaklah harus dengan menambah peralatan yang mahal dan dengan mengubah konfigurasi dari komputer tetapi cukup dengan menambah sebuah *interface card* dan programnya. Kartu interface ini dipasang pada slot-slot dari komputer yang tersedia dalam komputer yang belum digunakan. Slot-slot inilah yang akan digunakan sebagai tempat pemasangan antarmuka.

Dengan penambahan kartu interface ini maka akan diharapkan kemampuan dari komputer dapat digunakan secara maksimal. Kartu interface ini berguna untuk menghubungkan komputer dengan peralatan di luar komputer.

2.2. Komputer

Secara sederhana blok diagram komputer terlihat pada gambar 2.1. Bagian utama dari komputer adalah *Central Processing Unit* atau CPU, memori, dan *Input/output circuit* atau I/O. Untuk menghubungkan ketiga bagian utama dari komputer secara paralel digunakan tiga penghubung yang disebut bus. Ketiga bus itu adalah *address bus*, *data bus* dan *control bus*. (Hall, 1987)



Gambar 2.1.
Diagram blok mikrokomputer
(Hall, 1987)

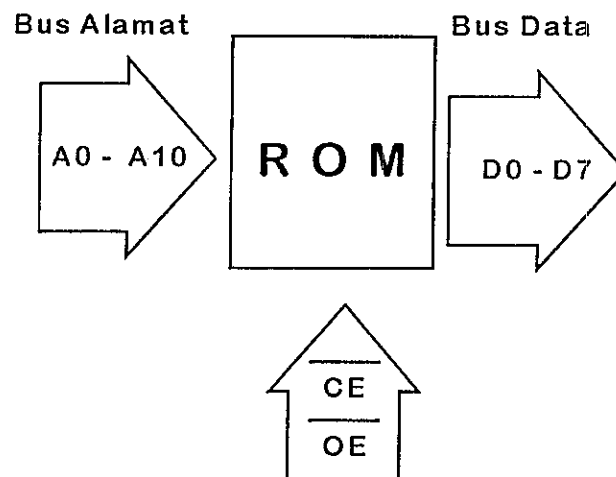
2.2.1. Memory

Memory digunakan untuk menyimpan kode biner dari instruksi yang akan dijalankan oleh CPU, juga digunakan untuk menyimpan data. Memori ini dapat berupa RAM yang hanya dapat menyimpan program atau data secara sementara atau ROM yang dapat menyimpan program atau data secara tetap.

2.2.1.1. ROM (*Read Only Memory*)

ROM merupakan jenis memori yang datanya tidak hilang ketika catu daya dimatikan. Gambar 2.2 adalah diagram blok ROM. Pada

blok tersebut terdapat tiga bagian ROM yang dinamakan masukan alamat, data keluaran, dan masukan data.



gambar 2.2.

Diagram blok ROM
(Rizal Rizkiawan, 1997)

Bus alamat digunakan untuk memilih data yang tersimpan pada lokasi ROM. Banyaknya lokasi yang tersedia adalah 2^n alamat. Jadi jika ROM terdiri dari 10 bit alamat, lokasi yang mungkin adalah $2^{10} = 1024$ lokasi.

Setiap bit data tersimpan pada ROM dalam bentuk biner 1 atau 0. Jika banyaknya lokasi dikalikan dengan banyaknya bit dalam tiap lokasi, untuk alamat 10 bit dan 8 bit data setiap lokasi akan diperoleh kapasitas 1024×8 bit. Pada blok diagram, *output enable* (OE) dan *chip enable* (CE) digunakan untuk mengaktifkan ROM. (Rizal Rizkiawan, 1997)

Untuk membaca ROM harus diberikan alamat pada sinyal alamat (misalkan A0 sampai A9). Alamat ini akan dikodekan oleh ROM untuk menentukan lokasi yang ingin dipilih. Kemudian harus diberikan sinyal 0 pada CE dan OE untuk membuat ROM berfungsi, dan data akan berada di jalur data keluaran.

2.2.1.2. RAM (*Random Acces Memory*)

Sifat dari RAM adalah sementara. Isi RAM akan hilang apabila sumber listrik dimatikan. Dengan kata lain, RAM masih kosong atau tak tentu isinya ketika sumber listrik dimasukkan (Busono, 1992).

Sedangkan macam-macam dari RAM ada 3 yaitu sebagai berikut (Rizal Rizkiawan 1997):

a. RAM Statik

Pada RAM ini data disimpan dalam deretan flip-flop sehingga data dapat ditulis dan dibaca. Jika catudaya dimatikan data akan hilang.

b. RAM Dinamik (DRAM)

Data disimpan dengan mengisi kapasitor kecil sehingga hanya dapat bertahan beberapa millidetik. Sebelum data hilang harus dilakukan penyegaran (*Refreshing*). Jika catu daya dimatikan, data akan hilang.

c. Non-Volatile RAM (NOVRAM)

Berisi RAM statik dan eprom yang dipasang paralel. Jika catu daya mati, data dari RAM dengan cepat ditransfer ke eprom. Jika catu daya kembali hidup data ditranfer kembali ke RAM.

2.2.2. Input/Output.

Bagian Input/output atau I/O ini berfungsi agar komputer mengambil data dari dunia luar komputer atau mengirim data ke dunia luar komputer. (Hall,1986). Keyboard, terminal video display, printer, modem dan sinyal dari pengindera (sensor) dihubungkan dengan komputer oleh bagian I/O ini. Dengan demikian bagian I/O memungkinkan pengguna dan komputer untuk saling berkomunikasi.

I/O memisahkan komputer dengan dunia luar sehingga disebut juga sebagai antarmuka (*interface*) (Busono, 1992). Antarmuka yang digunakan untuk menghubungkan bus komputer dengan sistem eksternal disebut "port". Port masukan digunakan untuk membaca data dari keyboard, pengubah dari analog ke digital (ADC) atau sumber data yang lain yang dapat dimengerti oleh komputer dan di bawah control CPU. Port keluaran digunakan untuk mengirim data dari komputer ke terminal video display, pencetak, atau

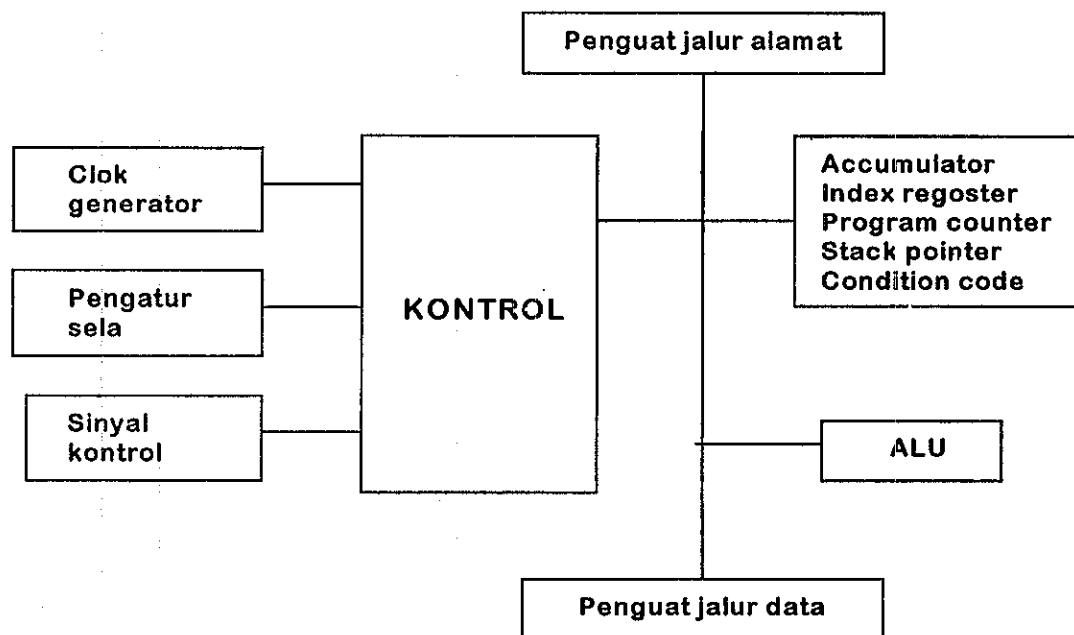
pengubah dari digital ke analog (DAC).

I/O juga membantu CPU agar mampu melayani aneka alat masukan/keluaran. I/O sebagai antar muka mengatur semua sinyal agar cocok dengan bahasa CPU.

2.2.3. CPU (*Central Processing Unit*)

CPU berfungsi untuk mengatur dan mengkoordinasi seluruh kegiatan mikrokomputer. CPU akan menjemput instruksi dari memori, dan menterjemahkan kode biner menjadi aksi yang bersesuaian. Dalam mewujudkan CPU dapat digunakan rangkaian-rangkaian logika yang dirancang secara cermat, tetapi yang lebih ekonomis pada saat ini adalah menggunakan mikroprosesor.

Secara garis besar, isi CPU dapat dikotak-kotak seperti pada gambar 2.3.



Gambar 2.3.
Diagram blok perangkat keras CPU
(Busono, 1992)

Penguat jalur alamat/data berfungsi untuk memperkuat CPU agar mampu melayani aneka perabotnya dan untuk penguat jalur data harus bersifat dua arah. CPU menulis dan membaca lewat jalur data yang sama. Jadi jalur data selain harus kuat juga harus dua arah. Sedangkan jalur alamat tidak perlu dua arah karena selalu berasal dari CPU.

Clock generator berfungsi untuk membangkitkan detak. CPU adalah rangkaian yang rumit sekali. Masing-masing bagian diatur dengan tata waktu yang sangat cermat agar selaras dan cepat. Untuk itu CPU digerakkan oleh lebih dari satu "clock".

Sela (*interrupt*) ialah tanda dari perabot komputer bahwa ada pekerjaan penting yang harus didahulukan CPU. Sarana sela ini tidak mutlak dibutuhkan oleh CPU namun sangat membantu meningkatkan kecepatan komputer dan mempermudah pemrograman.

Sinyal kontrol adalah aneka sinyal yang bertugas menyelaraskan kerja CPU. Sinyal sinyal ini belum tentu sama pada berbagai merek CPU. Yang umum ada misalnya reset, read, write, dan lain-lain. Sinyal-sinyal ini perlu diperhatikan pada saaty merancang perangkat keras tambahan agar serempak dengan CPU.

Accumulator (Ac) adalah tempat mengolah data, misalnya untuk membaca/ mengisi memori, menambah, mengurangi, membandingkan. Bila data yang diolah lebih dari sebuah, maka salah satunya harus berada di Ac. Kemudian hasil olah disimpan di Ac pula. Jadi isi Ac bisa berubah setelah olah data. Ini merupakan cirikhas dari Ac.

Index register (IR) berfungsi untuk register serba guna dan sebagai penunjuk indeks (nomor urut). Pekerjaan mengindeks berhubungan dengan tabel/daftar. IR tidak mutlak diperlukan oleh CPU namun karena pada prakteknya banyak hal yang dipermudah, maka semua CPU pasti mempunyainya. Bahkan ada yang lebih dari sebuah.

Program Counter (PC) adalah tempat catatan langkah (alamat) program. Isi dari PC adalah alamat tujuan CPU selanjutnya. Pada saat CPU

membaca tugas, bagian kontrol menentukan jumlah byte perintah tersebut. PC lalu diubah isinya agar menunjukkan alamat berikutnya yang harus dituju setelah perintah itu dikerjakan. Dengan demikian CPU segera tahu kemana akan pergi sehingga tidak membuang-buang waktu.

Stack pointer (SP) berisi alamat stack yang siap dipakai. Bila CPU melakukan tugas yang menyangkut stack, maka CPU juga mengubah isi SP agar selalu menunjukkan alamat stack yang siap di pakai. Stack merupakan bagian dari RAM yang dicadangkan untuk mencatat isi register.

Condition code register (CC) berisi rambu-rambu yang menandai hasil perintah terakhir (yang baru dikerjakan). Rambu ini sangat membantu pembuatan aliran program. Isi dari CC tidak berubah sampai diubah oleh CPU atau program.

Aritmetic Logic Unit (ALU) adalah pusat dari CPU. Disinilah letak kemampuan berfikir dari suatu CPU. Walaupun dikatakan mampu berpikir ternyata kebiasaan CPU tidak banyak. Kemampuan berpikir rumit harus disusun dari sederetan kebiasaan yang tidak banyak itu. Sebagian besar pekerjaan ALU dilakukan di Ac. Setiap menyelesaikan sebuah tugas, ALU memperbaharui CC seperlunya. Isi CC lalu dipakai untuk mengarahkan aliran program.

2.2.4. Bus Alamat

Banyaknya alamat yang dapat dituju ditentukan oleh jumlah bit alamat. Sedangkan alamat sendiri digunakan untuk membedakan satu perabot-perabot dalam komputer agar dapat dibedakan satu dengan yang lain (Busono, 1992).

Setiap perabot komputer harus disiapkan agar tanggap terhadap alamat tertentu. Atau sebaliknya, setiap perabot hanya dibedakan oleh nomor alamatnya saja. Dengan alamat yang jelas data akan sampai ke tujuan yang diinginkan. Karena komputer bekerja dengan bilangan biner, maka ia memerlukan kejelasan yang lebih jelas, misalnya bila alamat dinyatakan dalam n bit maka alamat yang jelas adalah yang ditulis dengan lengkap n bit. Ini dilakukan agar tidak terjadi alamat yang dianggap oleh komputer mempunyai kemiripan. Apabila ini terjadi maka alat-alat ini akan tanggap bersama-sama, namun tidak dijamin akan serempak karena akan berakibat saling adu logika.

2.2.5. Data bus

Bus data juga terdiri dari beberapa bit. Sedangkan komputer saat ini umumnya bekerja dengan 16, 32, atau 64 bit. Dari gambar 2.1 data bus mempunyai anak panah yang bermata dua hal ini menunjukkan

bahwa bus data bersifat *bidirectional* (Douglas V. Hall, 1986). Ini berarti bahwa CPU dapat membaca data dari memori atau dari port lewat bus data dan dapat mengirim data ke memori atau port juga melalui bus data ini. Semua peralatan dari sistem yang outputnya dihubungkan dengan data bus tetapi output hanya diberikan pada satu peralatan pada saat waktu tertentu.

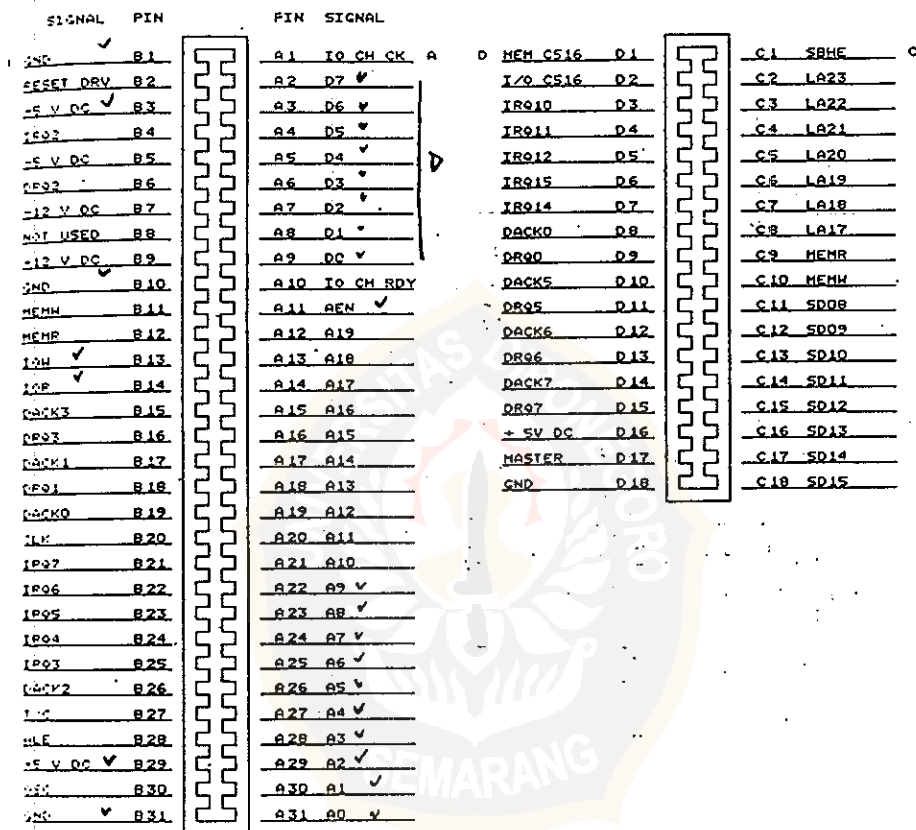
2.2.6. Kontrol bus

Jumlah sinyal kontrol ini sangat tergantung pada komputernya. Sinyal kontrol bertugas menentukan apa yang harus dilakukan dengan sinyal alamat dan sinyal data. Sinyal kontrol disini mempunyai sumber dan tujuan. Karena CPU merupakan pusat segala kegiatan maka sinyal kontrol harus mengacu pada CPU. Sumber dan tujuan ini tidak berhenti pada CPU saja tetapi berlanjut ke komputer. Dalam hal sinyal kontrol telah keluar dari komputer, maka istilah untuk CPU kini berlaku bagi komputer.

2.3. Slot Perluasan IBM-PC XT/AT

Sistem komputer IBM-PC XT/AT dirancang sebagai suatu sistem yang mudah dikembangkan. Hal ini terlihat dengan adanya slot

perluasan pada papan induk yang berfungsi untuk menghubungkan sistem komputer dengan peralatan-peralatan tambahan lain. Masing-masing slot terdapat 62 jalur penghubung (gambar 2.4). Tidak semua jalur yang tersedia digunakan pada pembuatan perangkat antar muka.



Gambar 2.4.

Blok diagram pengembangan

Untuk pembuatan perangkat antarmuka, pada skripsi ini digunakan 25 jalur yang terdiri dari 8 jalur untuk bus data, yakni D0..D7, 1 jalur AEN, 1 Jalur GND, 1 jalur IOW, 1 jalur IOR, 1 jalur CLK, 1 jalur VCC 5 volt, 1 jalur RST, dan 10 jalur bus alamat.

Fungsi dari jalur-jalur tersebut adalah :

1. Saluran bus data (D0 .. D7) nomor kaki penyemat (pin) A2 .. A9. D0 menunjukkan bit dengan signifikan terendah dan D7 menunjukkan bit dengan signifikansi tertinggi. Data bus merupakan jalur yang menyediakan pemindahan data antara memori pada mikrokomputer dan peralatan input/output secara dua arah.
2. AEN (*Adress Enable*) pada pin A11. Sinyal pada pin ini aktif pada status logika rendah (0) sebagai kendali penyedia alamat.
3. GND pada pin B1. Pin ini sebagai pentanahan (*ground*) dan penghubung untuk memberikan status logika rendah (0).
4. VCC +5 pada pin B3. Pin ini sebagai penyedia catu daya (*power Supply*) dan penghubung untuk memberikan status logika tinggi.
5. IOW (I/O Write) pada pin B14. Keluaran pin ini merupakan sinyal yang aktif pada status logika rendah (0). Saat sinyal ini aktif maka bus alamat akan berisi alamat gerbang I/O dan bus data akan berisi data yang akan ditulisi.
6. IOR (I/O Read) pada pin B13. Sinyal keluaran pin ini aktif pada status logika rendah (0) dan menunjukkan kepada I/O bahwa siklus bus merupakan siklus baca dengan alamat yang ada pada bus alamat.
7. RST pada pin B2. Pin ini aktif pada status logika tinggi (1).

8. Jalur bus alamat (A0 .. A9) pada pin A22 .. A31. Jalur ini digunakan untuk mengamati ingatan dan peralatan I/O dimana A0 adalah bit dengan signifikan terendah (LSB) dan A9 adalah bit dengan signifikansi tertinggi (MSB).

2.4. ADC (*Analog to Digital Conversion*)

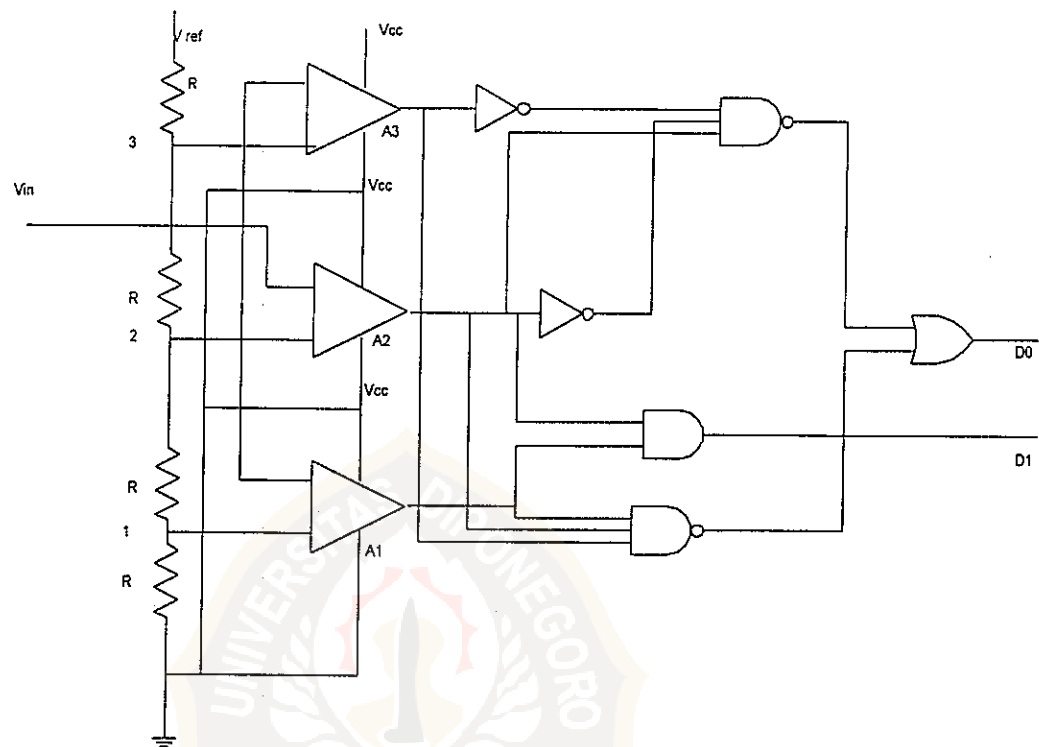
Seringkali data yang diperoleh dari suatu sistem fisis harus diubah ke dalam bentuk digital. Data tersebut biasanya terdapat dalam bentuk sinyal listrik analog. Sedangkan dalam pengaturan suhu cairan ini komputer menggunakan sinyal digital, sedangkan sensor suhu menghasilkan sinyal analog, maka digunakan suatu peralatan yang bisa mengubah sinyal analog menjadi sinyal digital. Alat tersebut adalah ADC (*Analog to Digital Conversion*).

Ada bermacam-macam metode yang dapat kita gunakan untuk mengubah kuantitas analog ke bentuk digitalnya. Di antaranya adalah metode perbandingan langsung, metode pencacah, metode pendekatan beruntun (*Rizal Rizkiawan, 1997*).

2.4.1. ADC metode perbandingan langsung

Gambar 2.5 merupakan rangkaian ADC metode perbandingan langsung. Cara kerja dari metode ini adalah dengan mengubah nilai-nilai

digital yang mungkin menjadi bentuk analog dan membandingkannya dengan sinyal analog dari masukan. Metode ini mempunyai waktu konversi sangat cepat tetapi memerlukan banyak komponen.



Gambar 2.5.
ADC metode pembanding langsung
(Riskal Rizkiawan, 1997)

Dari gambar 2.5. maka kita dapat mencari tegangan pada tiap-tiap titiknya. Misalkan $V_{ref} = 4\text{ V}$ maka :

Tegangan pada titik 1 :

$$\begin{aligned} V_{ref_1} &= R/4R \times V_{ref} \\ &= \frac{1}{4} \times 4\text{ V} \end{aligned}$$

$$= 1 \text{ V}$$

Tegangan pada titik 2 :

$$V_{ref_2} = 2R/4R \times V_{ref}$$

$$= \frac{1}{2} \times 4V$$

$$= 2 \text{ V}$$

Tegangan pada titik 2 :

$$V_{ref_3} = 3R/4R \times V_{ref}$$

$$= \frac{3}{4} \times 4 \text{ V}$$

$$= 3 \text{ V}$$

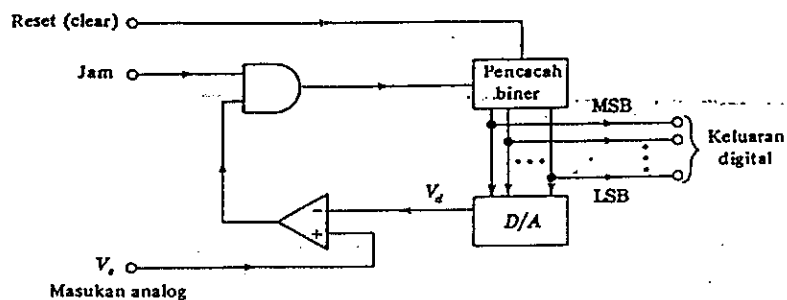
Dari perhitungan diatas dan gambar 2.5. maka dapat disusun tabel sebagai berikut :

Masukan analog	Keluaran Pembanding			Keluaran Biner	
	A3	A2	A1	D0	D1
$0 < V_{in} < 1$	0	0	0	0	0
$1 < V_{in} < 2$	0	0	1	0	1
$2 < V_{in} < 3$	0	1	1	1	0
$3 < V_{in} < 4$	1	1	1	1	1

2.4.2. ADC metode pencacah

Bagan dari ADC metode pencacah adalah terdapat pada gambar

2.6. Dalam metode ini digunakan suatu deretan kontinu dari pulsa-pulsa berjarak sama yang disalurkan melalui suatu gerbang.



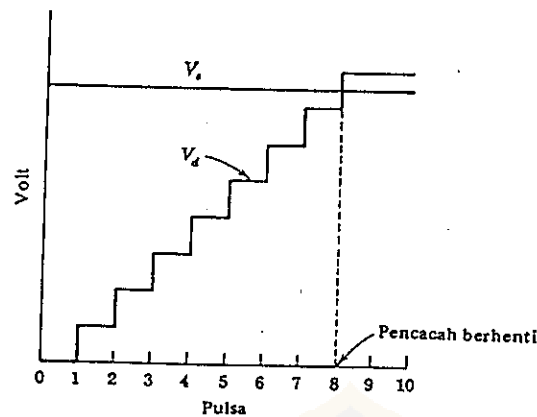
Gambar 2.6.

ADC metode pencacah
(Jacob Millman, 1985)

Prinsip kerja dari ADC metode pencacah adalah jika pulsa reset mengosongkan pencacah untuk hitungan 0. Kemudian pencacah akan mencatat dalam bentuk biner, jumlah pulsa dari saluran jam. Jam tersebut adalah sumber pulsa yang berjarak sama dalam waktu. Karena jumlah pulsa yang dicacah bertambah secara linear dengan waktu, kata biner yang mengungkapkan hitungan ini digunakan sebagai masukan dari suatu konverter D/A yang akan menghasilkan keluaran seperti pada gambar 2.7.

Selama masukan analog V_s lebih besar dari V_d , maka keluaran pembanding adalah tinggi dan gerbang and tetap terbuka bagi penyaluran pulsa jam kepada pencacah. Bilamana V_d melampaui V_s , keluaran komparator akan berubah kepada keadaan rendah, dan gerbang

and dilumpuhkan (*Millman & Halkias, 1985*). Prosedur ini akan menghentikan pencacahan pada saat $V_s \approx V_d$ dan hasil pencacah dapat dibaca sebagai bilangan digital yang mengungkapkan tegangan masuk analog.

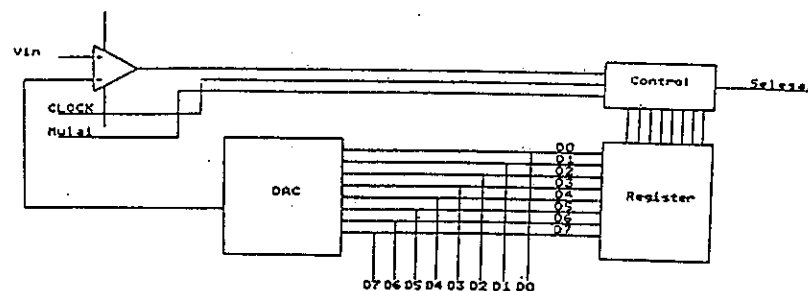


Gambar 2.7.

Bentuk gelombang lereng pencacah
(*Millman and Halkias, 1985*)

2.4.3. ADC Metode pendekatan beruntun

Metode ini paling banyak digunakan. Diagram blok dari ADC metode pendekatan beruntun terdapat pada Gambar 2.8.



Gambar 2.8.
ADC metode pendekatan beruntun
(Rizal Rizkiawan, 1997)

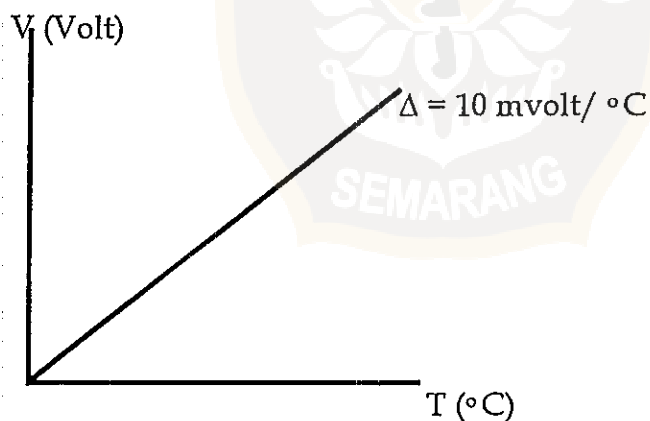
Cara kerja dari ADC pendekatan beruntun adalah sebagai berikut :

Bila sinyal MULAI diberi logika rendah, Vout akan menjadi nol. Ketika sinyal MULAI kembali tinggi terjadi konversi. Pada metoda ini pencacahan tidak naik satu persatu. Mula-mula bit MSB dibuat tinggi. Nilai biner yang dikeluarkan akan diubah oleh rangkaian DAC. Selanjutnya, keluaran DAC dibandingkan dengan tegangan masukan. Jika tegangan masukan lebih tinggi, keluaran pembanding akan membuat rangkaian kontrol tetap mengaktifkan MSB, sedangkan jika masukan lebih rendah, keluaran pembanding akan membuat rangkaian kontrol me-reset MSB. Kejadian tersebut akan diulang untuk bit-bit berikutnya.

2.5. Peraba (Sensor Suhu)

Pada instrumen ini ada dua besaran yang harus dihubungkan yaitu besaran suhu dan besaran listrik. Suhu yang akan diatur harus diterjemahkan dalam sinyal listrik karena pengontrol yang dipakai adalah elektrik. Untuk menghubungkan keadaan itu diperlukan suatu alat yang disebut sensor, dalam hal ini adalah sensor suhu ke listrik (*Anonim, 1985*).

Sensor harus dapat menterjemahkan suhu menjadi suatu besaran listrik baik berupa tegangan ataupun arus listrik. Banyak komponen yang bisa digunakan untuk sensor yang bisa digunakan untuk sensor ini misalnya NTC, bimetal, thermotransistor, zener dan lain-lain. Sedang pada sistem ini akan dipakai sensor yang mempunyai linearitas yang baik (gambar 2.9).



Gambar 2.9
Kelinieritasan yang baik

Komponen yang memenuhi kriteria itu dan yang akan dipakai adalah LM-335. Koefisien dari IC ini adalah $10 \text{ mV}/^{\circ}\text{C}$. Isinya berupa dioda zener yang peka terhadap suhu di sekelilingnya sehingga mengakibatkan tegangan zenernya berubah. Besarnya perubahan tegangan dapat dirumuskan sebagai berikut :

$$V_{\text{out } T} = V_{\text{out } T_0} \times \frac{T}{T_0} \dots\dots\dots (2.1)$$

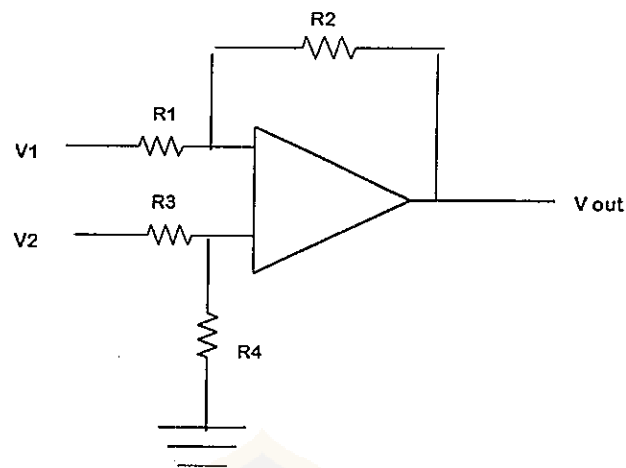
Dimana :

- $V_{\text{out } T}$: Tegangan pada suhu T
- $V_{\text{out } T_0}$: Tegangan pada suhu awal T_0 (pada saat kalibrasi)
- T_0 : Suhu pada saat kalibrasi
- T : Suhu pada saat dibaca

2.6. Penguat selisih tegangan dengan penguatan (op-amp)

Op-Amp adalah piranti solid-state yang mampu mengindera dan memperkuat sinyal masukan baik DC maupun AC. Penguat Op-amp cenderung mempunyai penguatan yang linear. Keluaran dikendalikan sebagai fungsi masukan (Gughes, 1990).

Sedangkan pembanding tegangan akan membandingkan tegangan sebuah masukan dengan tegangan masukan lainnya. Gambar rangkaian sederhana adalah seperti gambar 2.10.



Gambar 2.10.
penguat selisih tegangan dengan penguatan

dari gambar diatas maka kita dapat mengetahui V_{out} yaitu dari perbandingan antara V_1 dan V_2 sehingga V_{out} didapatkan :

$$V_{out} = \left(\frac{R_1 + R_2}{R_3 + R_4} \right) \frac{R_4}{R_1} V_2 - \frac{R_2}{R_1} V_1$$

Untuk $R_1 = R_3$ dan $R_2 = R_4$, maka

$R_1 // R_2 = R_3 // R_4$, jadi

$$V_{out} = \frac{R_2}{R_1} (V_2 - V_1) \dots\dots\dots (2.2)$$

2.7. Solid State Relay

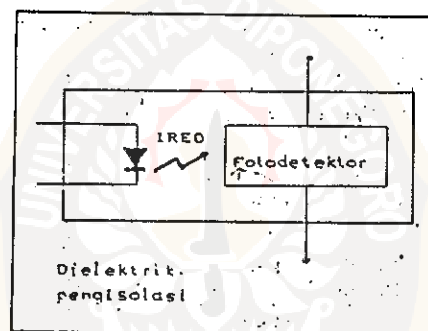
Untuk mengendalikan perangkat keras yang menggunakan catu daya AC 110 atau 220 volt dengan menggunakan mikrokomputer, maka diperlukan suatu perangkat keras pemisah *ground*. Lazimnya disebut pemisahan *hot ground* dan *cold ground*. *Hot ground* adalah groundnya perangkat keras yang menggunakan catu daya AC sedangkan *Cold ground* adalah groundnya perangkat keras yang menggunakan catu daya DC.

Relay mekanik mampu menghubungkan arus dari orde mili ampere sampai orde puluhan ampere. Relay mekanik memiliki beberapa masalah yang serius. Yang pertama dari semua masalah adalah karena terjadinya kontak terbuka dan tertutup yang terus menerus maka mata kontak akan teroksidasi dan lama-lama permukaan menjadi berlubang. Hal ini mengakibatkan hambatan dalam mata kontak membesar kemudian mata kontak menjadi panas yang dapat meleburkan mata kontak.

Untuk mengatasi masalah di atas maka digunakan *solid state relay*. *Solid state relay* dapat mengatasi masalah yang timbul di atas dan mampu memisahkan antara *hot ground* dan *cold ground*. Bagian utama dari *Solid state relay* adalah : *opto-couple* dan *Triac*. (Anonim, 1992)

2.7.1. Opto-coupler

Opto-coupler adalah suatu *devais* yang terdiri dari sedikitnya satu emiter (pemancar cahaya) yang mengkopel secara optik terhadap foto detektor melalui semacam medium terisolasi (Elektron, 1991). *Emiter* atau devais pemancar cahaya dapat berupa sebuah lampu pijar atau lampu neon atau LED (*Light Emiting Diode*). Medium isolasi dapat berupa udara, gelas, plastik atau fiber optik. Detektor dapat berupa fotokonduktor, fotodiode, dan fototriac. Diagram blok dari *opto-coupler* seperti pada gambar 2.11.



gambar 2.11.

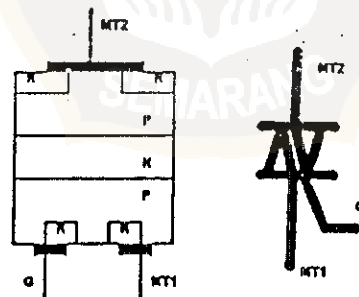
Blok diagram optocoupler
(Elektron, 1991)

Karena informasi dilewatkan secara optik melintasi celah isolasi, maka perpindahan terjadi dalam satu arah sehingga detektor tidak dapat mempengaruhi rangkaian input. Hal ini penting karena emiter mungkin dikendalikan oleh rangkaian bertegangan rendah yang menggunakan

sebuah mikroprosesor atau gerbang logik, sedangkan output foto detektor boleh jadi bagian dari tegangan tinggi DC atau bahkan rangkaian beban AC. Isolasi optik ini mencegah interaksi ataupun kerusakan terhadap rangkaian input yang disebabkan oleh perbedaan relatif rangkaian output.

2.7.2. Triac

Triac adalah salah satu jenis semikonduktor yang dapat melewatkan arus dalam dua arah apabila *pin gatenya* mendapat *triger*. Sedangkan pulsa triger dari *triac* ini dapat berupa pulsa *triger* positif atau pulsa *triger* negatif. Diagram blok dari *triac* adalah pada gambar 2.12.



gambar 2.12
susunan dalam triac
(anonim, 1992)

Dari gambar *triac* diatas dimana setiap terminal terhubung langsung dengan kedua semikonduktor yaitu jenis P dan semikonduktor jenis N, sehingga memungkinkan untuk aliran arus dalam dua arah. Pulsa *triger* pada *triac* sangat memungkinkan untuk pulsa positif maupun pulsa negatif. Arus utama dalam *triac* selalu mengalir melalui bahan semikonduktor jenis P - N - P - N dari terminal utama kesatu ke terminal utama kedua. Atau sebaliknya. Pemicuan pada pin *triger* menyebabkan terminal satu dan terminal dua terhubung (anonim 1988).

2.8. Perangkat lunak

Perangkat keras antarmuka memungkinkan terlaksananya tranfer data masukan dan keluaran melalui bus alamat, bus data dan bus kontrol. Perangkat lunak antar muka dapat bekerjasama dalam melaksanakan tugas yang diprogramkan.

Perangkat lunak (program komputer) ini akan disusun dengan menggunakan bahasa pemrograman tingkat tinggi yakni Bahasa Pemrograman Turbo Paskal Versi 7.0.

Bahasa pemrograman Turbo Pascal dengan perintah berbentuk "port" berguna untuk menentukan lokasi suatu alamat (*address*) tujuan maupun asal dari suatu data. Jika suatu nilai diberikan mengikuti

perintah port pada perangkat lunak maka instruksi tersebut berlaku sebagai keluaran dari mikrokomputer ke perangkat antar muka untuk diteruskan menuju peralatan tambahan yang digunakan.

Contoh :

```
Port[$ 220] := 8;
```

Nilai yang dimiliki suatu port dengan alamat yang telah ditentukan dapat diambil oleh mikrokomputer sebagai masukan.

Nilai masukan ini sebagai data masukan diambil dalam bentuk biner yang kemudian diubah menjadi nilai dalam bentuk desimal.

Contoh :

```
X:= port[$ 300];
```

