

## BAB 9

# DASAR SISTEM KONTROL

Satu dari pertanyaan yang sering ditanyakan oleh seorang pemula pada sistem kontrol adalah : Apakah yang dimaksud dengan sistem kontrol? Untuk menjawab pertanyaan itu, kita dapat mengatakan bahwa dalam kehidupan sehari-hari kita, terdapat sejumlah tujuan yang harus dicapai. Misalnya, dalam bidang rumah tangga, kita perlu mengatur suhu dan kelembaban rumah dan bangunan untuk kenyamanan hidup. Untuk transportasi, kita harus mengendalikan mobil dan pesawat untuk bergerak dari satu lokasi ke lokasi lainnya dengan aman dan akurat.



Gambar 9.1 Blok Diagram Sistem Kontrol Secara Umum

Pada bidang industri, proses manufaktur mempunyai sejumlah tujuan untuk mendapatkan hasil yang akan memuaskan permintaan ketelitian dan keefektifan biaya. Manusia mempunyai kemampuan untuk melaksanakan tugas dalam ruang lingkup yang luas, termasuk di dalamnya pembuatan keputusan. Beberapa tugas ini seperti mengambil benda dan berjalan dari satu tempat ke tempat lainnya, sering dikerjakan dengan cara yang biasa. Pada kondisi tertentu, beberapa dari tugas ini dilakukan dengan cara sebaik

mungkin. Misalnya, seorang pelari 100 yard mempunyai tujuan untuk berlari dalam jarak tersebut dalam waktu sesingkatnya. Seorang pelari maraton, tidak hanya harus berlari dalam, jarak tersebut secepat mungkin, tapi untuk mencapai hal tersebut dia harus mengatur pemakaian energi dan memikirkan cara terbaik untuk perlombaan tersebut. Cara untuk mencapai tujuan ini biasanya melibatkan penggunaan sistem kontrol yang melaksanakan strategi kontrol tertentu.

Kontrol otomatis (otomatis) telah memegang peranan yang sangat penting dalam perkembangan ilmu dan teknologi. Di samping sangat diperlukan pada pesawat ruang angkasa, peluru kendali, sistem pengemudian pesawat, dan sebagainya kontrol otomatis telah menjadi bagian yang penting dan terpadu dari proses-proses dalam pabrik dan industri modern. Misalnya, kontrol otomatis perlu sekali dalam kontrol numerik dari mesin alat-alat bantu di industri juga perlu sekali dalam operasi industri seperti pengontrolan tekanan, suhu, kelembaban, viskositas, dan arus dalam industri proses.

Karena kemajuan dalam teori dan praktek kontrol otomatis memberikan kemudahan dalam mendapatkan performansi dari sistem dinamik, mempertinggi kualitas dan menurunkan biaya produksi, mempertinggi laju produksi, meniadakan pekerjaan-pekerjaan rutin dan membosankan yang harus dilakukan oleh manusia, dan sebagainya, maka sebagian besar insinyur dan ilmuwan sekarang harus mempunyai pemahaman yang baik dalam bidang ini.

Sejarah Perkembangan system control dapat diceritakan sebagai berikut. Hasil karya pertama yang penting dalam kontrol otomatis adalah governor sentrifugal untuk pengontrolan kecepatan mesin uap yang dibuat oleh James Watt pada abad kedelapanbelas. Hasil karya lain yang penting pada tahap awal perkembangan teori kontrol dibuat oleh Minorsky, Hazen, Nyquist, dan sebagainya. Pada tahun 1922, Minorsky membuat kontroler otomatis untuk pengemudian kapal dan menunjukkan cara menentukan kestabilan dari persamaan diferensial yang melukiskan sistem.

Pada tahun 1932 Nyquist mengembangkan suatu prosedur yang relatif sederhana untuk menentukan kestabilan sistem loop tertutup pada basis respons loop terbuka terhadap masukan tunak (steady state) sinusoida. Pada tahun 1934 Hazen, yang memperkenalkan istilah servomekanisme untuk sistem kontrol posisi, membahas disain servomekanisme relay yang mampu mengikuti dengan baik masukan yang berubah.

Selama dasa warsa 1940-an, metode respons frekuensi memungkinkan para insinyur untuk mendisain sistem kontrol linear berumpan-balik yang memenuhi persyaratan kinerja. Dari akhir tahun 1940 hingga awal tahun 1950, metode tempat kedudukan akar dalam disain sistem kontrol benar-benar telah berkembang.

Metode respons frekuensi dan tempat kedudukan akar, yang merupakan inti teori kontrol fisik, akan membawa kita ke sistem yang stabil dan memenuhi seperangkat persyaratan kinerja yang hampir seimbang. Sistem semacam itu pada umumnya tidak optimal dalam setiap pengertian

yang berarti. Semenjak akhir tahun 1950, penekanan persoalan dalam disain sistem kontrol telah digeser dari disain salah satu dari beberapa sistem yang bekerja menjadi disain satu sistem optimal dalam suatu pengertian yang berarti.

Karena *plant* modern dengan multi-masukan dan multi-keluaran menjadi semakin kompleks, maka deskripsi sistem kontrol modern memerlukan banyak persamaan. Teori kontrol klasik, yang hanya membahas sistem satu masukan satu keluaran, sama sekali tidak dapat digunakan untuk sistem multi-masukan multi-keluaran. Semenjak sekitar tahun 1960, teori kontrol modern telah dikembangkan untuk mengatasi bertambah kompleksnya *plant* modern dan persyaratan yang keras pada ketelitian, berat, dan biaya untuk kebutuhan militer, ruang angkasa, dan industri.

Dengan adanya komputer elektronik analog, digital, dan hibrid yang dapat digunakan pada perhitungan-perhitungan yang kompleks, maka penggunaan komputer dalam disain sistem kontrol dan penggunaan komputer yang dipasang langsung pada sistem kontrol sekarang menjadi praktis dan umum. Komputer analog adalah istilah yang digunakan untuk menggambarkan alat penghitung yang bekerja pada level analog, dengan arus searah. Level analog di sini adalah lawan dari level digital, yang mana level digital adalah level tegangan *high* (tinggi) dan *low* (rendah), yang digunakan dalam implementasi bilangan biner (hanya mempunyai 2 jenis nilai, yaitu 0 atau 1). Secara mendasar, komponen elektronik yang digunakan sebagai inti dari komputer analog adalah op-amp.

Tentunya pada komputer digital yang bekerja adalah menggunakan level digital. Komputer yang sering kita jumpai sekarang ini termasuk dalam jenis komputer digital, kalkulator salah satunya. Sedangkan komputer hibrid sendiri merupakan gabungan antara komputer analog (dengan arus searah) dan digital. Dengan demikian kombinasi komputer analog yang memberikan kemampuan dalam hal kecepatan, keluwesan dan kemudahan untuk berkomunikasi langsung dengan kemampuan komputer digital dalam hal kecermatan, logika dan ingatan, maka sangatlah besar manfaatnya didunia keilmuan. Simulasi yang dinamik dan kemampuan pemecahan persamaan diferensial dengan kecepatan tinggi dapat dilaksanakan oleh bagian analog, sementara olahan statis dan aljabar dapat ditangani di bagian digital. Dengan demikian daya guna dan hal ekonomi, secara keseluruhan dari suatu sistem dapat dimaksimalkan.

Pada tahun-tahun belakangan ini, sistem kontrol memegang peranan penting dalam perkembangan dan kemajuan peradaban dan teknologi modern. Dalam prakteknya, setiap aspek aktivitas sehari-hari dipengaruhi oleh beberapa model sistem kontrol. Sistem kontrol sangat banyak ditemukan di setiap sektor industri, seperti pengendalian kualitas dari produk yang dihasilkan, lajur pemasangan otomatis, pengendalian mesin, teknologi luar angkasa dan sistem persenjataan, pengendalian komputer, sistem transportasi, sistem daya, robotik, dan lain-lain. Bahkan pengendalian dari sistem persediaan barang, sosial dan ekonomi dapat didekati dengan teori kontrol otomatis.

Perkembangan baru-baru ini dalam teori kontrol modern adalah dalam bidang kontrol optimal baik sistem deterministik (tertentu) maupun stokastik (acak), demikian juga kontrol belajar dan adaptif dari sistem yang rumit. Dewasa ini komputer digital telah menjadi lebih murah dan semakin ringkas, maka digunakan sebagai bagian integral dari sistem kontrol. Penerapan teori kontrol modern dewasa ini juga meliputi sistem yang bukan rekayasa, seperti sistem biologi, biomedikal, ekonomi dan sosial ekonomi.

Variabel yang dikontrol adalah besaran atau keadaan yang diukur dan dikontrol. Variabel yang dimanipulasi adalah besaran atau keadaan yang diubah oleh kontroler untuk mempengaruhi nilai variabel yang dikontrol. Dalam keadaan normal, variabel yang dikontrol adalah keluaran dari sistem. Kontrol berarti mengukur nilai dari variabel sistem yang dikontrol dan menerapkan variabel yang dimanipulasi ke sistem untuk mengoreksi atau membatasi penyimpangan nilai yang diukur dari nilai yang dikehendaki.

Pada penelaahan rekayasa, kita perlu menentukan istilah-istilah tambahan yang diperlukan untuk menjelaskan sistem kontrol, seperti misalnya: *plant*, *gangguan-gangguan*, *kontrol umpan balik*, dan *sistem kontrol umpan balik*. Berikut ini akan diberikan definisi-definisi tersebut. Kemudian penjelasan mengenai sistem loop tertutup dan loop terbuka, dan juga kelebihan-kelebihan dan kekurangan-kekurangannya dibandingkan dengan sistem kontrol loop terbuka dan loop tertutup. Akhirnya akan diberikan juga definisi sistem kontrol belajar dan adaptif.

Istilah *plant* didefinisikan sebagai seperangkat peralatan, mungkin hanya terdiri dari beberapa bagian mesin yang bekerja bersama-sama, yang digunakan untuk melakukan suatu operasi tertentu. Dalam buku ini, setiap obyek fisik yang dikontrol (seperti tungku pemanas, reaktor kimia, dan pesawat ruang angkasa) disebut *plant*.

Istilah proses (*process*) menurut kamus Merriam-Webster mendefinisikan proses sebagai operasi atau perkembangan alamiah yang berlangsung secara kontinu yang ditandai oleh suatu deretan perubahan kecil yang berurutan dengan cara relatif tetap dan menuju ke suatu hasil atau keadaan akhir tertentu; atau suatu operasi yang sengaja dibuat, berlangsung secara kontinu, yang terdiri dari beberapa aksi atau perubahan yang dikontrol, yang diarahkan secara sistematis menuju ke suatu hasil atau keadaan akhir tertentu. Dalam buku ini, setiap operasi yang dikontrol disebut proses. Sebagai contoh adalah proses kimia, ekonomi, dan biologi.

Istilah sistem (*system*) didefinisikan sebagai kombinasi dari beberapa komponen yang bekerja bersama-sama dan melakukan suatu sasaran tertentu. Sistem tidak dibatasi hanya untuk sistem fisik saja. Konsep sistem dapat digunakan pada gejala yang abstrak dan dinamis seperti yang dijumpai dalam ekonomi. Oleh karena itu, istilah *sistem* harus diinterpretasikan untuk menyatakan sistem fisik, biologi, ekonomi, dan sebagainya.

Istilah gangguan (*disturbances*) didefinisikan didefinisikan sebagai suatu sinyal yang cenderung mempunyai pengaruh yang merugikan pada harga keluaran

sistem. Jika suatu gangguan dibangkitkan dalam sistem, disebut internal, sedangkan gangguan eksternal dibangkitkan di luar sistem dan merupakan suatu masukan.

Istilah kontrol umpan balik dapat dijelaskan sebagai berikut. Kontrol umpan balik mengacu pada suatu operasi, yang dengan adanya gangguan, cenderung mengurangi perbedaan antara keluaran dari sistem dan suatu acuan masukan dan bahwa hal itu dilakukannya berdasarkan pada perbedaan ini. Di sini hanya gangguan yang tidak diperkirakan yang ditentukan demikian, karena gangguan yang dapat diperkirakan atau gangguan yang diketahui dapat selalu dikompensasi di dalam sistem tersebut.

Istilah sistem kontrol umpan balik. Sistem yang mempertahankan hubungan yang ditentukan antara keluaran dan beberapa masukan acuan, dengan membandingkan mereka dan dengan menggunakan perbedaan sebagai alat kontrol dinamakan sistem kontrol umpan balik. Contoh untuk sistem ini adalah sistem kontrol suhu ruangan. Dengan mengukur suhu ruangan sebenarnya dan membandingkannya dengan suhu acuan (suhu yang dikehendaki), termostat menjalankan alat pemanas atau pendingin, atau memamatkannya sedemikian rupa sehingga memastikan bahwa suhu ruangan tetap pada suhu yang nyaman tidak tergantung dari keadaan di luar.

Sistem kontrol umpan balik tidak terbatas di bidang rekayasa, tetapi dapat juga ditemukan di berbagai macam bidang bukan rekayasa. Tubuh manusia, misalnya, adalah sistem kontrol umpan balik yang sangat maju. Baik suhu tubuh

maupun tekanan darah dijaga tetap konstan dengan alat umpan balik faal tubuh. Kenyataannya, umpan balik melaksanakan fungsi yang vital. Ia membuat tubuh manusia relatif tidak peka terhadap gangguan eksternal, jadi memungkinkannya untuk berfungsi dengan benar di dalam lingkungan yang berubah.

Sebagai contoh yang lain, tinjau kontrol dari kecepatan mobil oleh operator manusia. Pengemudi memutuskan kecepatan, yang sesuai dengan suatu keadaan, yang mungkin adalah batasan kecepatan yang tertera pada jalan raya atau jalan bebas hambatan yang bersangkutan. Kecepatan ini bertindak sebagai kecepatan acuan. Pengemudi akan memperhatikan kecepatan sebenarnya dengan melihat speedometer. Jika dirasakan berjalan lebih lambat, ia akan menginjak pedal gas dan kecepatan mobil akan bertambah tinggi. Jika kecepatan sebenarnya terlalu tinggi, ia melepaskan pedal gas dan mobil akan menjadi lambat. Operator manusia ini dapat dengan mudah diganti oleh alat mekanik, listrik, atau yang serupa. Sebagai pengganti pengemudi yang memperhatikan speedometer, maka dapat digunakan generator listrik untuk menghasilkan tegangan yang sebanding dengan kecepatan. Tegangan ini dapat dibandingkan dengan tegangan acuan yang berkaitan dengan kecepatan yang dikehendaki. Perbedaan dalam tegangan ini kemudian digunakan sebagai sinyal kesalahan untuk menggerakkan tuas yang menaikkan atau menurunkan kecepatan sesuai dengan yang diperlukan.

Istilah Sistem Servo atau sistem servomekanisme didefinisikan

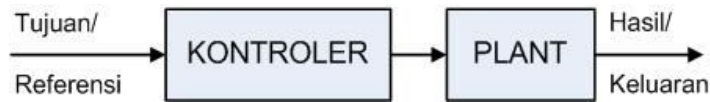
sebagai suatu sistem kontrol berumpan-balik dengan keluaran berupa posisi, kecepatan, atau percepatan mekanik. Oleh karena itu, istilah servomekanisme dan sistem pengontrolan posisi (atau kecepatan atau percepatan) adalah sinonim. Servomekanisme banyak digunakan dalam industri modern.

Contoh dari sistem servomekanisme ini adalah operasi mesin alat bantu yang otomatis secara menyeluruh atau lengkap, bersama-sama dengan instruksi yang diprogram, dapat dicapai dengan penggunaan sistem servo. Perlu diperhatikan bahwa sistem kontrol, yang keluarannya (seperti misalnya posisi pesawat terbang di angkasa pada suatu sistem pendaratan otomatis) perlu mengikuti jalan di angkasa yang telah ditentukan, dinamakan sistem servo juga. Contoh lainnya termasuk sistem kontrol lengan-robot, di mana lengan robot harus mengikuti jalan tertentu di ruangan yang telah ditentukan, dan sistem pendaratan otomatis pesawat udara, dengan pesawat udara harus mengikuti jalan di angkasa yang telah ditentukan.

Istilah sistem kontrol otomatis didefinisikan sebagai sistem kontrol yang mempunyai umpan balik dengan acuan masukan atau keluaran yang dikehendaki dapat konstan atau berubah secara perlahan dengan berjalannya waktu, mempunyai tugas utama yaitu menjaga keluaran sebenarnya berada pada nilai yang dikehendaki dengan adanya gangguan. Ada banyak contoh sistem kontrol otomatis, beberapa di antaranya adalah kontrol suhu ruangan mobil secara otomatis, pengatur otomatis tegangan pada *plant* daya listrik dengan adanya

variasi beban daya listrik, kontrol otomatis tekanan dan suhu dari proses kimiawi dan kontrol suhu secara otomatis di ruangan.

lain selama selang waktu tertentu yang lain pula. Pada pengontrolan dengan program seperti itu, titik setel diubah sesuai dengan jadwal waktu

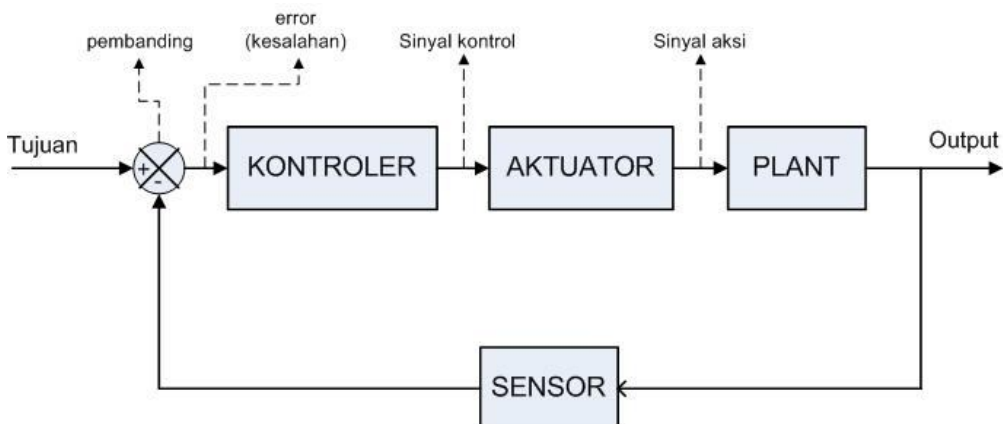


Gambar 9.2 Blok Diagram Sistem Kontrol Loop Terbuka

Istilah sistem pengontrolan proses (*process control system*) merupakan sistem kontrol secara otomatis dengan keluaran berupa besaran seperti temperatur, tekanan, aliran, tinggi muka cairan atau pH disebut sistem pengontrolan proses. Pengontrolan proses secara luas digunakan di industri. Pengontrolan dengan program seperti pengontrolan temperatur tungku pemanas dengan temperatur tungku dikontrol sesuai instruksi yang telah diprogram terlebih dahulu seringkali digunakan pada sistem seperti itu. Sebagai contoh, program yang harus

yang telah ditentukan. Kontroler (pengontrol) kemudian berfungsi untuk menjaga temperatur tungku agar mendekati titik setel yang berubah. Harus diperhatikan bahwa sebagian besar sistem pengontrolan proses servo mekanisme sebagai bagian yang terpadu.

Istilah sistem kontrol loop tertutup seringkali disebut sebagai sistem kontrol umpan balik. Secara praktis dan seringkali istilah kontrol umpan balik dan kontrol loop tertutup dapat saling dipertukarkan penggunaannya. Pada sistem kontrol loop tertutup,



Gambar 9.3 Blok Diagram Sistem Kontrol Loop Tertutup

diatur terlebih dahulu dapat berupa instruksi untuk menaikkan temperatur tungku sampai harga tertentu yang

sinyal kesalahan yang bekerja, yaitu perbedaan antara sinyal masukan sinyal umpan balik (yang mungkin

sinyal keluarannya sendiri atau fungsi dari sinyal keluaran dan turunannya), disajikan ke kontroler sedemikian rupa untuk mengurangi kesalahan dan membawa keluaran sistem ke nilai yang dikehendaki. Istilah kontrol loop tertutup selalu berarti penggunaan aksi kontrol umpan balik untuk mengurangi kesalahan sistem.

Istilah sistem kontrol loop terbuka dapat didefinisikan sebagai suatu sistem yang keluarannya tidak mempunyai pengaruh terhadap aksi kontrol disebut *sistem kontrol loop terbuka*. Dengan kata lain, sistem kontrol loop terbuka keluarannya tidak dapat digunakan sebagai perbandingan umpan balik dengan masukan. Suatu contoh sederhana adalah mesin cuci. Perendaman, pencucian, dan pembilasan dalam mesin cuci dilakukan atas basis waktu. Mesin ini tidak mengukur sinyal keluaran yaitu tingkat kebersihan pakaian.

Dalam suatu sistem kontrol loop terbuka, keluaran tidak dapat dibandingkan dengan masukan acuan. Jadi, untuk tiap masukan acuan berhubungan dengan kondisi operasi tertentu, sebagai akibat, ketetapan dari sistem tergantung pada kalibrasi. Dengan adanya gangguan, sistem kontrol loop terbuka tidak dapat melaksanakan tugas seperti yang diharapkan. Sistem kontrol loop terbuka dapat digunakan, hanya jika hubungan antara masukan dan keluaran diketahui dan tidak terdapat gangguan internal maupun eksternal.

Perbandingan antara sistem kontrol loop tertutup dan loop terbuka dijelaskan dibawah ini. Suatu kelebihan dari sistem kontrol loop tertutup adalah penggunaan umpan-balik yang membuat respons sistem

relatif kurang peka terhadap gangguan eksternal dan perubahan internal pada parameter sistem. Jadi, mungkin dapat digunakan komponen-komponen yang relatif kurang teliti dan murah untuk mendapatkan pengontrolan *plant* dengan teliti, hal ini tidak mungkin diperoleh pada sistem loop terbuka.

Dan segi kestabilan, sistem kontrol loop terbuka lebih mudah dibuat karena kestabilan bukan merupakan persoalan utama. Sebaliknya, kestabilan dapat menjadi persoalan pada sistem kontrol loop tertutup karena bisa terjadi kesalahan akibat koreksi berlebih yang dapat menimbulkan osilasi pada amplitudo konstan ataupun berubah.

Harus ditekankan bahwa untuk sistem dengan masukan yang telah diketahui sebelumnya dan tidak ada gangguan, maka disarankan untuk menggunakan kontrol loop terbuka. Sistem kontrol loop tertutup mempunyai kelebihan hanya jika terdapat gangguan yang tidak dapat diramal dan/atau perubahan yang tidak dapat diramal pada komponen sistem. Perhatikan bahwa batas kemampuan daya keluaran ikut menentukan biaya, berat, dan ukuran sebuah sistem kontrol. Jumlah komponen yang digunakan dalam sistem kontrol loop tertutup akan lebih banyak bila dibandingkan pada sistem kontrol loop terbuka. Sistem kontrol loop tertutup pasti membutuhkan instrumen untuk mengukur sebagian atau seluruh keluarannya. Oleh karena itu, sistem kontrol loop tertutup pada umumnya lebih besar dan mahal. Untuk memperkecil daya yang diperlukan oleh sistem, bila mungkin, dapat digunakan kontrol loop terbuka. Kombinasi yang sesuai antara kontrol

loop terbuka dan tertutup biasanya lebih murah dan akan memberikan kinerja sistem keseluruhan yang diinginkan.

Istilah sistem kontrol adaptif dijelaskan sebagai berikut. Karakteristik dinamik dari sebagian besar sistem kontrol adalah tidak konstan karena beberapa sebab, seperti memburuknya kinerja komponen dengan penambahan waktu atau perubahan parameter dan sekeliling (sebagai contoh, perubahan massa dan kondisi atmosfer pada sistem kontrol pesawat ruang angkasa). Walaupun pengaruh perubahan-perubahan kecil pada karakteristik dinamik diredam pada sistem kontrol berumpan-balik, jika perubahan sistem dan sekeliling cukup besar, maka suatu sistem yang baik harus mempunyai kemampuan untuk menyesuaikan diri (adaptasi). Adaptasi berarti kemampuan untuk mengatur diri atau memodifikasi diri sesuai dengan perubahan pada kondisinya sekeliling atau struktur yang tidak dapat diramal. Sistem kontrol yang mempunyai suatu kemampuan beradaptasi dalam keadaan bebas disebut *sistem kontrol adaptif*.

Pada sistem kontrol adaptif, karakteristik dinamik harus diidentifikasi setiap saat sehingga parameter kontroler dapat diatur untuk menjaga performansi optimal. Konsep ini menarik banyak perhatian disainer sistem kontrol karena sistem kontrol adaptif, di samping mengikuti perubahan sekeliling, juga akan menyesuaikan kesalahan-kesalahan atau ketidakpastian disain teknik yang layak dan akan mengkompensasi kerusakan sebagian kecil komponen-komponen sistem sehingga memperbesar keandalan sistem keseluruhan.

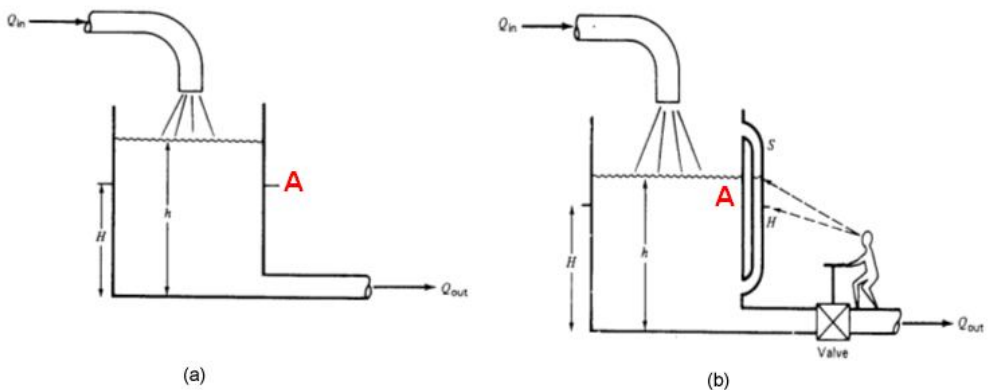
Istilah sistem kontrol dengan penalaran dijelaskan sebagai berikut di bawah ini. Beberapa sistem kontrol loop terbuka yang sering dijumpai dapat diubah menjadi sistem kontrol loop tertutup, jika operator manusia dipandang sebagai kontroler, membandingkan masukan dan keluaran kemudian melakukan aksi koreksi yang berdasarkan selisih atau kesalahan yang diperoleh.

Jika kita berusaha menganalisis sistem kontrol loop tertutup yang melibatkan operator manusia semacam itu, kita akan menjumpai persoalan yang sulit dalam menuliskan persamaan yang menggambarkan perilaku manusia. Salah satu dari beberapa faktor yang kompleks dalam kasus ini adalah kemampuan penalaran dari operator manusia. Jika operator mempunyai banyak pengalaman, ia akan menjadi kontroler yang lebih baik, dan hal ini harus diperhitungkan dalam menganalisis sistem semacam itu. Sistem kontrol yang mempunyai kemampuan untuk menalar disebut *sistem kontrol dengan penalaran* (learning control system). Konsep ini masih cukup baru dan menjadi kajian yang menarik.

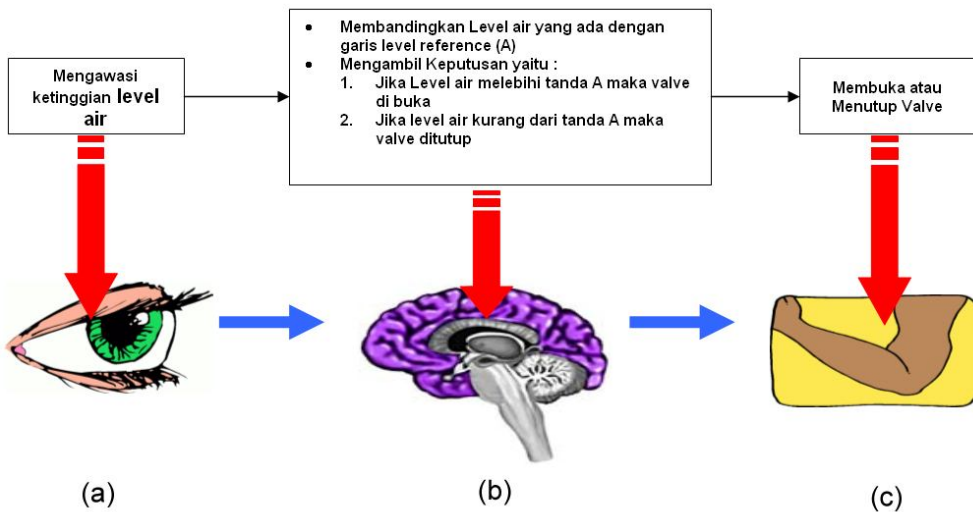
## 9.1 Ilustrasi Sistem Kontrol pada Manusia

Pada Gambar 9.4 (a) bisa dilihat gambar mengenai penampung/tangki air. Disana terdapat air dengan kedalaman  $h$ , debit air masuk sebesar  $Q_{in}$  dan debit air keluar sebesar  $Q_{out}$ . Besar dari air yang masuk tidak bisa kita prediksi. Bisa jadi  $Q_{in}$  besar sekali atautkah  $Q_{in}$  sangat kecil sekali, bahkan tidak mengalir sama sekali. Tanda A





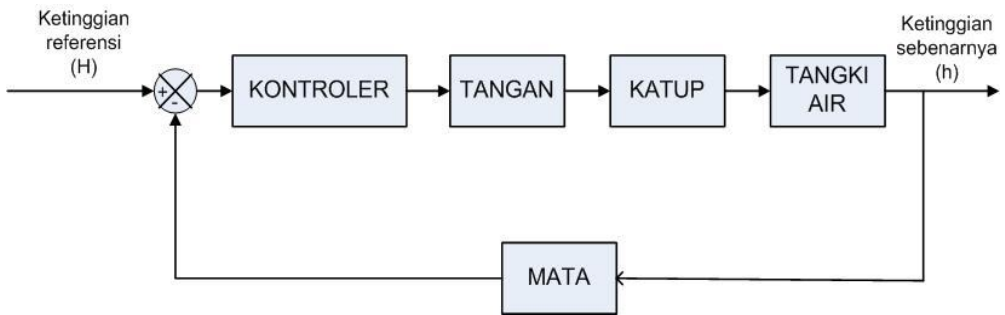
Gambar 9.4 Penampung Air (a) dan dengan Operator Manusia (b)



Gambar 9.5 Proses Kontrol pada Manusia (a) mata, (b) otak dan (c) tangan.

merupakan titik yang menunjukkan kedalaman/ketinggian air yang diinginkan, yaitu sebesar  $H$ . Diharapkan dari sistem ini ketinggian atau kedalaman air selalu sebesar  $H$ , tidak lebih tinggi dan tidak lebih rendah. Kalau memang yang terjadi adalah ketinggian atau kedalaman air tidak sama dengan  $H$  maka hal tersebut terjadi suatu kesalahan atau error.

Agar tujuan dari sistem tersebut bisa tercapai, yaitu mempertahankan nilai dari ketinggian atau kedalaman air sebenarnya ( $h$ ) selalu sama dengan  $H$ , maka diperlukan seorang operator untuk mengontrol setiap kondisi yang terjadi, seperti terlihat pada Gambar 9.4 (b). Apa yang dilakukan oleh seorang operator manusia tersebut agar ketinggian atau kedalaman air bisa dipertahan-



Gambar 9.6 Blok Diagram Proses Kontrol pada Manusia

kan pada level A (kedalaman sebesar H) ? Bagaimana proses yang terjadi dari ilustrasi tersebut ?

Tentunya dengan mudah bisa kita jawab ketika kita melihat Gambar 9.4 (b). Operator akan membuka atau menutup katup/valve pada pipa untuk arah keluarnya air. Tentunya jika kedalaman air sebenarnya (h) lebih besar dari nilai kedalaman yang diinginkan (H) maka seorang operator atau penjaga air tersebut akan membuka katup/valve, sehingga kedalaman air akan berkurang, tentunya nilai dari h akan mendekati H. Sebaliknya jika kedalaman air sebenarnya di bawah kedalaman air yang diharapkan, maka tentunya yang dilakukan oleh seorang operator atau penjaga air tersebut adalah menutup katup/valve, sehingga yang akan terjadi adalah ketinggian air yang sebenarnya akan naik seiring dengan mengalirnya air dari  $Q_{in}$ . Tampak bahwa katup tersebut seperti kran air yang bisa dibuka atau ditutup secara variabel, hubungannya dengan banyak sedikitnya volume air yang di alirkan keluar ( $Q_{ou}$ ).

Yang terjadi adalah dari pengamatan seorang operator tersebut dengan mata yang melihat kedalaman air hingga sampai ke tin-

dakan, yaitu menutup atau membuka valve. Bagaimana proses yang terjadi ? hal ini bisa dijelaskan sesuai Gambar 9.5. Masing-masing dari komponen tersebut yaitu mata, otak dan tangan bisa dijelaskan sebagai berikut :

a. Mata

Berfungsi untuk mengawasi ketinggian level air.

b. Otak

Mempunyai 2 fungsi yaitu membandingkan level air yang sebenarnya dengan garis level referensi (A) dan kemudian mengambil keputusan yaitu :

- Jika level air sebenarnya melebihi tanda A, maka valve harus dibuka
- Jika level air sebenarnya kurang dari tanda A, maka valve ditutup.
- Jika level air sebenarnya sama dengan tanda A, maka valve dibiarkan (tidak ditutup dan tidak dibuka)

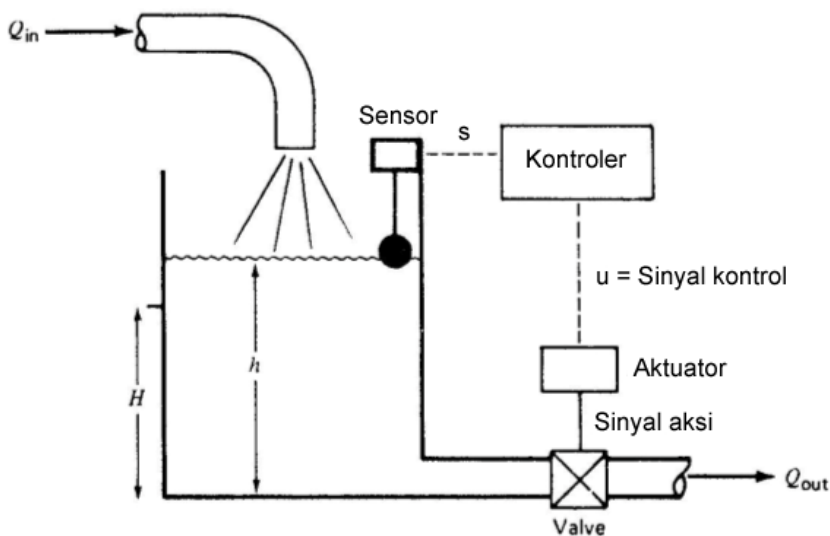
c. Tangan

Berfungsi untuk membuka atau menutup valve.

Dari blok diagram sesuai pada Gambar 9.6 bisa kita pahami bagaimana proses kontrol yang terjadi pada contoh *sistem kontrol mempertahankan level air* di atas. Ketinggian referensi ( $H$ ) dan Ketinggian sebenarnya ( $h$ ) dibandingkan oleh pembanding (tanda +/-). Dimana ketinggian sebenarnya diamati oleh sepasang mata, kemudian tentunya sinyal dari mata menuju otak dikirim melalui *syaraf sensorik*. Dalam hal ini pembanding tadi berada di *otak*. Sehingga didapatkan sinyal *error* atau *kesalahan*. Nilai dari error ini bisa bernilai positif, negatif ataukah nol. Kemudian oleh otak, diambil suatu

pipa air yang keluar berdasarkan hasil keputusan dari otak sebagai fungsi kontroler. Karena dibuka atau ditutup katupnya, mengakibatkan ketinggian level air tangki akan berubah, bisa semakin besar ataupun berkurang. Kemudian ketinggian air tersebut diamati oleh mata lagi yang kemudian dikirim ke otak, dibandingkan dan diambil keputusan oleh otak. Dan regulasi ini terjadi secara terus-menerus.

Apa yang terjadi antara harapan dan kenyataan ? tentunya akan terjadi kesalahan. Bagaimana jika tidak ada koreksi yang terjadi dari kesalahan yang ada, dalam hal ini tidak ada mata yang memperhatikan



Gambar 9.7 Sistem Kontrol Level Air secara Otomatis

keputusan sesuai dengan yang disebut di atas (ada 3 kemungkinan kondisi). Dalam hal ini fungsi otak sebagai kontroler. Sinyal kontrol yang dihasilkan oleh otak, dikirim melalui *syaraf motorik* ke tangan. Tentunya sinyal ini akan menyuruh tangan untuk membuka atau menutup katup

level dari ketinggian air. Tentunya kita akan berjalan didalam kebutaan, tidak tahu ketinggian level air sebenarnya (kenyataannya) berapa. Bisa saja yang terjadi adalah air terlalu sedikit atau air akan meluber karena terlalu banyak yang dialirkan. Disinilah diperlukan sistem kontrol yang lebih

akurat, yaitu diperlukannya balikan dari keluaran/kenyataan yang dihasilkan (level ketinggian) yang harus dibandingkan dengan referensi (harapan) kita. Dan tentunya kontroler juga harus bisa mengambil keputusan dengan akurat dan benar sesuai dengan yang seharusnya. Seperti inilah sistem kontrol dengan loop tertutup itu bekerja. Konsep seperti ini berlaku pada setiap sistem kontrol yang ada. Ada nilai referensi, pembanding, kontroler, aktuator, plant dan sensor. Istilah-istilah ini akan dijelaskan di bagian selanjutnya.

## 9.2 Sistem Kontrol Otomatis

Sistem kontrol otomatis adalah sistem kontrol umpan balik dengan acuan masukan atau keluaran yang dikehendaki dapat konstan atau berubah secara perlahan dengan berjalannya waktu dan tugas utamanya adalah menjaga keluaran sebenarnya berada pada nilai yang dikehendaki dengan adanya gangguan. Pemakaian sistem kontrol otomatis dalam segala bidang keteknikan masa kini semakin banyak dipakai. Hal ini disebabkan sistem kontrol otomatis mempunyai banyak keunggulan dibandingkan dengan sistem kontrol konvensional (manual), yaitu dari segi kecepatan, ketepatan dan pemakaian tenaga manusia yang relatif lebih sedikit. Apalagi ditunjang dengan pengembangan dunia elektronika, pneumatika maupun hidrolis. Banyak contoh sistem kontrol otomatis, beberapa di antaranya adalah kontrol suhu ruangan mobil secara otomatis, pengatur otomatis tegangan pada *plant* daya listrik di tengah-tengah adanya variasi beban daya listrik, dan kontrol otomatis tekanan dan suhu dari proses

kimiawi. Dalam sistem kontrol otomatis, terdapat elemen-elemen penyusun, yaitu :

### a. Sensor/Tranduser

Sensor adalah suatu komponen yang mendeteksi keluaran atau informasi lainnya yang diperlukan dalam siste kontrol. Sedangkan tranduser adalah suatu komponen yang mampu merubah besaran-besaran non listrik (mekanis, kimia atau yang lainnya) menjadi besaran-besaran listrik atau sebaliknya.

### b. Kontroler

Kontroler adalah suatu komponen, alat, atau peralatan (berupa mekanis, pneumatik, hidrolis, elektronik atau gabungan darinya) yang mampu mengolah data masukan dari membandingkan respon plant (hasil pembacaan dari keluaran plant) dan referensi yang dikehendaki untuk dikeluarkan menjadi suatu data perintah atau disebut *sinyal kontrol*.

### c. Aktuator

Aktuator adalah suatu komponen, alat atau peralatan (berupa mekanis, pneumatik, hidrolis, elektronik atau gabungan dari hal tersebut) yang mampu mengolah data perintah (sinyal kontrol) menjadi sinyal aksi ke suatu plant.

Untuk lebih mudah memahami cara kerja sistem kontrol otomatis, pada Gambar 9.7 diberikan contoh sistem kontrol secara otomatis pada aplikasi kontrol level air. Berbeda dengan bagian 9.2 di atas. Pada bagian ini sudah tidak menggunakan seorang operator manusia lagi untuk mempertahankan level air sesuai yang diinginkan, tetapi sudah menggunakan kontroler yang bekerja secara otomatis.

Sensor untuk aplikasi ini bisa berupa bahan pelampung dan tambahan komponen elektronik. Dengan komponen ini bisa diketahui berapa kedalaman atau ketinggian level air yang sebenarnya. Dari besaran fisika, yaitu kedalaman/ketinggian dengan satuan meter dirubah menjadi besaran listrik dengan satuan tegangan. Dengan adanya informasi ini, maka kontroler akan menghasilkan sinyal kontrol yang diolah sebelumnya. Kontroler bisa berupa rangkaian elektronik, mikrokontroler, mekanis, pneumatik, hidrolik ataupun gabungan dari nya. Karena sinyal kontrol tidak bisa langsung dimanfaatkan untuk memutar katup/valve pipa, maka sinyal ini harus dikonversi dulu menjadi sinyal aksi. Aktuatorlah yang mengkonversi sinyal ini. Aktuator dalam sistem ini bisa berupa motor listrik, komponen pneumatika atau komponen hidrolik.

### 9.3 Ilustrasi Sistem Kontrol

Pada bagian ini akan dijelaskan mengenai beberapa contoh ilustrasi sistem kontrol terbuka dan sistem kontrol tertutup.

#### 9.3.1 Sistem Kontrol Kecepatan Governor Watt

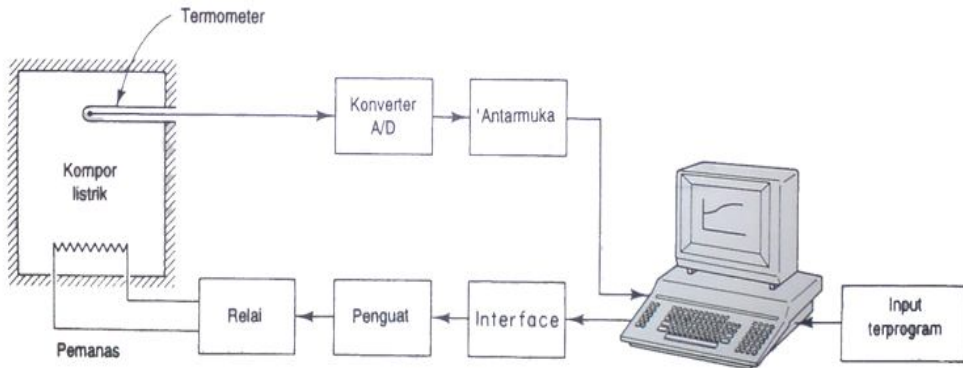
Prinsip dasar dari governor Watt untuk mesin dilukiskan dengan diagram skematik pada gambar 9.8. Besarnya laju aliran bahan bakar yang masuk ke silinder mesin diatur sesuai dengan selisih antara kecepatan mesin yang diinginkan dan kecepatan mesin yang sebenarnya. Kecepatan governor diatur sesuai dengan kecepatan yang diinginkan. Kecepatan yang sebenarnya turun di

bawah harga yang diinginkan, maka gaya sentrifugal *governor kecepatan* menjadi semakin mengecil, menyebabkan katup pengontrol bergerak ke bawah, mencatu bahan bakar yang lebih banyak sehingga kecepatan mesin membesar sampai dicapai harga yang diinginkan. Sebaliknya, jika kecepatan mesin melebihi nilai yang diinginkan, maka gaya sentrifugal dari *governor kecepatan* semakin membesar, maka menyebabkan katup pengontrol bergerak ke atas. Hal ini akan memperkecil catu bahan bakar sehingga kecepatan mesin mengecil sampai dicapai nilai yang diinginkan.

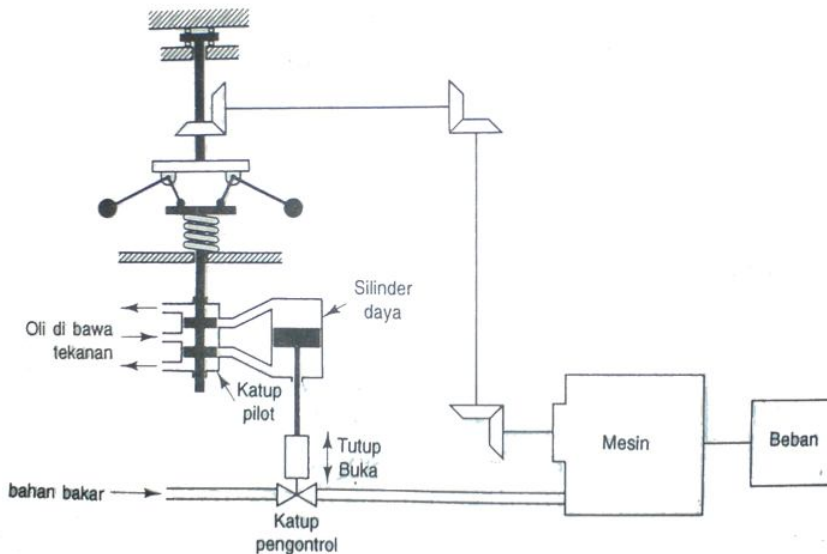
Pada sistem kontrol kecepatan ini, *plant* (sistem yang dikontrol) adalah mesin dan variabel yang dikontrol adalah kecepatan dari mesin tersebut. Perbedaan antara kecepatan yang dikehendaki dan kecepatan sebenarnya adalah sinyal, kesalahan. Sinyal kontrol (jumlah bahan bakar) yang akan diterapkan ke *plant* (mesin) adalah sinyal aktuasi. Masukan eksternal yang akan mengganggu variabel yang dikontrol adalah gangguan. Perubahan beban yang tidak diharapkan adalah gangguan.

#### 9.3.2 Sistem Kontrol Suhu

Gambar 9.9 menunjukkan diagram kontrol suhu dari kompor listrik. Suhu tersebut diukur oleh sensor suhu (komponen yang menghasilkan sinyal analog). Besaran tegangan suhu dalam bentuk sinyal analog dikonversi menjadi besaran digital oleh konverter A/D. Suhu digital tersebut dimasukkan ke kontroler melalui sebuah antarmuka. Suhu digital ini dibandingkan dengan suhu masukan yang diprogram, dan jika



Gambar 9.9 Sistem Kontrol Suhu



Gambar 9.8 Sistem Kontrol Kecepatan pada Mesin Governor Watt

terdapat penyimpangan (kesalahan), kontroler mengirim sinyal ke pemanas melalui sebuah antar muka penguat dan relai, untuk membawa suhu kompor ke nilai yang dikehendaki.

### 9.3.3 Sistem Kontrol Suhu Ruang Penumpang Mobil

Gambar 9.10 menunjukkan fungsi kontrol suhu dari ruang penum-

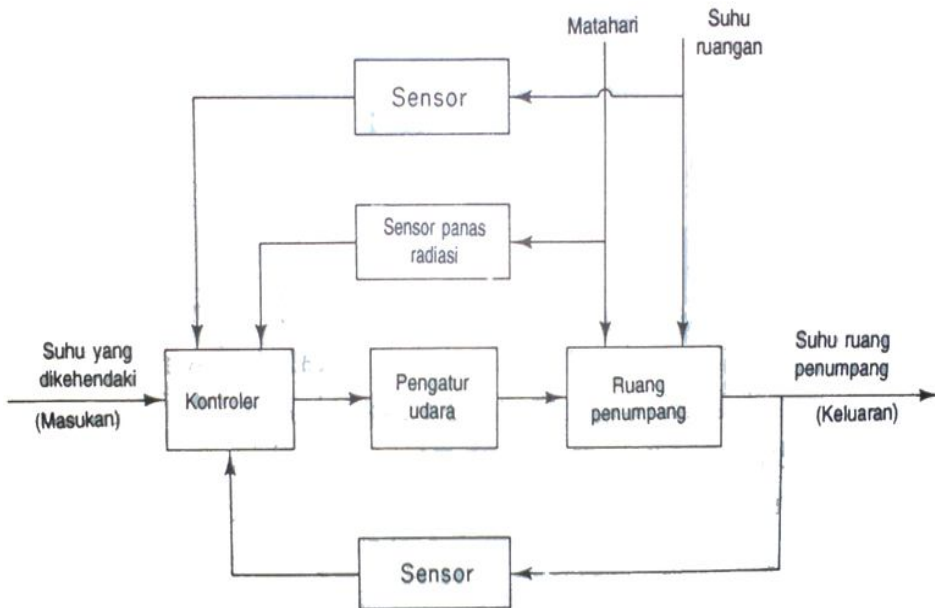
pang mobil. Suhu yang dikehendaki, dikonversi menjadi tegangan, adalah masukan ke kontroler. Suhu sesungguhnya dari ruang penumpang dikonversikan ke tegangan melalui sensor/transduser dan dimasukkan kembali ke kontroler untuk perbandingan dengan masukan. Suhu ruangan dan alih panas radiasi dari matahari, bertindak sebagai gangguan. Sistem ini menggunakan

baik kontrol umpan balik maupun kontrol umpan ke depan. (Kontrol umpan ke depan memberikan aksi koreksi sebelum gangguan mempengaruhi keluaran). Suhu ruang penumpang mobil berbeda cukup besar tergantung pada tempat di mana ia diukur. Daripada menggunakan banyak sensor untuk pengukuran

sehingga ruang penumpang sama dengan suhu yang dikehendaki.

### 9.3.4 Sistem Pengontrolan Lalu Lintas

Pengontrolan lalu-lintas dengan sinyal lalu-lintas yang dioperasikan pada basis waktu membentuk sebuah



Gambar 9.10 Sistem Kontrol Suhu di Ruang Penumpang Mobil

suhu dan meratakan nilai yang diukur, adalah lebih ekonomis memasang penghisap atau penghembus di tempat di mana penumpang biasanya merasakan suhu. Suhu udara dari penghisap atau penghembus adalah petunjuk suhu ruang penumpang (keluaran sistem). Kontroler menerima sinyal masukan, sinyal keluaran dan sinyal dari sensor sumber gangguan. Kontroler mengirimkan sinyal kontrol optimal ke alat pengatur udara (*air conditioner*) untuk mengontrol jumlah udara penyejuk sedemikian rupa

sistem kontrol loop terbuka. Meskipun demikian, jika jumlah mobil yang menunggu di setiap sinyal lalu-lintas pada suatu daerah yang ramai sekali, pada suatu kota, diukur secara kontinyu dan informasinya dikirim ke pusat komputer yang mengontrol sinyal-sinyal lalu lintas, maka sistem semacam itu menjadi loop tertutup.

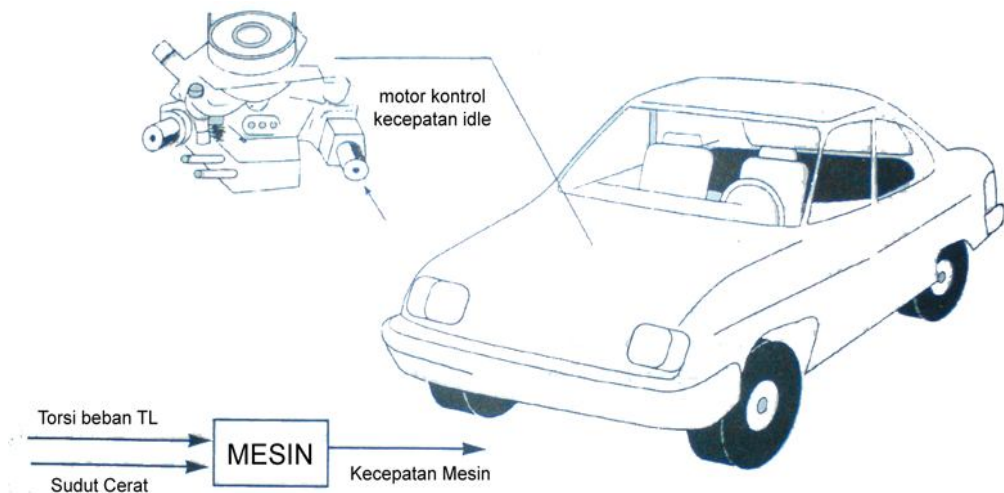
Pergerakan lalu lintas dalam jaringan adalah cukup kompleks karena variasi dari volume lalu-lintas sangat bergantung pada jam dan hari dalam satu minggu, maupun pada beberapa faktor yang lain. Dalam

beberapa hal, distribusi Poisson dapat diterapkan untuk kedatangan pada persimpangan, tetapi hal ini tidak perlu berlaku untuk semua persoalan lalu lintas. Pada kenyataannya, meminimalkan waktu tunggu rata-rata adalah suatu persoalan kontrol yang sangat kompleks.

"level" yang diinginkan, yang dipilih untuk memaksimalkan keuntungan.

### 9.3.6 Sistem Bisnis

Sistem bisnis bisa terdiri dari beberapa grup yang masing-masing mempunyai tugas (elemen dinamik sistem). Metode umpan-balik untuk



Gambar 9.11 Sistem Kontrol Kemudi dan Kecepatan Idle pada Mobil

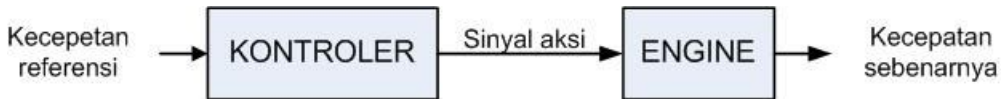
### 9.3.5 Sistem Kontrol Inventarisasi

Pemrograman laju produksi dan tingkat persediaan barang di industri merupakan contoh lain dari sebuah sistem kontrol loop tertutup. Tingkat persediaan yang sebenarnya, yang merupakan keluaran sistem, dibandingkan dengan tingkat persediaan yang diinginkan, yang dapat berubah dari waktu ke waktu sesuai dengan pasaran. Jika ada perbedaan antara tingkat persediaan yang sebenarnya dengan tingkat persediaan yang diinginkan, maka laju produksi distel sedemikian rupa sehingga keluaran selalu mendekati

melaporkan prestasi tiap grup harus ditetapkan dalam sistem tersebut, agar beroperasi dengan baik. Kopling silang antara grup-grup fungsional harus dibuat dalam orde minimum, untuk mengurangi waktu tunda yang tidak diinginkan dalam sistem. Semakin kecil kopling silang maka akan semakin halus aliran sinyal kerja dan bahan.

Sistem bisnis merupakan sistem loop tertutup. Disain yang bagus akan menyederhanakan kontrol manajerial yang diperlukan. Perhatikan bahwa gangguan pada sistem ini adalah cacat bahan atau manusia, interupsi komunikasi, kesalahan manusia, dan sejenisnya.





Gambar 9.12 Sistem Kecepatan Idle dengan Loop Terbuka

Penentuan perkiraan sistem yang baik didasarkan pada statistik dan kekuasaan manajemen yang baik. (Perhatikan bahwa hal ini dikenal dari kenyataan bahwa unjuk kerja sistem dapat ditingkatkan dengan pengaturan waktu atau antisipasi).

### 9.3.7 Sistem Kontrol Kemudi Mobil

Sebagai Suatu contoh sederhana dari sistem kontrol terbuka, bisa dilihat pada gambar 9.11, yaitu kontrol kemudi mobil. Arah dua roda depan dapat dianggap sebagai variabel yang dikendalikan atau keluaran ( $y$ ) arah dari roda kemudi adalah sinyal penggerak atau masukan ( $u$ ). Sistem kontrol kemudi mobil ini masih menggunakan sistem mekanis, karena memang unsur mekanis yang membentuk sistem kontrol ini.

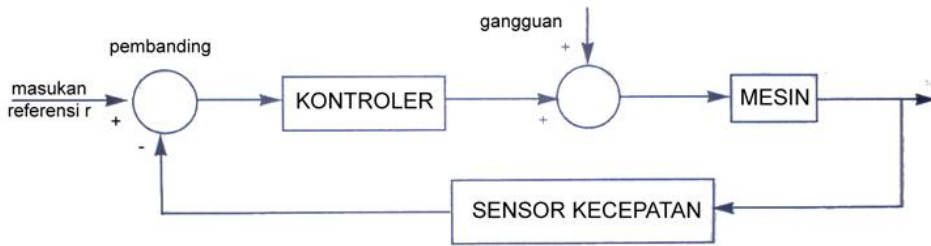
Sistem kontrol, atau proses pada masalah ini, terdiri dari mekanisme kemudi dan dinamika seluruh mobil. Walaupun demikian, jika tujuannya adalah untuk mengendalikan kecepatan mobil, maka besarnya tekanan yang dikerahkan pada pedal gas adalah sinyal penggerak, serta kecepatan kendaraan adalah variabel yang dikendalikan. Secara keseluruhan, kita dapat menyatakan bahwa sistem kontrol mobil yang sederhana merupakan satu kesatuan dengan dua masukan (kemudi dan pedal gas) dan dua keluaran satu tujuan dan kecepatan.

Dalam kasus ini, dua pengendalian dan dua keluaran tidak bergantung satu dengan yang lainnya, tetapi pada umumnya, terdapat sistem yang pengendaliannya saling berhubungan. Sistem dengan masukan dan keluaran lebih dari satu disebut sistem banyak variabel.

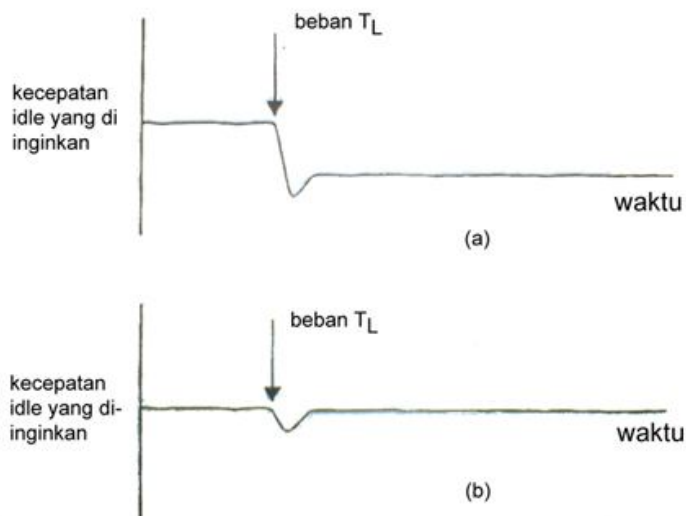
### 9.3.8 Sistem kontrol Kecepatan Idle mobil loop terbuka

Selain menggambarkan sistem kontrol kemudi, pada gambar 9.11 juga menggambarkan sistem kontrol kecepatan idle dari satu mesin mobil. Sistem kontrol kecepatan idle mobil bisa dirancang dengan menggunakan dua jenis pengontrolan, yaitu dengan loop terbuka atau loop tertutup.

Pada loop terbuka, tidak sulit untuk melihat bahwa sistem yang ditunjuk tersebut tidak akan memenuhi permintaan kinerja yang kritis. Misalnya, jika sudut katup  $\alpha$  ditentukan pada nilai awal tertentu, yang berhubungan dengan kecepatan tertentu, ketika suatu torsi beban  $T_L$  diberikan, tidak bisa dihindari suatu penurunan pada kecepatan mesin. Satu-satunya cara untuk membuat sistem tetap bisa bekerja adalah dengan menyesuaikan  $\alpha$  sebagai reaksi terhadap perubahan torsi beban yang berguna untuk mempertahankan kecepatan mesin  $\omega$  pada nilai yang diinginkan. Unsur sistem kontrol terbuka biasanya dibagi atas dua bagian yaitu



Gambar 9.13 Sistem Kecepatan Idle dengan Loop Tertutup



Gambar 9.14 Respon Sistem Kontrol Kecepatan Idle Loop Terbuka dan Tertutup

kontroler dan proses yang dikontrol, seperti yang ditunjukkan pada Gambar 9.12. Suatu sinyal masukan atau perintah  $r$  diberikan ke kontroler, dimana keluarannya bertindak sebagai sinyal penggerak  $u$ . Sinyal penggerak tersebut kemudian mengendalikan plant yang dikendalikan sehingga variabel yang dikendalikan  $y$  akan dihasilkan sesuai dengan persyaratan yang telah ditentukan. Dalam kasus sederhana, kontroler dapat berupa amplifier penguat), seperangkat alat mekanis

atau kontroler lainnya tergantung jenis sistem. Dalam kasus yang lebih canggih lagi, kontroler dapat berupa komputer seperti mikroprosesor. Karena kesederhanaan dan sifat ekonomis dari sistem kontrol loop terbuka, banyak ditemukan model sistem ini pada aplikasi yang tidak memerlukan ketelitian yang besar. Tujuan dari sistem ini adalah menghilangkan atau meminimumkan penurunan kecepatan ketika beban mesin digunakan.

### 9.3.9 Sistem kontrol Kecepatan Idle mobil loop tertutup

Sistem kontrol kecepatan idle dengan loop tertutup ditunjukkan pada gambar 9.13. Masukan referensi yang didefinisikan sebagai  $\omega_r$ , menentukan kecepatan idle yang diinginkan. Kecepatan mesin pada saat idle harus sesuai dengan nilai referensi  $\omega_r$ , setiap perubahan yang terjadi pada kecepatan mesin  $\omega$  jika torsi berubah, dideteksi oleh sensor kecepatan. Kontroler akan bekerja sesuai dengan perbedaan antara kecepatan referensi  $\omega_r$  dan kecepatan mesin  $\omega$  yang sebenarnya untuk menghasilkan suatu sinyal yang menyesuaikan sudut katup  $\alpha$  untuk mengurangi udara masuk, sehingga kecepatan mesin  $\omega$  akan sama dengan kecepatan referensi  $\omega_r$ .

Tujuan dari sistem kontrol ini adalah untuk mempertahankan kecepatan idle mesin pada suatu nilai yang relatif rendah (untuk penghematan bahan bakar) dengan mengabaikan beban mesin yang dipakai (seperti transmisi, kemudi servo, pengatur suhu, dan lain-lain). Tanpa kontrol kecepatan idle, setiap penggunaan beban mesin secara tiba-tiba akan menyebabkan suatu penurunan pada kecepatan mesin yang dapat menyebabkan mesin mati oleh karena itu tujuan utama dari sistem terkendali kecepatan idle dengan loop tertutup adalah untuk mempertahankan kecepatan idle mesin pada nilai yang diinginkan.

Gambar 9.14 membandingkan kinerja sistem kontrol kecepatan idle loop terbuka dan tertutup. Pada gambar 9.14 (a), kecepatan idle

sistem loop terbuka akan menurun dan berakhir pada nilai rendah setelah beban diberikan. Pada Gambar 9.14 (b), kecepatan idle sistem loop tertutup ditunjukkan untuk mengatasi nilai yang menurun setelah diberikan beban agar naik dengan cepat. Tujuan utama sistem kendali kecepatan idle yang telah diuraikan, dikenal sebagai *sistem regulator*, yang bertujuan mempertahankan keluaran sistem pada tingkat yang telah ditentukan.

## 9.4 Jenis Sistem Kontrol

Bagian ini membahas mengenai sistem kontrol mekanis, sistem kontrol pneumatik, sistem kontrol hidrolik dan sistem kontrol elektronik.

### 9.4.1 Sistem Kontrol Mekanis

Sistem kontrol mekanis merupakan suatu sistem kontrol yang menggunakan bahan-bahan mekanis sebagai kontrolernya. Hukum yang mendasari prinsip kerja kontroler secara mekanis adalah hukum kedua Newton, yaitu  $F = m \times a$ , dimana :

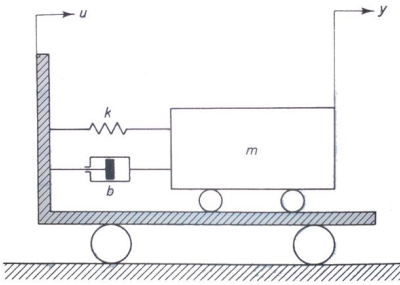
$F$  = gaya (N)

$m$  = massa (kg)

$a$  = percepatan ( $m/s^2$ )

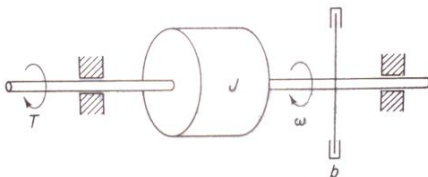
Contoh sistem mekanis adalah sistem translasi mekanika dan sistem rotasi mekanika. Tinjau sistem *dashpot massa pegas* yang dipasang pada kereta seperti yang ditunjukkan dalam Gambar 9.15. Dashpot adalah alat yang memberikan gesekan liat atau redaman. Ia terdiri dari sebuah torak dan silinder yang berisi minyak.

Gerakan relatif apapun antara besi torak dan silinder ditahan oleh minyak, karena minyak tersebut



Gambar 9.15 Sistem dashpot-massa-pegas yang dipasang di atas kereta

harus mengalir di sekitar torak (atau melalui lubang-lubang kecil yang terdapat pada torak) dari sisi yang satu ke sisi yang lain dari torak. Pada dasarnya dashpot menyerap energi. Energi yang diserap tersebut dikeluarkan sebagai panas dan dashpot tidak menyimpan energi kinetik ataupun tegangan. Dashpot dinamakan juga peredam (*damper*).



Gambar 9.16 Sistem Rotasi Mekanika

Tinjau sistem rotasi mekanika yang diunjukkan dalam Gambar 9.16. Sistem terdiri dari beban inersia dan peredam gesekan liat. Untuk sistem rotasi mekanika demikian, maka Hukum Newton kedua menyatakan

$$T = J\alpha$$

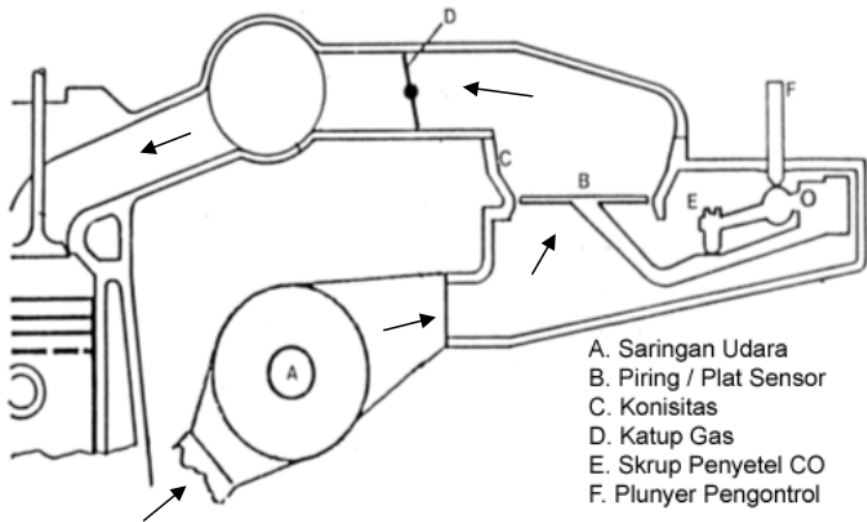
Dimana  $T$  = torsi yang diterapkan ke sistem (Nm)

$J$  = Momen Inersia dari beban ( $\text{kgm}^2$ )

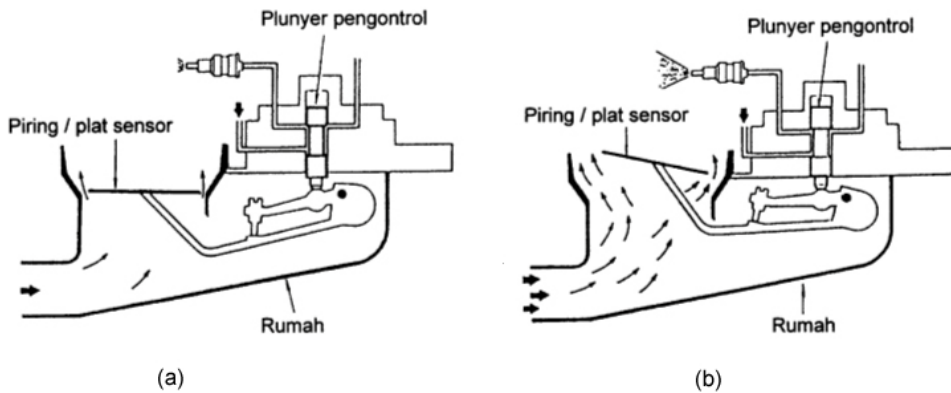
$\alpha$  = percepatan sudut dari beban ( $\text{rad/s}^2$ )

Contoh di kendaraan adalah pada sistem kontrol pengaliran bahan bakar yang menggunakan sistem karburator dan injeksi K. Pada bagian ini proses kerja karburator tidak perlu di bahas. Siswa dianggap sudah memahami bagaimana prinsip kerjanya. Sistem kontrol pengaliran bahan bakar yang menggunakan injeksi K ini bisa dijelaskan sebagai berikut. Sama dengan prinsip yang ada di karburator, pada sistem injeksi K pada kendaraan berbahan bakar bensin, bahan bakar dikabutkan secara terus-menerus. Yang membedakan adalah komponen yang digunakan. Pada sistem injeksi K, untuk menyemprotkan bahan bakar agar terbentuk kabut, digunakan injektor (komponen mekanis). Disini injektor menyemprot secara terus-menerus. Banyak sedikitnya bahan bakar yang disemprot, berdasarkan informasi yang diperoleh dari tekanan udara yang masuk. Semakin besar tekanan udara yang masuk, mengindikasikan bahwa massa udara yang terhisap di ruang bakar tentunya banyak. Agar didapatkan perbandingan yang ideal dari massa udara dan massa bahan bakar yang masuk ke ruang bakar, maka tentunya kontroler mekanis dari sistem K-Jetronik ini bisa mengatur berapa banyak bahan bakar yang disemprotkan melalui injektor. Dengan perbandingan yang ideal antara massa udara dan bahan bakar sebesar 14.7 : 1 akan didapatkan pembakaran yang sempurna. Hal ini yang menjadi masalah di semua sistem kontrol pengaliran bahan bakar, baik sistem karburator, K-Jetronik dan sistem kontrol pengaliran bahan bakar secara elektronik.

Pada gambar 9.17, dipelihatkan sistem pengaliran bahan bakar K-



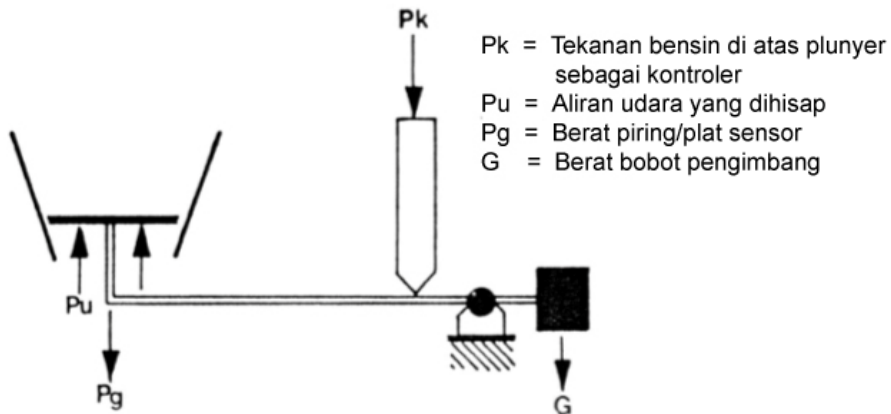
Gambar 9.17 Sistem Pengaliran bahan Bakar K-Jetronik



Gambar 9.18 Kontroler Mekanis pada Sistem K-Jetronik

Jetronik. Ketika pedal gas diinjak oleh pengemudi, maka katup gas D akan semakin terbuka. Sebaliknya jika pedal gas sedikit penginjakkannya oleh pengemudi maka katup gas sedikit pula terbukanya. Dengan semakin besar bukaan throttle maka udara yang masuk akan semakin besar pula. Udara masuk melalui sari-

ngan udara A, dengan arah aliran udara sesuai dengan arah anak panah. Setelah melalui saringan udara A, aliran udara menekan Piring/plat sensor B. Karena ada tekanan ini, maka plat sensor akan terangkat ke atas yang akhirnya menyebabkan plunyer pengontrol bahan bakar juga terangkat ke atas.



Gambar 9.19 Kontroler Mekanis pada Sistem K-Jetronic

Semakin besar tekanan udara yang masuk, maka semakin terangkat pula plat sensor yang akhirnya plunyer pengontrol juga semakin terangkat yang nantinya menyebabkan semakin banyak bahan bakar yang disemprotkan ke ruang bakar. Hal ini bisa diamati pada Gambar 9.18.

Ketika tekanan udara kecil (tampak dengan tanda anak panah yang sedikit pada Gambar 9.17 (b)), maka plunyer pengontrol terangkat sedikit. Sehingga aliran bahan bakar (bensin) yang menuju ke injektor terhambat dan sisanya dikembalikan lagi ke saluran menuju tangki bensin. Bahan bakar bensin dari tangki bensin mempunyai tekanan yang besar yang ditimbulkan oleh pompa di tangki bensin. Karena mempunyai tekanan yang besar tersebut, maka bahan bakar yang menuju injektor tadi menyemprot hingga mengabut. Hanya saja karena sedikit yang diteruskan ke injektor tadi, maka tentunya pengkabutan bensin tadi sedikit pula yang menuju ke ruang bakar.

Hal ini berbeda dengan kondisi seperti yang terlihat di Gambar 9.18

(b). Dengan semakin besar tekanan udara yang masuk (tampak gambar anak panah yang banyak), maka piring/plat sensor akan lebih terangkat ke atas. Hal ini mengakibatkan plunyer pengontrol semakin terangkat pula. Karena inilah, bahan bakar yang berada di saluran yang menuju injektor lebih banyak dari pada kondisi pada gambar 9.18 (b). Sehingga bahan bakar yang dikabutkan oleh injektor karena adanya tekanan bensin yang besar di saluran semakin banyak yang terhisap oleh mesin.

Selain dipengaruhi oleh tekanan udara, ada faktor lain yang mempengaruhi besar pengangkatan plunyer pengontrol. Hal ini bisa dijelaskan dengan melihat Gambar 9.19. Tampak bahwa selain tekanan udara ( $P_u$ ) ada juga faktor lain yang mempengaruhi seberapa besar plunyer pengontrol terangkat. Faktor-faktor tersebut adalah berat piring atau plat sensor ( $P_g$ ) dan berat bobot pengimbang ( $G$ ). Agar tercapai kesetimbangan maka  $P_u + G = P_g + P_k$ . Disinilah model matematik kontrolernya. Sehingga dari sini bisa

didapatkan besarnya keluaran kontroler (plunyer pengontrol), yaitu  $P_k = (P_u + G) - P_g$ . Ada dua kondisi yang bisa dijelaskan di sini, yaitu : (pada kondisi pedal gas)

- Ketika Katup gas lebih menutup  
Dimana  $P_u + G < P_g + P_k$ , maka piring/plat sensor lebih menutup saluran masuk.
- Ketika Katup gas lebih membuka  
 $P_u + G > P_g + P_k$ , maka plat sensor lebih membuka saluran masuk.

Faktor lain yang mempengaruhi aliran udara adalah bentuk konisitasnya (B pada Gambar 9.17). Dengan bentuk konisitas yang sedemikian rupa, maka aliran udara tersebut bisa terhambat atau mengalir lancar. Konisitas merupakan bentuk saluran. Sehingga dari penjelasan tersebut diatas dapat kita simpulkan sebagai berikut :

- Jumlah udara yang mengalir tergantung dari tinggi pengangkatan piring/plat sensor dan bentuk konisitasnya. Sesuai dengan pedal gas yang diinjak oleh seorang sopir.
- Jumlah bahan bakar yang diinjeksikan tergantung dari jumlah udara yang mengalir. Semakin besar udara yang mengalir, maka semakin besar bahan bakar yang diinjeksikan. Sebaliknya semakin kecil udara yang mengalir maka semakin sedikit bahan baka yang diinjeksikan. Dengan perhitungan yang sesuai, maka akan didapatkan perbandingan udara dan bahan bakar sebesar 14.7 : 1 di setiap kondisi bukaan katup gas. Dan hal ini yang diharapkan

pada sistem kontrol pengaliran bahan bakar K-Jetronik.

#### 9.4.2 Sistem Kontrol Pneumatik

Sebagai media yang paling berdaya guna untuk menyalurkan sinyal dan daya, fluida, baik dalam bentuk cairan ataupun gas, mempunyai banyak kegunaan dalam industri. Cairan dan gas pada dasarnya dapat dibedakan oleh relatif kemungkinan pemampatannya dan fakta bahwa cairan mungkin mempunyai permukaan yang bebas, sedang gas membesar memenuhi tempatnya. Dalam bidang rekayasa, istilah pneumatika menjelaskan sistem fluida yang menggunakan udara atau gas, dan hidrolika berlaku untuk sistem yang menggunakan minyak pelumas atau oli.

Sistem pneumatika digunakan secara ekstensif dalam otomatisasi mesin-mesin produksi dan dalam bidang kontroler otomatis. Misalnya, rangkaian pneumatika yang mengubah energi udara yang dimampatkan menjadi energi mekanika digunakan secara luas, dan berbagai jenis kontroler pneumatika ditemukan dalam industri. Karena sistem pneumatika dan sistem hidrolika sering saling dibandingkan, maka berikut ini kita akan memberikan perbandingan antara kedua sistem tersebut secara singkat. Perbedaan-perbedaan tersebut adalah sebagai berikut :

1. Udara dan gas dapat dimampatkan sedang oli tidak dapat dimampatkan.
2. Udara kekurangan sifat pelumas dan selalu mengandung uap air. Fungsi oli adalah sebagai fluida hidrolika dan juga pelumas.





3. Tekanan operasi normal sistem pneumatika jauh lebih rendah daripada sistem hidrolika.
4. Daya keluaran sistem pneumatika jauh lebih kecil daripada sistem hidrolika.
5. Ketepatan aktuator pneumatika adalah buruk pada kecepatan rendah, sedangkan ketepatan aktuator hidrolika dapat dibuat memuaskan pada semua kondisi kecepatan.
6. Pada sistem pneumatika kebocoran eksternal diperbolehkan sampai tingkat tertentu, tetapi kebocoran internal harus dihindarkan karena perbedaan tekanan efektif agak kecil. Pada sistem hidrolika, kebocoran internal diperbolehkan sampai tingkat tertentu, tetapi kebocoran eksternal harus dihindarkan.
7. Tidak diperlukan pipa kembali pada sistem pneumatika bila yang digunakan udara, sedang pipa kembali selalu diperlukan oleh sistem hidrolika.
8. Suhu operasi normal sistem pneumatika adalah 5° sampai 60°C (41° sampai 140°F). Namun sistem pneumatika dapat beroperasi pada suhu 0° sampai 200°C (32° sampai 392°F). Sistem pneumatika tidak peka terhadap perubahan suhu, tetapi sebaliknya dengan sistem hidrolika, dengan gesekan fluida disebabkan oleh kecepatan yang bergantung besar sekali pada suhu. Suhu operasi normal untuk sistem hidrolika adalah 20° sampai 70°C (68° sampai 158°F).
9. Sistem pneumatika tahan api dan ledakan, sedang sistem hidrolika tidak demikian.

### 9.4.3 Sistem Kontrol Hidrolik

Kecuali untuk kontroler pneumatik tekanan rendah, udara yang dimampatkan jarang digunakan untuk mengontrol kesinambungan gerakan alat-alat yang mempunyai massa.

Perbandingan antara sistem pneumatika dan sistem hidrolika. Fluida yang umumnya ditemukan dalam sistem pneumatika adalah udara. Dalam sistem hidrolika, fluidanya adalah oli atau minyak pelumas. Perbedaan sifat-sifat fluida terutama menjadi karakteristik perbedaan yang berarti di bawah beban eksternal. Untuk kasus demikian, kontroler hidrolika umumnya lebih dikehendaki.

Penggunaan yang meluas dari rangkaian hidrolika dalam aplikasi alat-alat bantu mesin, sistem kontrol pesawat terbang, dan operasi yang mirip dengan itu terjadi karena faktor-faktor seperti sifatnya yang positif, ketepatan, fleksibilitas, perbandingan daya kuda-berat yang tinggi, start yang cepat, berhenti dan ke belakang dengan lancar dan presisi, dan kesederhanaan operasinya.

Tekanan operasi dalam sistem hidrolika sekitar 145 dan 5000 lb/in<sup>2</sup> (antara 1 dan 35 MPa). Dalam beberapa aplikasi khusus, tekanan operasi mungkin sampai 10.000 lbf/in<sup>2</sup> (70 MPa). Untuk persyaratan daya yang sama, berat dan ukuran dari unit hidrolika dapat dibuat lebih kecil dengan meningkatkan tekanan pasokan. Pada sistem hidrolika tekanan tinggi, gaya yang sangat besar dapat diperoleh. Aksi yang cepat, peletakan posisi yang tepat dari beban yang berat dimungkinkan dengan sistem hidrolika. Kombinasi sistem elektronika dan hidrolika digunakan secara luas, karena ia

mengombinasikan kelebihan-kelebihan baik dari kontrol elektronika maupun daya hidrolika.

Terdapat kelebihan dan kekurangan tertentu dalam penggunaan sistem hidrolika dibandingkan dengan sistem lain. Beberapa kelebihan-kelebihannya adalah:

1. Fluida hidrolika bertindak sebagai pelumas, disamping membawa pergi panas yang dihasilkan dalam sistem ke tempat pertukaran panas yang baik (*convenient heat exchanger*).
2. Aktuator hidrolika yang secara perbandingan ukurannya kecil dapat mengembangkan gaya dan torsi yang besar.
3. Aktuator hidrolika mempunyai kecepatan tanggapan yang lebih tinggi dengan start, stop, dan kecepatan kebalikan yang cepat.
4. Aktuator hidrolika dapat dioperasikan di bawah keadaan berkesinambungan, terputus-putus (*intermittent*), kebalikan, dan melambat tanpa mengalami kerusakan.
5. Tersedianya aktuator balik linear maupun putar memberikan fleksibilitas dalam desain.
6. Karena kebocoran yang rendah dalam aktuator hidrolika, maka kecepatan akan jatuh bila beban yang diterapkan kecil.
3. Bahaya api dan ledakan ada, kecuali jika menggunakan fluida tahan api.
4. Karena sukar sekali merawat sistem hidrolika yang bebas dari kebocoran, maka sistem tersebut cenderung kotor.
5. Oli yang terkontaminasi mungkin menyebabkan kegagalan sistem hidrolika untuk fungsi dengan benar.
6. Sebagai hasil dari karakteristik non linear dan karakteristik rumit lainnya, maka desain dari sistem hidrolika yang canggih sangat memerlukan waktu dan usaha yang besar.
7. Rangkaian hidrolika umumnya mempunyai karakteristik redaman yang buruk. Jika rangkaian hidrolika tidak didesain dengan benar, maka beberapa fenomena yang tidak stabil mungkin terjadi atau hilang, tergantung pada keadaan operasi.

Di lain pihak, beberapa kekurangan cenderung membatasi penggunaannya :

1. Daya hidrolika tidak siap tersedia dibandingkan dengan daya listrik.
2. Biaya sistem hidrolika mungkin lebih tinggi daripada sistem listrik yang sebanding dan mengerjakan fungsi yang mirip.

#### 9.4.4 Sistem Kontrol Elektronik

Pada sistem kontrol elektronik, kontroler yang digunakan merupakan suatu unit yang terdiri dari komponen elektronika. Unit elektronika disini merupakan rangkaian yang terintegrasi dari banyak komponen elektronika, yaitu resistor, kapasitor, induktor, dioda, transistor, op-amp, IC dan masih banyak komponen elektronika yang lain. Unit elektronika tersebut, bisa berupa rangkaian yang sederhana maupun rangkaian yang kompleks. Salah satu komponen elektronika yang bisa dijadikan sebagai kontroler adalah potensiometer. Dengan komponen ini, sudah bisa mengolah sinyal tegangan, yaitu sebagai pelemah, tidak bisa digunakan untuk

menguatkan sinyal. Untuk menguatkan suatu sinyal, tentu saja harus digunakan komponen aktif, misalnya adalah op-amp. Apabila kita dapatkan selisih dari nilai referensi dan dari output plant (sinyal kesalahan/error) dan ternyata jenis kontroler yang diperlukan adalah pelemahan sinyal, maka dengan potensiometer tadi sudah bisa kita terapkan untuk membuat kontroler ini. Kontroler ini disebut *kontroler proposional*.

Lebih canggih lagi, komponen elektronik yang dijadikan sebagai kontroler adalah yang menggunakan mikroprosesor. Disini sudah digunakan teknologi digital. Beberapa tahun belakangan ini, teknologi digital sangat berkembang pesat. Baik yang tanpa menggunakan program atau yang memerlukan program. Mikroprosesor merupakan komponen elektronik yang memerlukan program agar bisa bekerja. Dengan program, maka bisa digunakan untuk berbagai aplikasi berdasarkan logika pemikiran dari seorang programmer dan perancang aplikasi tersebut.

Di dunia otomotif, ada suatu unit elektronik yang menggunakan mikroprosesor, berfungsi untuk mengatur jumlah bahan bakar dan mengatur waktu penyalaan pengapian. Unit ini disebut sebagai ECU (*Electronic Control Unit*). Ada banyak fungsi ECU yang lain di kendaraan. Ada yang digunakan untuk EPS (*Electronic Power Steering*), ABS (*Antilock Brake System*), Airbag System, AC (*Air Conditioning*), *Automatic Transmission* dan masih banyak sistem kontrol yang lain di kendaraan. Penjelasan mengenai sistem kontrol ini dijelaskan lebih detail pada bab-bab setelah ini.

## 9.5 Topologi Sistem Kontrol Elektronik

Pada sistem kontrol elektronik, ada beberapa komponen-komponen yang digunakan, yaitu sensor, pengkondisian sinyal, mikroprosesor dan mikrokontroler, memori, driver dan aktuator. Masing-masing komponen ini bisa dijelaskan di bawah ini.



Gambar 9.21 *Electronic Control Unit (ECU)*

### 9.5.1 Sensor

Sensor adalah piranti atau komponen yang digunakan untuk merubah suatu besaran non listrik (fisika maupun kimia) menjadi besaran listrik sehingga dapat dianalisa dengan rangkaian listrik tertentu. Ada beberapa istilah yang perlu diperhatikan, yaitu kesalahan (*error*), akurasi (*accuracy*), sensitivitas (*sensitivity*), repeabilitas (*repeability*), histerisis (*hysteresis*), linearitas (*linearity*).

Istilah kesalahan (*error*) didefinisikan sebagai perbedaan

antara nilai variabel yang sebenarnya dan nilai pengukuran variabel. Seringkali nilai sebenarnya tidak diketahui. Untuk kasus tertentu, akurasi akan menunjukkan range/bound kemungkinan dari nilai sebenarnya.

Istilah akurasi (*accuracy*) digunakan untuk menentukan kesaahan (*error*) keseluruhan maksimum yang diharapkan dari suatu alat dalam pengukuran. Ada beberapa jenis akurasi, yaitu :

1. Terhadap variabel yang diukur.

Misalnya akurasi dalam pengukuran suhu ialah  $2^{\circ}\text{C}$ , berarti ada ketidakakuratan (*uncertainty*) sebesar  $2^{\circ}\text{C}$  pada setiap nilai suhu yang diukur.

2. Terhadap prosentase dari pembacaan Full Scale suatu instrumen.

Misalnya akurasi sebesar 0.5% FS (*Full Scale*) pada meter dengan 5 V Full Scale, berarti ketidakakuratan pada sebesar 0.025 volt.

3. Terhadap prosentase span (*range kemampuan pengukuran instrumen*).

Misalnya jika sebuah alat mengukur 3% dari span untuk pengukuran tekanan dengan range 20 - 50 psi, maka akurasinya menjadi sebesar  $(0.03)(50 - 20) = 0.9$  psi.

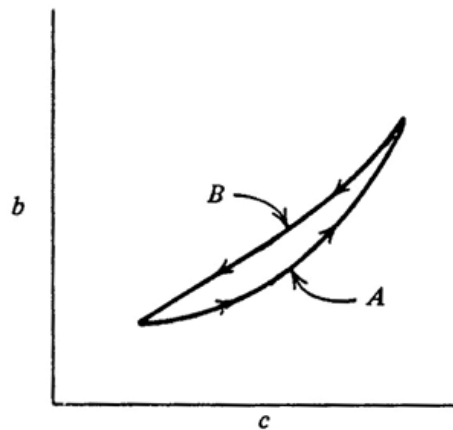
Istilah sensitivitas (*sensitivity*) didefinisikan sebagai perubahan pada output instrumen untuk setiap perubahan input terkecil. Sensitivitas yang tinggi sangat diinginkan karena jika perubahan output yang besar terjadi saat dikenai input yang kecil, maka pengukuran akan semakin mudah dilakukan. Misalnya, jika sensitivitas sensor temperatur

sebesar  $5 \text{ mV}/^{\circ}\text{C}$  berarti setiap perubahan input  $1^{\circ}\text{C}$  akan muncul output sebesar 5 mV.

Istilah repeabilitas (*repeability*) didefinisikan sebagai pengukuran terhadap seberapa baik output yang dihasilkan ketika diberikan input yang sama beberapa kali.

$$\text{repeatability} = \frac{\text{max} - \text{min}}{\text{fullscale}} \times 100\%$$

Istilah histerisis (*hysterisis*) didefinisikan sebagai perbedaan output yang terjadi antara pemberian input menaik dan pemberian input menurun dengan besar nilai input sama. Merupakan salah satu indikator repeabilitas.



Gambar 9.22 Grafik Histerisis

Istilah linearitas (*linearity*) didefinisikan sebagai hubungan antara output dan input dapat diwujudkan dalam persamaan garis lurus. Linearitas sangat diinginkan karena segala perhitungan dapat dilakukan dengan mudah jika sensor dapat diwujudkan dalam persamaan garis lurus.

Dalam pemilihan dan penggunaan suatu sensor, diperlukan pertimbangan-pertimbangan, agar sesuai dengan yang diharapkan. Pertimbangan-pertimbangan tersebut meliputi :

1. *Identifikasi sinyal yang sebenarnya.*

Tahap ini meliputi nilai nominal dan range pengukuran sensor, kondisi fisik lingkungan dimana pengukuran dilakukan, kecepatan pengukuran yang diperlukan, dan lain-lain.

2. *Identifikasi sinyal output yang dibutuhkan.*

Kebanyakan output yang dihasilkan sebesar arus standar 4 – 20 mA (contoh pada sensor temperatur) atau tegangan yang besarnya diskalakan untuk mewakili range pengukuran sensor. Mungkin ada kebutuhan lain seperti isolasi impedansi output, dan lain-lain. Dalam beberapa kasus mungkin diperlukan konversi secara digital pada output.

3. *Memilih sensor yang tepat.*

Berdasar langkah pertama, kita pilih sensor yang sesuai dengan spesifikasi range dan lingkungan. Selanjutnya, harga dan ketersediaan sensor juga harus dipertimbangkan.

4. *Mendesain pengkondisi sinyal yang sesuai.*

Dengan pengkondisi sinyal, output dari sensor akan diubah menjadi bentuk sinyal output yang kita perlukan.

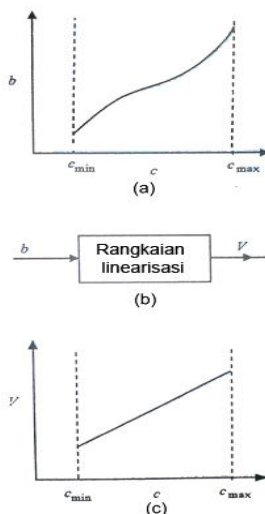
Sensor dibedakan menjadi 2 jenis, yaitu sensor fisika dan sensor kimia. Sensor fisika mendeteksi besaran suatu besaran berdasarkan hukum-hukum fisika. Contoh sensor

fisika adalah sensor cahaya, sensor suara, sensor kimia, sensor gaya, sensor kecepatan, dan sensor percepatan, dan sensor suhu. Sedangkan Sensor kimia mendeteksi jumlah suatu zat kimia dengan cara mengubah besaran kimia menjadi besaran listrik. Biasanya melibatkan beberapa reaksi kimia. Contoh sensor kimia adalah sensor pH, sensor Oksigen, sensor ledakan, dan sensor gas. Ada penggolongan lain berdasarkan keperluan dari sumber energi, yaitu sensor pasif dan sensor aktif. Untuk mengkonversi sifat-sifat fisik atau kimia ke besaran listrik sensor pasif tidak memerlukan bantuan sumber energi, contohnya adalah termocouple. Termocouple menghasilkan tegangan output sebanding dengan suhu pada sambungan termcouple tersebut. Berbeda dengan sensor aktif, untuk mengkonversi sifat-sifat fisik atau kimia ke besaran listrik sensor aktif ini memerlukan bantuan sumber energi. Ada 6 tipe isyarat penggolongan sensor, yaitu :

1. *Mechanical*, contoh : panjang, luas, mass flow, gaya, torque, tekanan, kecepatan, percepatan, panjang gel acoustic dan lain-lain.
2. *Thermal*, contoh : temperature, panas, entropy, heat flow dan lain-lain.
3. *Electrical*, contoh : tegangan, arus, muatan, resistance, frekuensi dan lain-lain.
4. *Magnetic*, contoh : intensitas medan, flux density dan lain-lain.
5. *Radiant*, contoh : intensitas, panjang gelombang, polarisasi dan lain-lain.
6. *Chemical*, contoh : komposisi, konsentrasi, pH, kecepatan reaksi dan lain-lain.

## 9.5.2 Pengkodisian Sinyal (*Signal Conditioning*)

Pengondisi sinyal merupakan suatu operasi elektronik untuk mengkonversi sinyal tersebut menjadi sinyal yang sesuai dengan komponen elektronik lain yang diperlukan di dalam sistem kontrol. Pengkondisian sinyal dibagi menjadi dua bagian, yaitu pengkondisi sinyal secara analog dan secara digital. Pengkondisian secara analog menghasilkan sinyal keluaran yang masih merepresentasikan sinyal analog yang variabel. Pada aplikasi pemrosesan digital, beberapa pengkondisi sinyal analog tertentu dilakukan sebelum konversi analog ke digital dikerjakan.



Gambar 9.23 Linearisasi pada Sinyal yang Tidak Linear

### 9.5.2.1 Pengkondisian Sinyal Analog (*Analog Signal Conditioning*)

Sebuah sensor menghasilkan nilai variabel dalam besaran listrik

setelah melewati proses konversi. Tentunya besar sinyal ini bergantung terhadap karakteristik materialnya. Agar sinyal yang dihasilkan oleh sebuah sensor sesuai dengan yang diinginkan maka kita harus mengkonversinya setelah didapatkan keluarannya. Kita tidak bisa merubah karakteristik material didalamnya, karena tentunya sensor tersebut sudah menjadi satu kesatuan yang terintegrasi. Hanya industri pembuat sensor tersebut yang mampu merubahnya, karena kita hanya sebagai pemakai sensor tersebut dan bukan kita sendiri yang membuatnya.

Sehingga hanya ada pilihan yang sedikit untuk kita terapkan ke sistem kontrol nantinya. Sebagai contoh adalah *cadmium sulfida* mempunyai nilai resistansi yang bervariasi yang berkebalikan dan tidak linear berdasarkan intensitas cahaya. Pengkondisi sinyal secara analog diperlukan dalam kasus ini untuk merubah sinyal yang dihasilkan tersebut untuk dihubungkan dengan komponen lain dalam sistem kontrol. Tentunya konversi ini dilakukan secara elektrik. Kita sering menguraikan bahwa akibat dari pengkondisian sinyal membentuk suatu transfer fungsi tertentu. Dengan rangkaian penguat tegangan yang sederhana, ketika diberi masukan tegangan pada rangkaian tersebut, maka memberikan tegangan keluaran. Hal ini memungkinkan membagi rangkaian pengkondisi sinyal secara umum sebagai berikut :

a. Merubah level sinyal

Metode yang sederhana pada rangkaian pengkondisi sinyal adalah merubah level atau nilai dari sinyal tersebut. Contoh yang sering dipakai adalah penguatan (*amplifier*) dan pelemahan (*attenuate*) level

tegangan. Secara umum, aplikasi sistem kontrol dengan sinyal dc atau frekuensi rendah dapat dikuatkan dengan mudah. Faktor penting untuk memilih rangkaian penguatan adalah impedansi input dari keluaran sensor.

Dalam sistem kontrol, sinyal selalu menggambarkan variabel proses (atau keluaran sistem) yang nantinya akan dibandingkan dengan nilai variabel (nilai referensi) untuk diolah oleh kontroler. Dalam beberapa kasus respon frekuensi dalam rangkaian penguatan sangat penting untuk diperhatikan, contohnya pada sensor *accelerometer* dan *optical detector*.

#### b. Linearisasi

Sesuai dengan penjelasan sebelumnya, bahwa pembuat sistem kontrol mempunyai pilihan yang sedikit dari karakteristik keluaran sensor terhadap variabel proses. Seringkali hubungan antara masukan dan keluaran dari sensor adalah tidak linear. Bahkan sensor yang mendekati linear pun juga bisa menjadi masalah ketika pengukuran yang presisi dari variabel sinyal diperlukan.

Menurut sejarah, rangkaian analog dikhususkan pada penggunaan sinyal yang linear. Sebagai contoh, diperkirakan keluaran dari sebuah sensor bervariasi dan tidak linear dengan variabel proses. Ditunjukkan pada Gambar 9.23 (a). Rangkaian linearisasi di buat blok diagram ditunjukkan pada Gambar 9.23 (b), kondisi yang ideal, yaitu hubungan yang linear dari keluaran sensor yang berupa tegangan dan variabel proses didapatkan, seperti terlihat pada Gambar 9.23 (c). Rangkaian seperti

itu sulit untuk mendesainnya dan biasanya operasi daerah kerjanya dibatasi.

Pendekatan modern untuk masalah ini adalah menjadikan sinyal yang tidak linear tersebut sebagai masukan dari sebuah komputer dan membentuk linearisasi dengan menggunakan software. Secara virtual, banyak ketidaklinearan dapat diatasi dengan cara ini dengan komputer modern yang cepat pemrosesannya secara *real time*.

#### c. Konversi

Seringkali pengkondisi sinyal digunakan untuk mengkonversi dari besaran listrik yang satu ke besaran listrik yang lain. Sebagian besar dari kelompok sensor/transduser, memperlihatkan perlunya merubah resistensinya dengan variabel yang dinamis. Dalam kasus ini, perlu disediakan rangkaian untuk mengkonversi resistansi tersebut menjadi sinyal tegangan (Volt) atau sinyal arus (Ampere). Hal ini biasanya bisa terpenuhi oleh rangkaian jembatan saat perubahan resistansinya kecil dan/atau dengan rangkaian penguat (*amplifier*) dengan variasi penguatannya.

Tipe penting dari suatu pengkonversian dihubungkan dengan kontrol proses yang standar dari sinyal yang ditransmisikan berupa level arus sebesar 4-20 mA pada kabel. Hal ini memerlukan pengkonversian resistansi dan level tegangan menjadi level arus yang diperlukan pada akhir pengiriman sinyal dan untuk pengkonversian balik dari arus menjadi tegangan pada akhir penerimaan sinyal yang dikirim. Tentunya pengiriman sinyal (*signal transmission*) dengan arus dipakai karena sinyal tidak

bergantung dengan beban yang bervariasi. Dengan begitu, maka diperlukan perubahan tegangan ke arus dan perubahan arus ke tegangan.

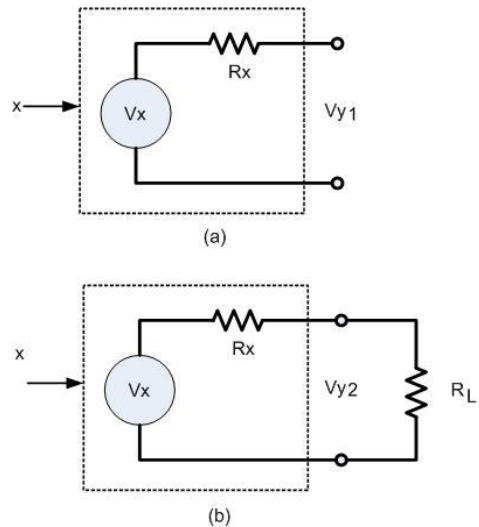
Pemakaian komputer atau mikrokomputer dalam sistem kontrol memerlukan pengkonversian data analog menjadi data digital (digital interfacing) oleh rangkaian yang terintegrasi. Rangkaian ini disebut *Analog to Digital Converter (ADC)*. Konversi sinyal analog biasanya diperlukan untuk mengatur sinyal analog yang diukur agar sesuai menjadi sinyal digital yang diperlukan sebagai masukan ADC. Sebagai contoh, ADC memerlukan sinyal masukan yang bervariasi antara 0 sampai dengan 5 Volt, tetapi sensor memberikan sinyal yang bervariasi antara 30 sampai dengan 80 mV. Rangkaian pengkonversi sinyal tersebut dapat dibuat untuk menghubungkan keluaran sensor tersebut ke masukan ADC yang diperlukan.

#### d. Filter dan Penyesuaian Impedansi

Ada dua pengkondisi sinyal bersama lainnya yang diperlukan, yaitu proses pemfilteran (*filtering*) dan penyesuaian impedansi (*matching impedance*).

Seringkali sinyal informasi yang sering dijumpai di dunia industri sekarang ini mempunyai frekuensi 60 Hz. Motor listrik sewaktu di start, menyebabkan sinyal *pulse* dan sinyal lain yang tidak diinginkan dalam sistem kontrol tertentu. Pada banyak kasus, hal ini memerlukan pemakaian filter high-pass, *filter* low-pass atau filter notch untuk mengurangi atau menghilangkan sinyal yang tidak diinginkan tersebut. Contoh proses filter yang dapat dipenuhi oleh filter pasif adalah hanya dengan memakai

resistor, kapasitor, dan induktor, atau filter aktif dengan memakai penguatan dan balikan (*feedback*).



Gambar 9.24 Konsep Pembebanan

Penyesuaian impedansi adalah elemen yang penting dalam pengkondisian sinyal ketika impedansi internal dari sensor atau impedansi saluran transmisi dapat menyebabkan kesalahan (*error*) dalam pengukuran variabel dinamis. Rangkaian yang menggunakan komponen aktif dan pasif digunakan untuk mengadakan penyesuaian impedansi tersebut.

#### e. Konsep Pembebanan

Salah satu yang menjadi perhatian utama dalam pengkondisian sinyal analog adalah pembebanan satu rangkaian oleh rangkaian lainnya. Disini dikenalkan adanya ketidakpastian amplitudo dari suatu sinyal tegangan. Jika tegangan ini merepresentasikan beberapa variabel proses, maka ada ketidakpastian dalam nilai variabel tersebut.



Pembebanan dapat dijelaskan sebagai berikut. Sebagai misal keluaran dari rangkaian terbuka dari beberapa komponen elektronika menghasilkan suatu tegangan  $V_{y1} = V_x$ , sesuai gambar 9.24 (a). Rangkaian terbuka berarti tidak terhubung dengan rangkaian yang lain. Pembebanan terjadi ketika kita menghubungkannya dengan sebuah beban atau rangkaian terintegrasi yang ditambahkan ke keluaran tadi (lihat Gambar 9.24 (b)) dan tegangan keluaran tadi menjadi turun beberapa volt jika dibandingkan dengan rangkaian yang terbuka sebelumnya, dimana  $V_{y2} < V_{y1}$ . Pembebanan yang berbeda akan menghasilkan pengurangan (*drop*) tegangan yang berbeda pula. Nilai  $V_{y1}$  jika diukur dengan voltmeter akan menunjukkan sebesar  $V_{y1} = V_x$ . Berbeda dengan dengan sewaktu kita beri beban sesuai gambar 9.24 (b), maka nilai  $V_{y2}$  yang ditunjukkan oleh voltmeter

sebesar  $V_{y2} = V_x \left( \frac{R_L}{R_L + R_x} \right)$ , atau

sebesar  $V_{y2} = V_x \left( 1 - \frac{R_x}{R_L + R_x} \right)$ .

Jika besaran listrik berupa sinyal yang berfrekuensi atau sinyal digital, maka pembebanan bukan merupakan suatu masalah. Dalam hal ini, sinyal setelah ada pembebanan tidak akan terjadi error dalam hal besaran frekuensinya. Pembebanan sangat penting ketika besaran yang dipakai adalah amplitudonya.

Ada dua jenis rangkaian pengkondisi sinyal, yaitu rangkaian pasif dan rangkaian aktif. Ada beberapa contoh rangkaian pasif, yaitu rangkaian pembagi (*divider*

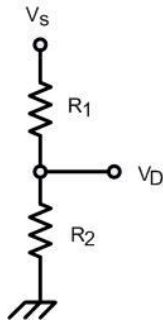
*circuits*), rangkaian jembatan (*bridge circuits*), filter RC (*RC filter*) dan lain-lain. Sedangkan contoh rangkaian aktif yaitu rangkaian dengan menggunakan komponen op-amp. Dalam hal ini tidak perlu dibahas pada bagian ini, karena rangkaian op-amp sudah dijelaskan pada bab 8.

Rangkaian jembatan dan pembagi merupakan dua teknik rangkaian pasif yang telah digunakan untuk pengkondisi sinyal sudah lama sekali. Meskipun rangkaian aktif yang modern menggantikan teknik ini, masih banyak aplikasi yang menggunakan teknik ini dengan keuntungannya.

Rangkaian jembatan secara khusus dipakai untuk mendapatkan akurasi tinggi dalam pengukuran impedansinya. Ada rangkaian yang mempunyai perubahan impedansi yang sangat kecil, maka disinilah diperlukan rangkaian jembatan ini.

Tipe rangkaian pasif lain yang dilibatkan dalam pengkondisian sinyal adalah memfilter frekuensi yang tidak diinginkan dari sinyal yang terukur. Di dalam praktek industri atau di bidang elektronika yang lain, ditemukan sinyal dengan *noise* (sinyal yang tidak diinginkan) yang mempunyai frekuensi rendah atau frekuensi tinggi, padahal sinyal yang seperti ini tidak diharapkan untuk muncul. Sebagai contoh adalah sensor untuk mengkonversi temperatur menghasilkan sinyal tegangan dc, proposional terhadap temperatur. Karena sumber power yang digunakan di lingkungan sekitar menggunakan sinyal ac 60 Hz (tegangan listrik PLN), ada kemungkinan sinyal 60 Hz tersebut mempengaruhi keluaran tegangan sensor yang tentunya ada perbedaan dengan temperatur yang seharusnya proposional tadi. Rangkaian pasif yang terdiri dari resistor dan kapasitor

seringkali dipakai untuk mengeliminir *noise* yang mempunyai frekuensi tinggi dan rendah tanpa ada perubahan sinyal yang seharusnya.



Gambar 9.25 Rangkaian Pembagi Tegangan yang Sederhana

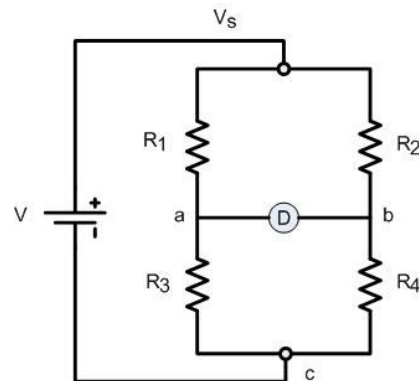
Rangkaian pembagi tegangan (*divider circuits*) dasar diperlihatkan pada gambar 9.25 seringkali digunakan untuk mengkonversi (merubah) nilai resistansi yang bervariasi menjadi tegangan yang bervariasi pula. Hubungan tegangan keluaran dari rangkaian pembagi  $V_D$ , resistor ( $R_1, R_2$ ) dan tegangan sumber ( $V_s$ ) adalah

$$V_D = \frac{R_2 V_s}{R_1 + R_2}$$

dimana  $V_s$  = tegangan sumber  
 $R_1, R_2$  = resistor pembagi

Rangkaian jembatan (*bridge circuit*) digunakan untuk mengkonversi impedansi yang bervariasi menjadi tegangan yang bervariasi pula. Salah satu keuntungan menggunakan rangkaian jembatan ini adalah dapat didesain

untuk menghasilkan sinyal tegangan yang bervariasi terhadap *ground* (tegangan yang bernilai nol). Jika pada mobil maka badan mobil atau minus baterai yang menjadi *ground*. Ini berarti bahwa penguatan dapat dipakai untuk menambah level tegangan untuk penambahan sensitivitas pada impedansi yang bervariasi. Gambar 9.26 menunjukkan rangkaian jembatan yang disebut jembatan *wheatstone*. Rangkaian ini dipakai untuk aplikasi pengkondisi sinyal dimana sebuah sensor yang merubah resistansi menjadi tegangan sebagai variabel proses, menjadi masukan ke kontroler yang sebelumnya dibandingkan dengan referensi (lihat blok diagram sistem kontrol tertutup pada gambar 9.3).



Gambar 9.26 Rangkaian Jembatan Wheatstone DC Dasar

Banyak modifikasi yang dilakukan pada rangkaian jembatan dasar ini untuk aplikasi lain yang lebih spesifik. Pada Gambar 9.26, obyek dengan label D adalah sebuah detektor tegangan (*voltage detector*) dipakai untuk membandingkan tegangan (*potensial*) di titik antara a

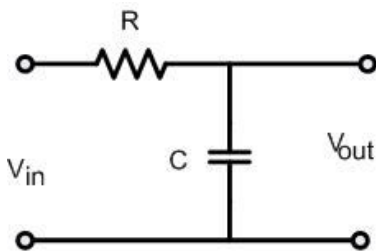
dan pada rangkaian. Dalam aplikasi yang modern, detektor merupakan amplifier diferensial dengan impedansi input yang tinggi. Dalam kasus ini beda potensial ( $\Delta V$ ) antara titik a dan b dirumuskan menjadi

$$\Delta V = V_a - V_b$$

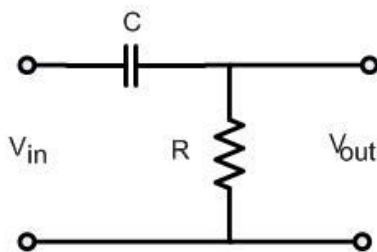
dimana :

$V_a$  = Tegangan titik a terhadap titik c (ground atau tegangan referensi)

$V_b$  = Tegangan titik b terhadap titik c (ground atau tegangan referensi)



(a) Filter Lowpass



(b) Filter Highpass

Gambar 9.27 Rangkaian Filter pasif

Nilai  $V_a$  dan  $V_b$  sekarang dapat ditentukan, dimana  $V_a$  merupakan

tegangan sumber  $V$  yang dibagi oleh  $R_1$  dan  $R_3$

$$V_a = \frac{VR_3}{R_1 + R_3}$$

dengan cara yang sama maka  $V_b$ , pembagi tegangan diberikan oleh

$$V_b = \frac{VR_4}{R_2 + R_4}$$

dimana

$V$  = sumber tegangan rangkaian

Jika persamaan di atas dikombinasikan, maka beda tegangan dapat ditulis menjadi

$$\Delta V = \frac{VR_3}{R_1 + R_3} - \frac{VR_4}{R_2 + R_4} \quad \text{dengan}$$

memakai beberapa persamaan aljabra matematika, dapat ditunjukkan persamaan tersebut menjadi

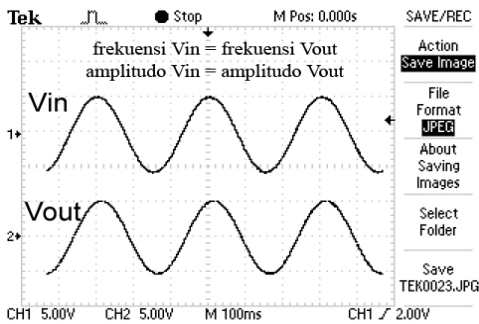
$$\Delta V = V \frac{R_3R_2 - R_1R_4}{(R_1 + R_3).(R_2 + R_4)}$$

Persamaan ini menunjukkan bagaimana beda potensial dari detektor adalah sebuah fungsi dari sumber tegangan dan nilai-nilai dari resistor. Jika hasil yang didapatkan dari nilai beda potensial  $\Delta V$  adalah nol, maka akan kita dapatkan bahwa

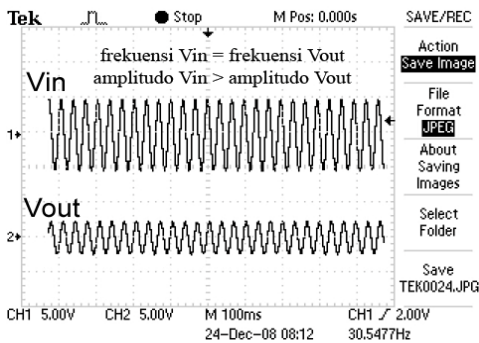
$$R_3R_2 = R_1R_4$$

Untuk mengeleminasi sinyal *noise* yang tidak diinginkan dari pengukuran, seringkali diperlukan

pemakaian rangkaian untuk meneruskan atau menghilangkan sinyal dalam daerah frekuensi tertentu. Rangkaian ini disebut filter. Filter yang sederhana dapat dibuat dari sebuah resistor dan sebuah kapasitor. Rangkaian ini bisa membentuk rangkaian filter *Lowpass* dan filter *Highpass*. Filter *Lowpass*



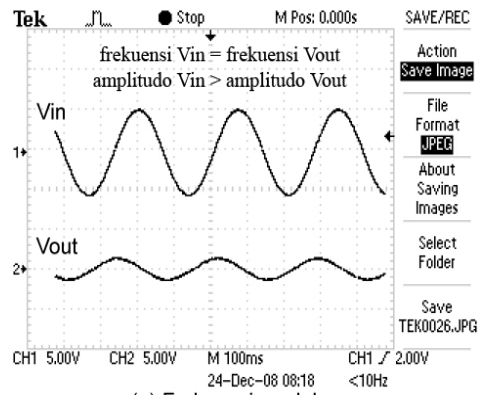
(a) frekuensi rendah



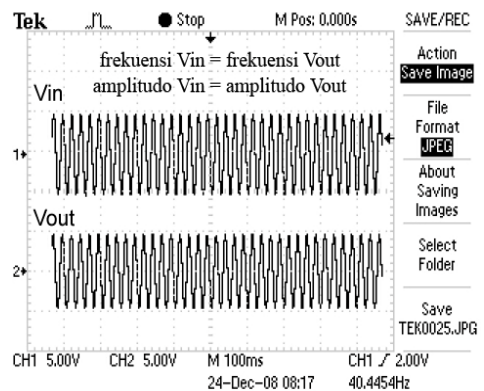
(b) frekuensi tinggi

Gambar 9.28 Hasil Eksperimen dari Rangkaian Filter Pasif *Lowpass*

dapat menghilangkan frekuensi tinggi dan meneruskan sinyal frekuensi rendah. Dalam hal ini adalah nilai amplitudonya yang nilainya dibuat tetap pada frekuensi tertentu atau dibuat berkurang hingga menjadi nol pada frekuensi tertentu. Dengan filter *Lowpass*, jika ada sinyal yang dengan



(a) Frekuensi rendah



(b) Frekuensi tinggi

Gambar 9.29 Hasil Eksperimen dari Rangkaian Filter Pasif *Highpass*

frekuensi yang semakin tinggi, maka amplitudo sinyal tersebut akan semakin berkurang. Rangkaian ini bisa dibuat seperti yang terlihat di Gambar 9.27 (a). Hasil eksperimen dengan menggunakan alat ukur osiloskop ditunjukkan pada Gambar 9.28.

Dengan memberikan nilai frekuensi yang berbeda antara sinyal  $V_{in}$  pada Gambar 9.28 (a) dan  $V_{in}$ , Gambar 9.28 (b) maka menghasilkan sinyal  $V_{out}$  yang berbeda pula dalam hal amplitudo. Tetapi nilai frekuensinya tidak berubah. Dan ada

pergeseran phase antara  $V_{in}$  dan  $V_{out}$ , hanya saja masalah pergeseran phase tidak di bahas dalam buku ini.

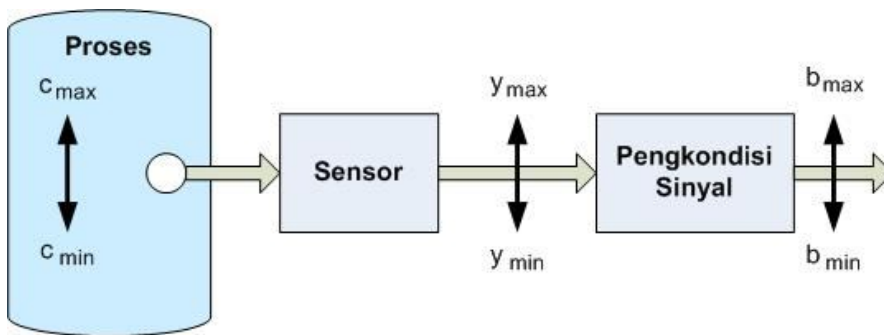
Sebaliknya pada filter *Highpass*, filter ini mampu menghilangkan sinyal dengan frekuensi rendah dan meneruskan frekuensi tinggi. Tentunya nilai amplitudo yang menjadi kecil atautkah tetap. Rangkaian RC untuk filter ini ditunjukkan pada Gambar 9.27 (b).

Hasil eksperimen dengan menggunakan alat ukur osiloskop ditunjukkan pada Gambar 9.29. Dengan memberikan nilai frekuensi yang berbeda antara sinyal  $V_{in}$  pada Gambar 9.29 (a) dan  $V_{in}$ , Gambar 9.29 (b) maka menghasilkan sinyal  $V_{out}$  yang berbeda pula dalam hal amplitudo. Tetapi nilai frekuensinya tidak berubah. Dan ada pergeseran phase antara  $V_{in}$  dan  $V_{out}$ , hanya saja masalah pergeseran phase tidak di bahas dalam buku ini.

bagaimana petunjuk ini dipakai untuk mengembangkan suatu desain. Petunjuk di bawah untuk memastikan bahwa suatu masalah bisa dipahai dengan benar dan disini dibahas mengenai hal-hal yang penting.

Tidak semua petunjuk akan menjadi sesuatu yang penting dalam setiap pendesainan, bisa saja beberapa tidak sesuai dengan aplikasi yang kita buat. Pada banyak kasus, tidak cukup informasi yang kurang dalam menunjukkan suatu masalah dengan baik, maka seorang desainer harus mempunyai kemampuan teknik yang baik dan terlatih dalam setiap bagian desain.

Gambar 9.30 menunjukkan model pengukuran dan pengkondisi sinyal. Dalam beberapa hal pada keseluruhan sistem dikembangkan, dari pemilihan sensor sampai mendesain pengkondisi sinyal. Dalam hal yang lain, hanya pengkondisi sinyal yang akan dikembangkan. Petunjuk dibawah dibuat secara umum. Karena sensor dipilih dari



Gambar 9.30 Model dari Pengukuran dan Pengkondisi Sinyal

Pada bagian ini dibahas mengenai petunjuk pendesainan, sesuatu hal yang harus dipertimbangkan sewaktu mendesain sebuah sistem pengkondisian sinyal analog. Contoh disini menunjukkan

yang tersedia, desain dibuat secara aktual dan benar-benar untuk pengkondisi sinyal yang sesuai. Petunjuk untuk mendesain pengkondisi sinyal analog adalah sebagai berikut :

1. *Definisikan apa yang menjadi tujuan pengukuran.*
  - a. *Parameter:* Apa yang menjadi sifat alami dari variabel yang diukur? tekanan, temperatur, aliran, level, tegangan, arus, resistansi atau lainnya? *Daerah kerja (range):* Berapa daerah kerja dari pengukuran? 100 s/d 200 °C, 45 s/d 85 psi, 2 s/d 4 Volt atau lainnya?
  - b. *Akurasi (accuracy):* Berapa akurasi yang diperlukan ? 5% *Full Scale (FS)*, 3% atau lainnya ?
  - c. *Linearitas (Linearity):* Apakah keluaran output harus linear ?
  - d. *Noise:* Apa yang menjadi *noise* dalam hal level dan frekuensi pada lingkungan pengukuran?
2. *Memilih sensor (jika bisa diterapkan).*
  - a. *Parameter:* Apa yang menjadi sifat alami dari keluaran sensor ? resistansi, tegangan atau lainnya?
  - b. *Fungsi transfer (transfer function):* Bagaimana hubungan antar keluaran sensor dengan variabel terukur ? linear, grafis, persamaan, akurasi atau lainnya ?
  - c. *Waktu respon (time response):* Bagaimana waktu repon dari sensor ? apakah lambat atau cepat ?
  - d. *Daerah kerja (range) :* Bagaimana daerah kerja keluaran sensor untuk daerah kerja pengukuran yang diberikan.
  - e. *Daya (Power):* Apa yang menjadi spesifikasi daya dari sensor ? disipasi resistansi maksimum, arus yang diserap atau lainnya ?
3. *Mendesain pengondisi sinyal analog (S/C)*
  - a. *Parameter:* Apa yang menjadi sifat alami dari keluaran ? Yang sering dipakai adalah tegangan, tetapi arus dan frekuensi kadang-kadang juga dipakai pula. Masih banyak kasus yang memakai tegangan sebagai langkah pertama yang dipakai.
  - b. *Daerah Kerja (range):* Berapa daerah kerja yang diharapkan dari parameter keluarannya (misal 0 s/d 5 V, 4 s/d 20 mA, 5 s/d 10 kHz) ?
  - c. *Impedansi masukan (input impedance):* Impedansi masukan bagaimana yang seharusnya terjadi pada pengkondisi sinyal (S/C) jika untuk sinyal masukannya. Hal ini penting untuk menjaga pembebanan dari masukan sinyal tegangan.
  - d. *Impedansi keluaran (Output Impedance):* Impedansi keluaran apa yang seharusnya dari pengkondisi sinyal jika dihubungkan dengan rangkaian beban pada keluarannya.
4. *Catatan untuk mendesain pengkondisi sinyal analog.*
  - a. Jika masukan adalah perubahan resistansi, dan rangkaian jebatan dan pembagi yang harus dipakai, yakinkan untuk mempertimbangkan keduanya dari segi pengaruh ketidaklinearan tegangan keluarannya terhadap resistansi dan pengaruh arus listrik yang melewati sensor yang mempunyai hambatan tersebut.
  - b. Untuk bagian op-amp yang didesain, pendekatan desain yang paling mudah adalah membangun persamaan

matematis antara masukan dan keluarannya. Dari persamaan matematis ini, akan menjawab rangkaian apa yang diperlukan untuk dipakai pada kasus itu. Persamaan ini merepresentasikan fungsi transfer statis dari pengkondisi sinyal. Untuk contoh mendesain rangkaian pengkondisi sinyal dengan op-amp ini bisa dilihat di bab 8.

- c. Selalu mempertimbangkan banyak kemungkinan dari pembebanan sumber tegangan dari pengkondisi sinyal. Pembebanan seperti ini merupakan kesalahan yang vital dalam sistem pengukuran.

### 9.5.2.2 Pengkondisian Sinyal Digital (Digital Signal Conditioning)

Keseluruhan survei menunjukkan bahwa aplikasi elektronika yang terjadi di bidang industri menunjukkan bahwa perkembangan teknik digital terjadi sangat cepat. Ada banyak alasan kenapa hal ini bisa terjadi, tetapi hanya dua alasan pada bagian ini yang penting. Salah satunya adalah pengurangan terhadap ketidakpastian, hubungannya informasi yang dikodekan secara digital dibandingkan dengan informasi secara analog.

Sebagai catatan yang kita bicarakan adalah ketidakpastian (*uncertainty*), buka akurasi (*accuracy*). Jika sebuah sistem menunjukkan informasi secara analog, harus sangat diperhatikan pengaruh noise secara elektronik, penyimpangan penguatan *amplifier*, efek pembebanan, dan masalah lainnya yang biasa terjadi pada pendesainan elektronika analog. Pada

sinyal yang terkodekan secara digital, dimana, kabel pembawa dengan level *high* (1) atau *low* (0), bukan merupakan masalah hubungannya dengan pemrosesan analognya. Maka ada kepastian yang tak terpisahkan di dalam representasi informasi pengkodean secara digital karena tidak mungkin adanya pengaruh yang sifatnya palsu dari informasi tersebut.

Pecacah desimal	Pencacah biner				
	16	8	4	2	1
0					0
1					1
2				1	0
3				1	1
4			1	0	0
5			1	0	1
6			1	1	0
7			1	1	1
8		1	0	0	0
9		1	0	0	1
10		1	0	1	0
11		1	0	1	1
12		1	1	0	0
13		1	1	0	1
14		1	1	1	0
15		1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1
	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
	Pangkat 2				

Gambar 9.31 Pencacahan dalam biner dan desimal

Alasan kedua dari perkembangan elektronika digital adalah pertumbuhan keinginan dalam pemakaian komputer digital dalam proses industri. Komputer digital, secara

Pangkat dari 2	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
Nilai-tempat	16	8	4	2	1	
Biner	1	0	0	1	1	.
Desimal	16	+	0	+	0	+
				2	+	1
						= 19

(a) Pengubah biner menjadi desimal

Pangkat dari 2	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
Nilai-tempat	32	16	8	4	2	1	
Biner	1	0	1	1	1	0	.
Desimal	32	+	0	+	8	+	4
					2	+	0
							= 46

(b) Pengubah biner menjadi desimal

$$10011_2 = 19_{10} \qquad 101110_2 = 46_{10}$$

(c) Ikhtisar dari pengubahan-pengubahan dan penggunaan tikalas kecil (small subscripts) untuk menyatakan radiks dari bilangan

Gambar 9.32 Konversi Bilangan biner - desimal

alami, memerlukan informasi yang terkodekan dalam format digital sebelum informasi tersebut dipakai. Pemakaian pengkodisi sinyal secara digital tentunya menjadi pertanyaan mengapa komputer dipakai secara luas di di industri. Ada beberapa alasan yang bisa menjawab pertanyaan tersebut, yaitu :

1. Sebuah komputer bisa dipakai untuk mengontrol dengan mudah dari suatu sistem kontrol dengan banyak variabel.
2. Melalui pemrograman komputer, ketidaklinearan dari sebuah keluaran sensor dapat di linearkan.
3. Persamaan kontrol yang rumit dapat diselesaikan untuk menentukan fungsi kontrol yang diperlukan.
4. komputer mempunyai kemampuan dalam bentuknya yang kecil berupa rangkaian pemrosesan digital yang

kompleks, sebagai rangkaian yang terintergrasi (IC = *integrated circuit*).

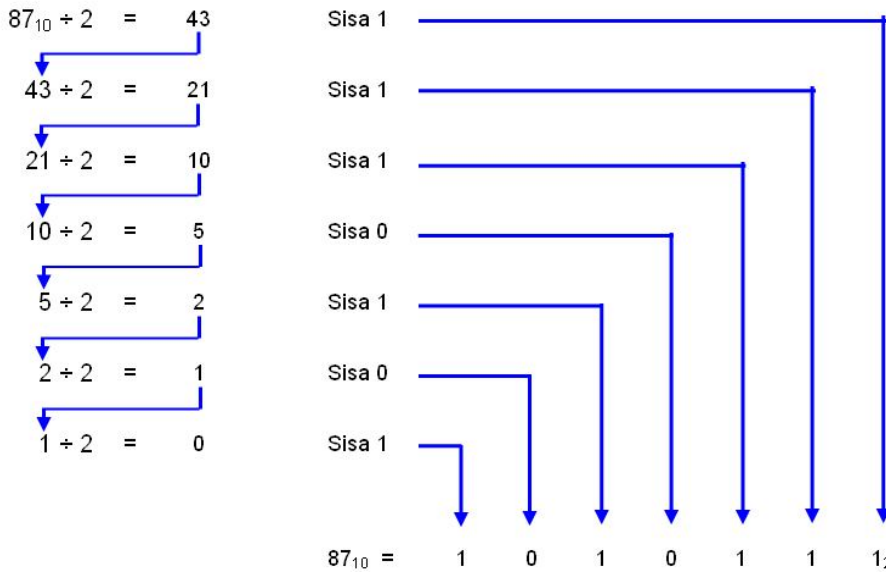
Akhirnya, perkembangan mikro-prosesor telah menyempurnakan suatu perubahan bentuk kontrol proses secara digital sebagai dasar sistem kontrol. Dengan mikroprosesor (dasar dari komputer), implementasi sebuah komputer sebagai dasar sistem kontrol telah menjadi hal yang praktis, dan dengan itu tentunya diperlukan pengetahuan mengenai pengkondisian sinyal secara digital. Teknologi tersebut mengurangi tidak hanya dalam ukuran fisiknya, tetapi juga konsumsi daya dan rata-rata kegagalan yang terjadi.

Pemakaian teknik digital di dalam sistem kontrol memerlukan pengukuran variabe proses dan informasi kontrol yang dikodekan ke dalam bentuk digital. Sinyal digital mempunyai dua jenis level tegangan



Pangkat dari 2	$2^3$	$2^2$	$2^1$	$2^0$	$1/2^1$	$1/2^2$	$1/2^3$	
Nilai-tempat	8	4	2	1	0,5	0,25	0,125	
Biner	1	1	1	0	.	1	0	1
Desimal	8 + 4 + 2 + 0 + 0,5 + 0 + 0,125 = 14,625							

Gambar 9.33 Pengubahan Biner menjadi Desimal



Gambar 9.34 Pengubahan Desimal menjadi Biner

yang sederhana di dalam sebuah kabel. Kita katakan bahwa informasi digital mempunyai kondisi *high* (H atau 1) dan *low* (L atau 0) pada sebuah kabel yang membawa sinyal digital. Sebelum belajar mengenai pengolahan sinyal digital, sebaiknya perlu kita pelajari dulu mengenai konsep bilangan dan dasar elektronika digital terlebih dahulu.

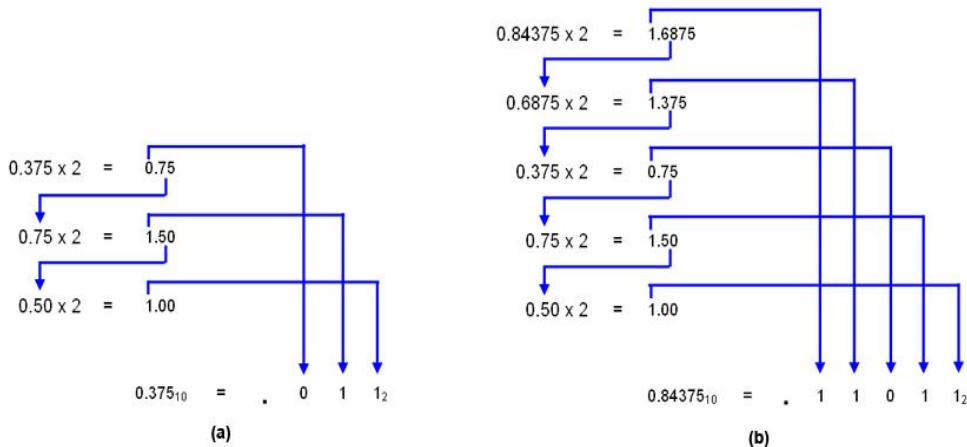
#### 9.5.2.2.1 Sistem Bilangan

Setiap orang mengenal sistem bilangan desimal. Sistem ini memakai simbol 0 s/d 9. Sistem desimal juga

mempunyai karakteristik nilai-tempat (*place-value*). Marilah kita tinjau bilangan desimal 238. Pada bilangan tersebut, angka 8 terdapat pada angka satuan atau posisi satuan. Angka 3 pada posisi puluhan, dan oleh karena itu tiga puluhan berarti 30 satuan. Angka 2 tersebut pada posisi ratusan dan berarti dua ratusan, atau 200 satuan. Penambahan 200 + 30 + 8 menghasilkan angka desimal total sebesar 238. Sistem bilangan desimal juga disebut sistem basis 10 (*base 10 system*). Disebut demikian karena sistem ini mempunyai 10 simbol yang berbeda. Sistem basis

10 juga dikatakan mempunyai suatu radiks (radix) 10. Radiks dan basis merupakan istilah yang mempunyai arti yang persis sama.

sebelah kanan) merupakan *bit yang tidak signifikan* (LSB, *least significant bit*). Dengan kata lain, bila angka 1 muncul pada kolom sebelah kanan,



Gambar 9.35 Pengubahan Pecahan Desimal menjadi Pecahan Biner

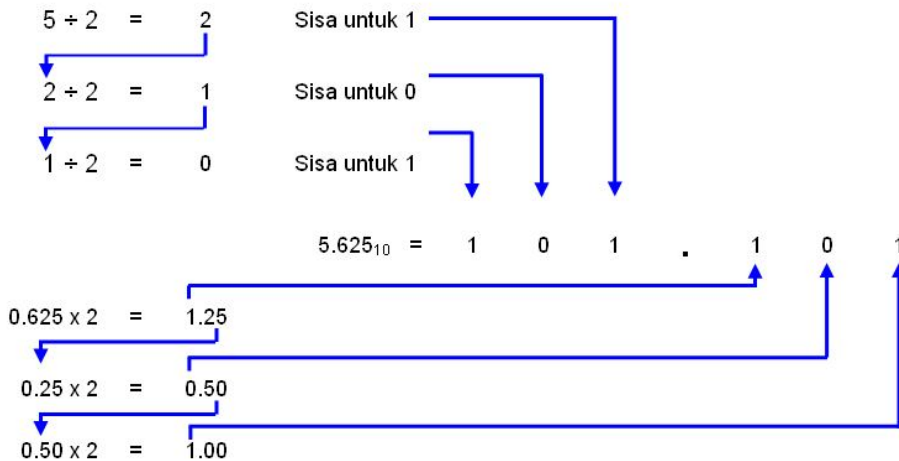
Bilangan biner (basis 2) banyak digunakan di dalam rangkaian digital. Bilangan oktal (basis 8) dan heksadesimal (basis 16) juga digunakan sampai suatu tingkatan tertentu di dalam sistem digital.

Kita dapat menghitung dengan menggunakan semua sistem bilangan tersebut di atas (desimal, biner, oktal, dan heksadesimal). Semua sistem bilangan ini juga mempunyai karakteristik nilai-tempat.

Sistem bilangan biner hanya menggunakan dua simbol (0,1). Bilangan ini dikatakan mempunyai radiks 2 dan biasanya disebut sistem bilangan basis 2. Setiap biner digit (biner digit) disebut bit. Pencacahan dalam biner diilustrasikan pada Gambar 9.31. Bilangan biner diperlihatkan pada kolom sebelah kanan dari Gambar 9.31 tersebut. Sedangkan ekivalen-desimalnya diperlihatkan pada kolom sebelah kiri. Perhatikan bahwa angka 1 (kolom

maka hitungan biner ditambah dengan 1. Bagian kedua dari kanan adalah angka 2. Angka 1 yang muncul pada kolom ini (seperti pada baris desimal dua) berarti bahwa hitungan ditambah dengan 2. Tiga nilai-tempat lainnya juga ditunjukkan pada Gambar 9.31 (angka empatan, delapanan, dan enambelasan). Catatan bahwa masing-masing nilai-angka yang lebih besar merupakan pangkat dari 2 yang ditambahkan. Angka satuan sebenarnya adalah  $2^0$ , angka duaan adalah  $2^1$ , angka empatan adalah  $2^2$ , angka delapanan adalah  $2^3$ , dan angka enambelasan adalah  $2^4$ . Dalam elektronika digital, adalah biasa untuk menghafal sekurang-kurangnya urutan pencacahan biner dari 0000 sampai 1111 (dibaca: satu, satu, satu, satu) atau desimal 15.

Tinjau bilangan yang ditunjukkan pada Gambar 9.32 (a). Gambar ini menunjukkan bagaimana mengubah



Gambar 9.36 Pengubahan Desimal Pecahan Biner

biner 10011 (dibaca: satu, nol, nol, satu, satu) menjadi ekuivalen desimalnya. Perlu kita perhatikan bahwa, untuk setiap 1 bit dalam bilangan biner, ekuivalen-desimal untuk nilai-tempat tersebut dituliskan di bawahnya. Kemudian bilangan-bilangan desimal tersebut ditambahkan ( $16 + 2 + 1 = 19$ ) untuk menghasilkan ekuivalen-desimalnya. Dengan demikian biner 10011 sama dengan desimal 19.

Tinjau bilangan biner 101110 pada Gambar 9.32 (b). Dengan menggunakan prosedur yang sama, setiap 1 bit pada bilangan biner menghasilkan suatu ekuivalen-desimal untuk nilai-tempat tersebut. *Bit paling signifikan* (MSB, *most significant bit*) dari bilangan biner tersebut adalah sama dengan 32. Tambahkan  $8 + 4 + 2$  pada 32 sehingga menghasilkan jumlah total 46. Bilangan biner 11110 sama dengan 46. Gambar 9.32 (b) juga menentukan titik biner (sama dengan titik desimal pada bilangan desimal). Bila bekerja dengan bilangan biner bulat, biasanya titik biner diabaikan.

Bagaimana dengan pengubahan bilangan-bilangan pecahan? Gambar 9.33 melukiskan bilangan biner 1110.101 yang sedang diubah menjadi ekuivalen-desimalnya. Nilai-tempat diberikan sepanjang baris sebelah atas. Perlu kita perhatikan nilai setiap posisi di sebelah kanan titik biner. Tata cara untuk mengubah bilangan biner pecahan sama dengan pada bilangan bulat. Nilai-tempat dari setiap 1 bit pada bilangan biner ditambahkan satu sama lain untuk menghasilkan bilangan desimal. Pada soal ini,  $8 + 4 + 2 + 0,5 + 0,125 = 14,625$  dalam desimal. Berapakah nilai bilangan 111? Bilangan tersebut dapat bernilai seratus sebelas dalam desimal atau satu, satu, satu dalam biner. Beberapa buku menggunakan sistem yang ditunjukkan pada Gambar 9.32 (c) untuk menunjukkan basis atau radiks dari suatu bilangan. Dalam hal ini, 10011 merupakan suatu bilangan basis 2 seperti yang ditunjukkan oleh *tikalas-kecil* 2 setelah bilangan tersebut. Bilangan 19 merupakan suatu bilangan basis 10 seperti yang ditunjukkan oleh

Desimal	Biner	Heksadesimal	Desimal	Biner	Heksadesimal
0	0000	0	16	10000	10
1	0001	1	17	10001	11
2	0010	2	18	10010	12
3	0011	3	19	10011	13
4	0100	4	20	10100	14
5	0101	5	21	10101	15
6	0110	6	22	10110	16
7	0111	7	23	10111	17
8	1000	8	24	11000	18
9	1001	9	25	11001	19
10	1010	A	26	11010	1A
11	1011	B	27	11011	1B
12	1100	C	28	11100	1C
13	1101	D	29	11101	1D
14	1110	E	30	11110	1E
15	1111	F	31	11111	1F

Gambar 9.37 Pencacahan dalam sistem bilangan desimal, biner dan heksadesimal

tikalas 10 setelah bilangan tersebut. Gambar 9.32 (c) merupakan ikhtisar pengubahan biner menjadi desimal pada Gambar 9.32 (a) dan (b).

Ubahlah bilangan desimal 87 ke bilangan biner. Gambar 14 menunjukkan suatu metode yang memudahkan untuk membuat pengubahan ini. Mula-mula bilangan desimal 87 dibagi dengan 2, menghasilkan 43 dengan sisa 1. Sisa ini adalah penting dan dicatat pada sebelah kanan. Pada bilangan biner, sisa ini menjadi LSB. Setelah itu hasil bagi (43) dipindahkan seperti yang ditunjukkan oleh anak panah dan menjadi bilangan yang dibagi. Hasilnya ini dibagi 2 secara berulang-ulang sampai hasil bagi menjadi 0 dengan sisa 1, seperti pada baris terakhir pada Gambar 9.34. Pada Gambar tersebut ditunjukkan bahwa desimal 87 sama dengan biner 1010111.

Ubahlah bilangan desimal 0,375 ke bilangan biner. Gambar 9.35 (a) menggambarkan suatu metode untuk mengerjakan tugas ini. Perhatikan bahwa bilangan desimal tersebut (0,375) dikalikan dengan 2, hasil kalinya adalah 0,75. Angka 0 dari bilangan bulat (angka satuan) menjadi bit yang paling dekat dengan titik biner. Kemudian 0,75 dikalikan dengan 2, hasilnya adalah 1,50. 1 pada bilangan bulat (angka satuan) ini merupakan bit berikutnya dalam bilangan biner tersebut. Selanjutnya 0,50 ini dikalikan dengan 2 dan menghasilkan 1,00. 1 pada angka bilangan bulat ini merupakan angka 1 terakhir dalam bilangan biner tersebut. Bila hasil kali adalah 1, maka proses pengubahan telah selesai. Gambar 9.35 (a) menunjukkan desimal 0,375 yang sedang diubah menjadi ekuivalen-biner 0.011.

Pangkat dari 16	$16^2$	$16^1$	$16^0$
Nilai-tempat	256-an	16-an	1-an
Bilangan heksadesimal	2	B	6
	256	16	1
	<u>x 2</u>	<u>x 11</u>	<u>x 6</u>
Desimal	512	+ 176	+ 6 = 694 <sub>10</sub>

(a) Pengubahan heksadesimal menjadi desimal

Pangkat dari 16	$16^2$	$16^1$	$16^0$	$1/16^1$
Nilai-tempat	256-an	16-an	1-an	0,625-an
Bilangan heksadesimal	A	3	F	• C
	256	16	1	0,625
	<u>x 10</u>	<u>x 3</u>	<u>x 15</u>	<u>x 12</u>
Desimal	2560	+ 48	+ 15	+ 0,75 = 2623,75 <sub>10</sub>

(b) Pengubahan pecahan heksadesimal menjadi desimal

Gambar 9.38 Pengubahan heksadesimal ke desimal dan sebaliknya

Gambar 9.35 (b) menunjukkan bilangan desimal 0,84375 yang sedang diubah menjadi biner. Perhatikan lagi bahwa 0.84375 dikalikan dengan 2. Bilangan bulat dari masing-masing hasil kali diletakkan dibawah dan membentuk bilangan biner. Bila hasil kali mencapai 1,00 maka pengubahan telah selesai. Persoalan ini menunjukkan suatu bilangan desimal 0,84375 diubah ke biner 0.11011.

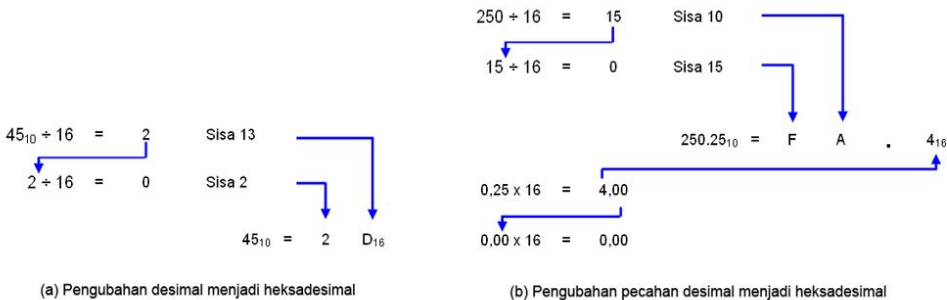
Tinjau angka desimal 5,625. Pengubahan bilangan ini menjadi biner melibatkan dua proses. Bagian bilangan bulat dari bilangan tersebut (5) diproses dengan pembagian berulang di sebelah atas pada Gambar 9.36. Desimal 5 diubah ke biner 1011. Bagian pecahan dan bilangan desimal tersebut (0.625) diubah menjadi biner 0.101 di sebe-

lah bawah Gambar 9.36. Bagian pecahan ini diubah menjadi biner melalui proses perkalian-berulang. Kemudian, bagian bilangan bulat dan pecahan tersebut dikombinasikan. Dengan demikian menunjukkan bahwa desimal 5.625 sama dengan biner 101.101.

Sistem bilangan heksadesimal mempunyai radiks 16 dan disebut sebagai sistem bilangan basis 16. Bilangan heksadesimal menggunakan simbol 0 - 9, A, B, C, D, E, dan F sebagaimana yang ditunjukkan pada kolom heksadesimal dari tabel pada Gambar 9.37. Huruf A adalah untuk cacahan 10, B untuk 11, C untuk 12, D untuk 13, E untuk 14, dan F untuk 15. Keuntungan dari sistem heksadesimal adalah kegunaannya dalam pengubahan secara langsung dari bilangan biner 4-bit. Perhatikan

dalam pengubahan secara langsung dari bilangan biner 4-bit. Perhatikan bahwa pada seksi yang dihitamkan dari Gambar 9.37, setiap bilangan biner 4-bit dari 0000 sampai 1111 dapat dinyatakan oleh suatu digit heksadesimal yang unik, mempunyai 16 kemungkinan yang tidak sama.

bagian bawah pada Gambar 9.38 (a). Kolom 1 menunjukkan enam satuan. Kemudian 6 kini ditambahkan pada baris desimal. Nilai-nilai desimal tersebut ditambahkan ( $512 + 176 + 6 = 694$ ), dan menghasilkan  $694_{10}$ . Gambar 9.38 (a) menunjukkan bahwa  $2B6_{16}$  sama dengan  $694_{10}$ .

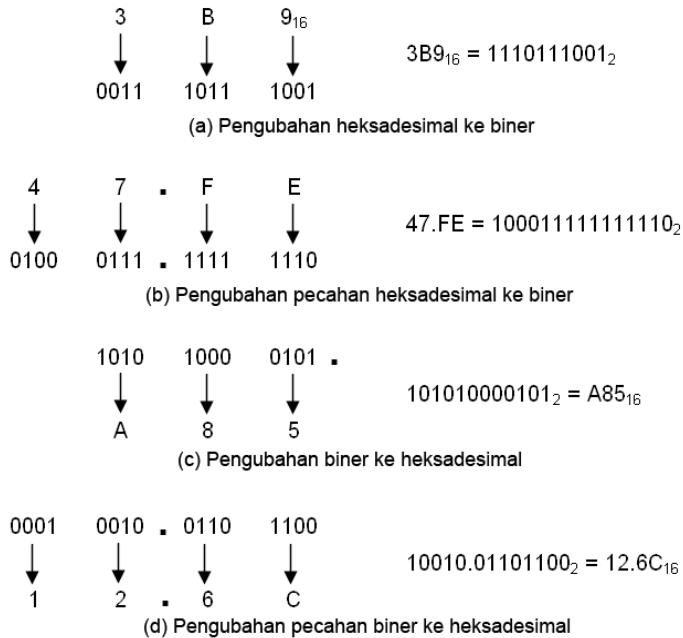


Gambar 9.39 Pengubahan desimal ke heksadesimal dari bilangan pecahan dan non-pecahan

Lihatlah baris yang diberi label 16 pada kolom desimal dalam Gambar 9.37. Ekuivalen heksadesimalnya adalah 10. Hal ini menunjukkan bahwa sistem bilangan heksadesimal menggunakan gagasan nilai-tempat. Angka 1 (dalam  $10_{16}$ ) mempunyai nilai 16 satuan, sedangkan angka 0 bernilai nol unit.

Ubahlah bilangan heksadesimal  $2136$  ke bilangan desimal. Gambar 9.38 (a) menunjukkan proses yang telah kita kenal. Angka 2 terdapat pada posisi 256-an sehingga  $2 \times 256 = 512$ , yang tertulis pada baris desimal. Digit heksadesimal B muncul pada kolom 16-an. Perhatikan, bahwa pada Gambar 9.37, heksadesimal B bersesuaian dengan desimal 11. Ini berarti bahwa terdapat sebelum angka 16-an ( $16 \times 11$ ), yang menghasilkan 176. Bilangan 176 ini ditambahkan pada jumlah desimal di

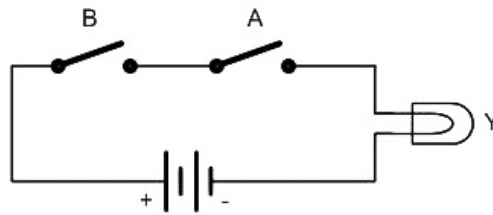
Ubahlah bilangan desimal heksadesimal  $A3F.C$  ke ekuivalen desimalnya. Gambar 9.38 (b) merinci soal ini. Mula-mula tinjau kolom 256-an. Digit heksadesimal A berarti bahwa 256 harus dikalikan dengan 10 dan menghasilkan 2560. Bilangan desimal itu menunjukkan bahwa bilangan tersebut mengandung tiga 16-an, dan oleh karena itu  $16 \times 3 = 48$ , yang ditambahkan pada baris desimal. Kolom 1 berisi digit heksadesimal F, yang berarti  $1 \times 15 = 15$ . Angka 15 ini ditambahkan pada baris desimal. Kolom 0,0625-an berisi digit heksadesimal C, yang berarti  $12 \times 0,0625 = 0,75$ . Selanjutnya 0,75 ini ditambahkan pada baris desimal. Penambahan seluruh isi baris desimal tersebut ( $2560 + 48 + 15 + 0,75 = 2623,75$ ) menghasilkan bilangan desimal  $2623,75$ . Gambar 9.38 (b) mengubah  $A3F.C_{16}$  menjadi  $2623,75_{10}$ .



Gambar 9.40 Pengubahan heksadesimal ke biner dan sebaliknya pada angka pecahan dan non-pecahan

Sekarang balikkan proses tersebut dan ubahlah bilangan desimal 45 ke ekuivalen-heksadesimalnya. Gambar 9.39 (a) merinci proses pembagian dengan 16 yang kita kenal. Mula-mula bilangan desimal 45 dibagi dengan 16, dan menghasilkan 2 dengan sisa 13. Sisa 13 (D dalam heksadesimal) ini menjadi LSB dari bilangan heksadesimal terdekat. Hasil bagi (2) dipindahkan ke posisi bilangan yang dibagi dan kemudian dibagi dengan 16. Ini menghasilkan 0 dengan sisa 2. Angka 2 ini menjadi digit berikutnya dalam bilangan heksadesimal tersebut. Proses telah selesai karena bagian bilangan bulat dari hasil bagi adalah 0. Proses pada Gambar 9.39 (a) tersebut mengubah bilangan desimal 45 menjadi angka heksadesimal 2D.

Ubahlah angka desimal 250,25 ke angka heksadesimal. Pengubahan ini harus dikerjakan dengan menggunakan dua proses seperti yang ditunjukkan pada Gambar 9.39 (b). Bagian bilangan bulat dari bilangan desimal (250) diubah menjadi heksadesimal dengan menggunakan proses pembagian dengan-16 yang berulang. Sisa 10 (A dalam heksadesimal) dan sisa 15 (F dalam heksadesimal) membentuk bilangan bulat heksadesimal FA. Bagian pecahan dari 250,25 dikalikan dengan 16 ( $0,25 \times 16$ ), hasilnya adalah 4,00. Bilangan bulat 4 ini dipindahkan ke posisi seperti yang ditunjukkan pada Gambar 9.39 (b). Pengubahan yang telah disebutkan tersebut menunjukkan bilangan desimal 250,25 sama dengan bilangan heksadesimal FA.4.



(a) Rangkaian AND yang menggunakan saklar

Saklar masukan		Nyala keluaran
B	A	Y
buka	buka	tidak
buka	tutup	tidak
tutup	buka	tidak
tutup	tutup	ya

(b) Tabel kebenaran

Gambar 9.41 Rangkaian AND dengan menggunakan saklar dan tabel kebenarannya

Keuntungan utama sistem heksadesimal adalah kemudahannya pengubahannya menjadi biner. Gambar 9.40 (a) menunjukkan bilangan heksadesimal 3B9 yang diubah menjadi biner. Perhatikan bahwa setiap digit heksadesimal membentuk suatu kelompok yang terdiri atas empat digit biner atau bit. Kemudian, kelompok-kelompok bit tersebut dikombinasikan untuk membentuk bilangan biner. Dalam hal ini,  $3B9_{16}$  sama dengan  $1110111001_2$ .

Pengubahan heksadesimal menjadi biner yang lain dirinci pada Gambar 9.40. Sekali lagi setiap digit heksadesimal membentuk sekelompok 4-bit dalam bilangan biner. Titik heksadesimal dikeluarkan untuk membentuk titik biner. Bilangan heksadesimal 47.FE diubah menjadi bilangan biner  $1000111.1111111$ . Oleh karena kekompakannya, angka

heksadesimal jelas jauh lebih mudah dituliskan daripada deretan panjang 1 dan 0 dalam biner. Sistem heksadesimal dapat dianggap sebagai suatu metode penulisan cepat untuk menuliskan bilangan biner. Gambar 9.40 (b) menunjukkan angka biner  $101010000101$  yang sedang dikonversikan menjadi heksadesimal. Mula-mula kita bagi angka biner tersebut menjadi kelompok-kelompok 4-bit yang dimulai pada titik biner. Kemudian setiap kelompok dari empat bit ini diterjemahkan menjadi digit ekuivalen-heksadesimal. Gambar 9.40 (c) menunjukkan bahwa biner  $101010000101$  sama dengan heksadesimal A85.

Pengubahan biner menjadi heksadesimal yang lain dilukiskan pada Gambar 9.40 (a). Di sini biner  $10010.011011$  akan diterjemahkan



menjadi heksadesimal. Mulai dari titik biner, bilangan biner tersebut mula-mula dibagi menjadi kelompok-kelompok yang terdiri atas empat-bit. Tiga angka 0 ditambahkan pada kelompok yang paling kiri dan membentuk 0001. Dua angka 0 ditambahkan pada kelompok yang paling kanan dan membentuk 1100. Sekarang kelompok mempunyai empat bit dan diubah menjadi digit heksadesimal seperti yang ditunjukkan pada Gambar 9.40 (d). Bilangan biner 1010.011011 sama dengan  $12.6C_{16}$ .

digunakan dalam gerbang logika adalah TINGGI (HIGH) atau RENDAH (LOW). Dalam buku ini, tegangan TINGGI berarti biner 1, sedangkan tegangan RENDAH berarti biner 0. Harus kita ingat bahwa gerbang logika merupakan rangkaian elektronika. Rangkaian ini hanya tanggap (*respond*) terhadap tegangan TINGGI (yang disebut satuan), kebanyakan menggunakan level 5 Volt atau tegangan RENDAH (tegangan tanah) yang disebut nol atau level 0 Volt. Semua sistem digi-



(a) Simbol gerbang AND

masukan		keluaran
B	A	Y
0	0	0
0	1	0
1	0	0
1	1	1

0 = tegangan rendah  
1 = tegangan tinggi

(b) Tabel kebenaran AND

Gambar 9.42 Simbol gerbang AND dan tabel kebenarannya

#### 9.5.2.2.2 Gerbang Logika

Gerbang logika (*logic gate*) merupakan dasar pembentuk sistem digital. Gerbang logika beroperasi dengan bilangan biner. Oleh karena itu gerbang tersebut disebut gerbang logika biner. Tegangan yang

tal disusun hanya menggunakan tiga gerbang logika dasar. Gerbang-gerbang dasar ini disebut gerbang AND, gerbang OR, dan gerbang NOT. Bab ini membahas gerbang-gerbang logika dasar yang sangat penting.

$$A \cdot B \cdot C = Y$$

(a) Pernyataan boolean 3 variabel



(b) Simbol gerbang AND 3 masukan

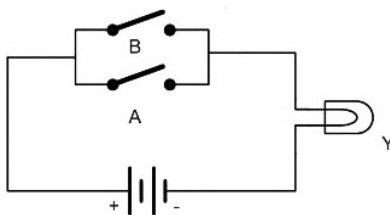
masukan			keluaran
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(c) Tabel kebenaran dengan 3 variabel

Gambar 9.43 Simbol gerbang AND 3 masukan dan tabel kebenarannya

Gerbang AND disebut gerbang "semua atau tidak satu pun". Bagan pada Gambar 9.41 (a) mengilustrasikan gagasan gerbang AND. Lampu (Y) hanya akan menyala bila kedua saklar masukan (A dan B) tertutup. Semua kemungkinan kombinasi untuk saklar A dan B ditunjukkan pada Gambar 9.41 (b). Tabel pada gambar ini disebut *tabel kebenaran (truth table)*. Tabel kebenaran ini menunjukkan bahwa keluaran Y menyala jika semua masukannya tertutup.

dan B, sedangkan keluaran dinyatakan sebagai Y. Simbol tersebut merupakan simbol untuk suatu gerbang AND 2-masukan. Tabel kebenaran untuk gerbang AND 2-masukan ini ditunjukkan pada Gambar 9.42 (b). Masukan-masukan ditunjukkan sebagai digit biner (bit). Perlu kita perhatikan bahwa keluaran akan menjadi 1 hanya bila kedua masukan A dan B adalah 1. Biner 0 didefinisikan sebagai suatu tegangan RENDAH atau tegangan tanah. Biner 1 didefinisikan sebagai tegangan



(a) Rangkaian OR yang menggunakan saklar

Saklar masukan		Nyala keluaran
B	A	Y
buka	buka	tidak
buka	tutup	ya
tutup	buka	ya
tutup	tutup	ya

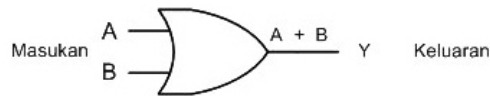
(b) Tabel kebenaran

Gambar 9.44 Rangkaian OR dengan menggunakan saklar dan tabel kebenarannya

*Simbol logika standar* untuk gerbang AND digambarkan pada Gambar 9.42 (a). Simbol ini menunjukkan masukan sebagai A

TINGGI. Kebanyakan tegangan TINGGI mempunyai nilai sekitar + 5 volt (V).

*Aljabar Boolean* merupakan bentuk logika simbolik yang



(a) Simbol gerbang OR

masukan		keluaran
B	A	Y
0	0	0
0	1	1
1	0	1
1	1	1

0 = tegangan rendah  
1 = tegangan tinggi

(b) Tabel kebenaran OR

Gambar 9.45 Simbol gerbang OR dan tabel kebenarannya

menunjukkan bagaimana gerbang-gerbang logika beroperasi. *Pernyataan Boolean* merupakan suatu metode "*tulisan cepat*" untuk menunjukkan apa yang terjadi di dalam rangkaian logika. *Pernyataan Boolean* untuk rangkaian pada Gambar 9.42 adalah

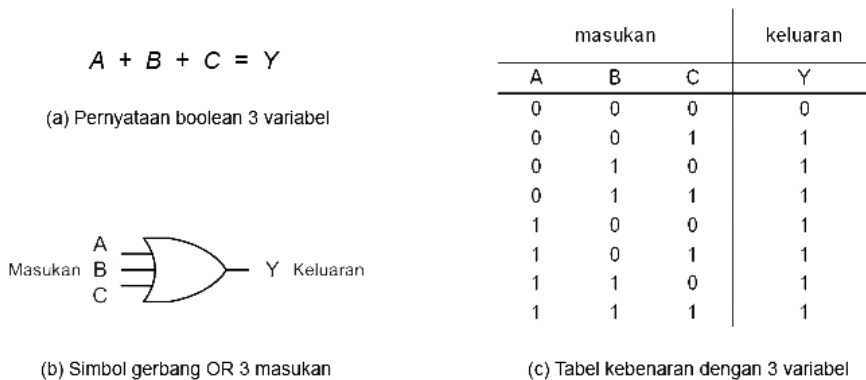
$$A \cdot B = Y$$

Pernyataan Boolean tersebut dibaca sebagai A AND (. berarti AND) B sama dengan keluaran Y. Kadangkala tanda titik (.) dihilangkan dari pernyataan Boolean. Pernyataan Boolean untuk gerbang AND 2-masukan tersebut akan menjadi:

$$AB = Y$$

Pernyataan Boolean tersebut dibaca sebagai A AND B sama dengan keluaran Y. Tanda titik (.) dalam aljabar Boolean berarti fungsi logika AND dan *bukan* berupa tanda kali seperti pada aljabar biasa.

Rangkaian logika banyak yang mempunyai tiga variabel. Gambar 9.43 (a) melukiskan pernyataan Boolean untuk satu gerbang AND 3-masukan. Variabel-variabel masukan tersebut adalah A, B, dan C. Keluarannya dinyatakan sebagai Y. Simbol logika untuk pernyataan AND 3-masukan diilustrasikan pada Gambar 9.43 (b). Tiga masukan (A, B, C) tersebut terdapat pada sebelah kiri dari simbol. Keluaran tunggal (Y) terdapat pada sebelah kanan simbol. Tabel kebenaran pada Gambar 9.43 (c) memperlihatkan delapan kemungkinan kombinasi dari variabel A, B, dan C. Perlu kita perhatikan bahwa garis bagian atas pada tabel tersebut merupakan hitungan biner 000. Kemudian hitungan biner dilanjutkan dengan urutan ke atas 001, 010, 011, 100, 101, 110, dan akhirnya 111. Kita perhatikan pula bahwa keluaran gerbang AND tersebut akan menjadi 1 hanya bila semua masukannya 1.



Gambar 9.46 Simbol gerbang OR 3 masukan dan tabel kebenarannya

Aturan-aturan aljabar Boolean mengatur bagaimana, gerbang AND beroperasi. Aturan formal untuk fungsi AND adalah:

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot A = A$$

$$A \cdot \bar{A} = 0$$

Kita dapat membuktikan tabel dari aturan ini dengan memeriksa kembali tabel kebenaran pada Gambar 9.42. Aturan-aturan tersebut merupakan pernyataan umum mengenai fungsi AND. Gerbang-gerbang AND harus mengikuti aturan tersebut. Perhatikan tanda strip di atas variabel pada aturan terakhir. Tanda strip di atas variabel tersebut berarti *tidak* A, atau lawan dari A.

Gerbang OR disebut gerbang "*setiap atau semua*". Bagan pada Gambar 9.44 (a) di atas mengilustrasikan gagasan bagaimana gerbang OR berkerja. Lampu (Y) akan menyala bila saklar A *atau* saklar B tertutup. Lampu juga akan menyala apabila kedua saklar, baik

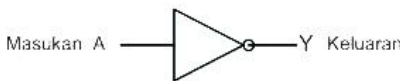
saklar A maupun B tertutup. Lampu (Y) *tidak* akan menyala bila kedua saklar (A dan B) terbuka. Semua kemungkinan kombinasi saklar tersebut ditunjukkan pada Gambar 9.44 (b). Tabel kebenaran tersebut akan menggambarkan dengan terperinci fungsi OR dari rangkaian saklar dan lampu. Keluaran rangkaian OR akan memungkinkan bila setiap atau semua saklar masukan tertutup.

Simbol logika standar untuk suatu gerbang OR digambarkan pada Gambar 9.45 (a). Perhatikan perbedaan bentuk gerbang OR tersebut. Gerbang OR mempunyai dua masukan yang diberi label A dan B. Keluaran diberi label Y. Pernyataan Boolean "*tulisan cepat*" untuk fungsi OR ini diberikan sebagai  $A + B = Y$ . Perlu kita perhatikan bahwa simbol tanda tambah (+) dalam aljabar Boolean berarti OR. Pernyataan  $(A + B = Y)$  dibaca sebagai A OR (+ berarti OR) B sama dengan keluaran Y. Akan kita perhatikan pula bahwa tanda tambah tidak berarti penambahan seperti pada aljabar biasa.

Tabel kebenaran untuk gerbang OR 2-masukan digambarkan pada Gambar 9.45 (b). Variabel masukan

(A dan B) diberikan pada sebelah kiri. Hasil keluaran (Y) diperlihatkan pada kolom sebelah kanan dari tabel tersebut. Gerbang OR tersebut terbuka (keluaran adalah 1) setiap kali muncul 1 pada setiap atau semua masukan. Seperti sebelumnya, 0 didefinisikan sebagai tegangan RENDAH (tegangan tanah), sedangkan 1 pada tabel kebenaran menyatakan tegangan TINGGI (+ 5 V).

Pernyataan Boolean untuk suatu



(a) Simbol gerbang NOT

masukan	keluaran
A	Y
0	1
1	0

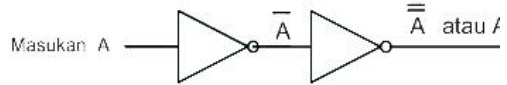
(b) Tabel kebenaran Gerbang NOT

Tabel kebenaran untuk gerbang OR 3-masukan diilustrasikan pada Gambar 9.46 (c). Variabel-variabel (A, B, dan C) ditunjukkan pada sisi sebelah kiri dari tabel. Keluaran (Y) dituliskan pada kolom sebelah kanan. Setiap kali muncul 1 pada setiap masukan, maka keluaran akan menjadi 1.

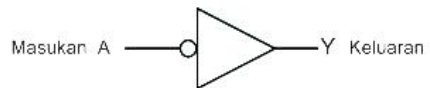
Aturan-aturan aljabar Boolean mengatur bagaimana suatu gerbang OR akan beroperasi. Aturan aturan formal untuk fungsi OR adalah :

$$Y = \bar{A}$$

(c) Pernyataan boolean NOT



(d) Pembalik ganda



(e) Simbol alternatif pembalik

Gambar 9.47 Simbol gerbang NOT, tabel kebenaran, pernyataan boolean, pembalik ganda dan simbol alternatif pembalik

gerbang OR 3-masukan dituliskan pada Gambar 9.46 (a). Pernyataan tersebut dibaca A OR B OR C sama dengan keluaran Y. Sekali lagi, tanda tambah menandakan fungsi OR.

Simbol logika untuk gerbang OR 3-masukan digambarkan pada Gambar 9.46 (b). Masukan A, B, dan C ditunjukkan pada sebelah kiri dari simbol. Keluaran Y ditunjukkan pada sebelah kanan dari simbol. Simbol ini menyatakan beberapa rangkaian yang akan menghasilkan fungsi OR.

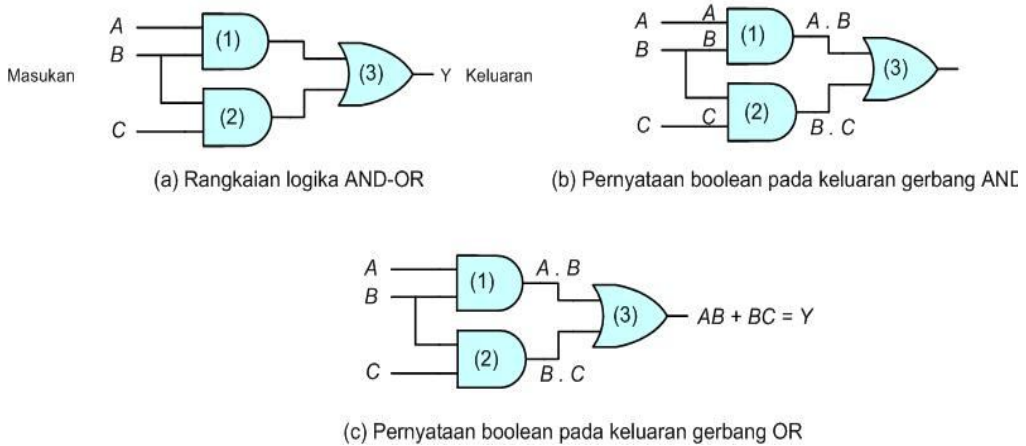
$$A + 0 = 1$$

$$A + 1 = 1$$

$$A + A = A$$

$$A + \bar{A} = 1$$

Dengan melihat tabel kebenaran pada Gambar 9.45 (b) akan membantu pemeriksaan aturan-aturan ini. Pernyataan-pernyataan umum tersebut berlaku untuk fungsi



Gambar 9.48 Rangkaian kombinasi gerbang AND dan OR

OR. Tanda strip di atas variabel terakhir tersebut di atas berarti *tidak* A atau lawan dari A.

Gerbang NOT disebut juga pembalik. Gerbang NOT atau pembalik merupakan suatu gerbang yang tidak biasa dalam hal jumlah input dan outputnya. Gerbang NOT hanya mempunyai satu masukan dan satu keluaran, hal ini berbeda dengan gerbang lainnya. Gambar 9.47 (a) mengilustrasikan simbol logika untuk pembalik atau gerbang NOT.

Proses pembalikan merupakan hal yang sederhana. Gambar 9.47 (b) menunjukkan tabel kebenaran untuk gerbang NOT. Masukan selalu berubah menjadi lawannya. Bila masukan adalah 0, maka gerbang NOT akan memberikan *komplemen* atau lawannya yaitu 1. Bila masukan gerbang NOT adalah 1, maka rangkaian gerbang NOT akan mengkomplemennya menjadi 0. Pembalikan ini juga disebut *pengkomplemenan*. Istilah pembalikan, pengkomplemenan dan penginversian semuanya mempunyai arti yang sama.

Pernyataan Boolean untuk sistem pembalikan ditunjukkan pada Gambar 9.47 (c). Pernyataan  $Y = \bar{A}$  dibaca sebagai Y sama dengan keluaran bukan A. Tanda strip di atas A berarti komplemen A. Gambar 9.47 (d) mengilustrasikan apa yang akan terjadi bila dua pembalik digunakan sekaligus. Pernyataan Boolean dituliskan di atas garis di antara pembalik tersebut. Masukan A dibalik menjadi  $\bar{A}$  (bukan A). Kemudian,  $\bar{\bar{A}}$  dibalik lagi menjadi berbentuk  $\bar{\bar{\bar{A}}}$  (bukan bukan A). A yang diinversikan dua kali ( $\bar{\bar{A}}$ ) sama dengan A aslinya seperti ditunjukkan pada Gambar 9.47 (d). Pada bagian yang dihitamkan di bawah pembalik, bit 0 merupakan masukan. Bit 0 dikomplemenkan menjadi 1. Bit 1 ini dikomplemenkan lagi menjadi 0. Setelah sinyal digital melalui dua pembalik, maka sinyal tersebut kembali ke bentuk aslinya.

Alternatif simbol logika untuk gerbang NOT atau perubahannya ditunjukkan dalam Gambar 9.47 (e) *inverter*

*bubble* besi sebagai keluaran ataupun sebagai masukan dengan simbol segitiga. Di mana invert bubble timbul pada sisi masukan dengan simbol NOT (Gambar 9.47 (e)) biasanya dicoba didisain pada masukan dengan keadaan yang RENDAH pada beberapa fungsi di dalam rangkaian logika. Alternatif simbol gerbang NOT biasanya digunakan di dalam diagram logika oleh pengusaha.

Aturan-aturan aljabar boolean mengatur aksi dari pembalik atau gerbang NOT. Aturan aljabar boolean formal gerbang NOT adalah sebagai berikut :

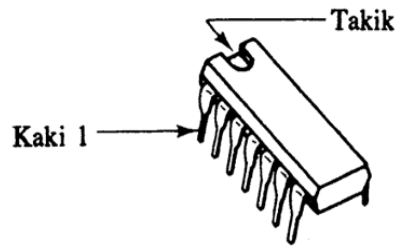
$$\begin{aligned} \bar{0} &= 1 & \bar{1} &= 0 \\ \text{Bila } A &= 1, \text{ maka } \bar{A} &= 0 \\ \text{Bila } A &= 0, \text{ maka } \bar{A} &= 1 \\ &= & \\ \bar{\bar{A}} &= A \end{aligned}$$

Pernyataan-prnyataan umum ini dapat kita periksa dengan diagram dan tabel kebenaran pada Gambar 9.47.

$(A \cdot B) + (B \cdot C) = Y$

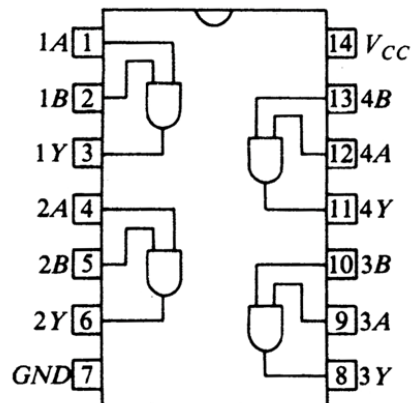
masukan			keluaran
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Gambar 9.49 Perlengkapan kolom keluaran tabel kebenaran dari suatu pernyataan boolean



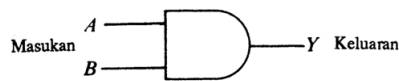
Gambar 9.50 Rangkaian terpadu DIP 14-kaki

Persoalan logika digital sehari-hari banyak yang menggunakan beberapa gerbang logika, bagaimana mengkombinasikan gerbang-gerbang logika AND dan OR.. Pola yang paling umum dari gerbang ditunjukkan pada Gambar 9.48 (a). Pola ini disebut pola AND-OR. Keluaran gerbang AND (1 dan 2) diumpankan ke masukan gerbang OR (3). Akan kita perhatikan bahwa rangkaian logika ini mempunyai tiga masukan (A, B, dan C). Keluaran dari keseluruhan rangkaian dilabelkan sebagai Y.

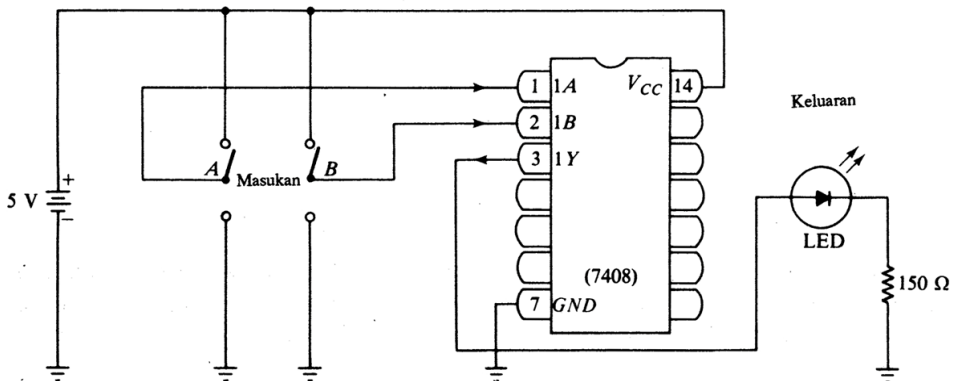


Gambar 9.51 Diagram kaki IC 7408

Mula-mula, marilah kita tentukan pernyataan Boolean yang akan menggambarkan rangkaian logika ini.



(a) Simbol logika gerbang AND



(b) Rangkaian gerbang AND yang menggunakan suatu IC 7408

Gambar 9.52 Simbol logika gerbang AND dan rangkaian gerbang AND dengan IC 7408

Kita mulai dengan memeriksa gerbang (1). Gerbang ini merupakan gerbang AND 2-masukan. Keluaran gerbang ini akan menjadi  $A \cdot B$  ( $A$  AND  $B$ ). Pernyataan ini dituliskan pada keluaran gerbang (1) pada Gambar 9.48 (b). Gerbang (2) juga merupakan suatu gerbang AND 2-masukan. Keluaran dari gerbang ini akan menjadi  $B \cdot C$  ( $B$  AND  $C$ ). Pernyataan ini dituliskan pada keluaran gerbang (2). Selanjutnya, keluaran-keluaran dari gerbang (1) dan (2) di-OR-kan dengan  $BC$ . Pernyataan Boolean yang dihasilkan adalah  $AB + BC = Y$ . Pernyataan tersebut akan sama dengan 1 pada keluaran  $Y$ . Akan kita perhatikan bahwa mula-mula dikerjakan peng-AND-an, dan akhirnya dikerjakan peng-OR-an.

Selanjutnya muncul pertanyaan. Bagaimanakah label kebenaran untuk diagram logika AND-OR pada

Gambar 9.48? Gambar 9.49 akan membantu kita menentukan label kebenaran untuk pernyataan Boolean  $AB + BC = Y$  tersebut. Pernyataan Boolean tersebut mengatakan kepada kita bahwa bila kedua variabel  $A$  AND  $B$  adalah 1, maka keluaran akan menjadi 1. Gambar 9.49 mengilustrasikan bahwa dua garis terakhir dari tabel kebenaran mempunyai satuan baik pada posisi  $A$  maupun pada posisi  $B$ . Maka keluaran 1 ditempatkan di bawah kolom  $Y$ .

Selanjutnya, pernyataan Boolean tersebut di atas menyatakan bahwa kondisi lain juga akan membangkitkan keluaran 1. Pernyataan tersebut menyatakan bahwa  $B$  AND  $C$  juga akan membangkitkan keluaran 1. Dengan melihai ke tabel kebenaran, kita peroleh bahwa baris kelima dari bawah mempunyai satuan pada



Masukan		Keluaran	
A	B		
Tegangan	Tegangan	Tegangan	LED menyala?
GND	GND	GND	tidak
GND	+5 V	GND	tidak
+5 V	GND	GND	tidak
+5 V	+5 V	near +5 V	ya

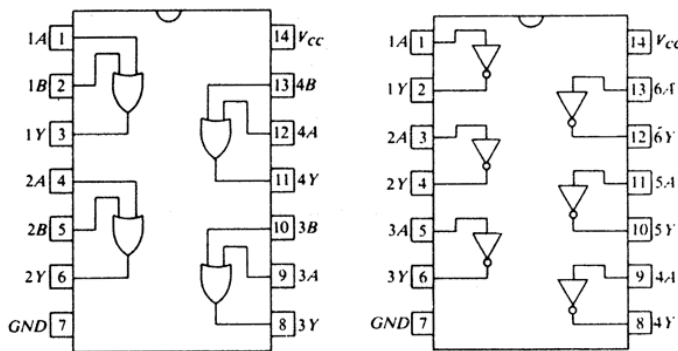
Gambar 9.53 Tabel kebenaran untuk suatu gerbang AND jenis TTL

kedua posisi B AND C. Baris bagian bawah juga mempunyai satuan dalam kedua posisi B AND C. Kedua baris ini Akan menimbulkan keluaran 1. Baris bagian bawah telah mempunyai 1 pada kolom keluaran (Y). Baris kelima dari bawah juga akan memberikan 1 pada kolom keluaran (Y). Hanya baris-baris inilah yang merupakan kombinasi yang akan membangkitkan keluaran 1. Selanjutnya, sisa dari kombinasi tersebut dituliskan sebagai keluaran 0 pada kolom (Y).

Penggunaan gerbang-gerbang logika yang dibahas di atas, dapat dilakukan secara praktis. Fungsi logika dapat diimplemen-

tasikan dalam beberapa cara. Pada masa yang lalu, fungsi logika dihasilkan dengan rangkaian relay dan tabung hampa. Sekarang, *rangkaian terpadu* (IC, *Integrated Circuit*) yang sangat kecil menghasilkan gerbang-gerbang logika.

IC ini berisi ekivalen rangkaian resistor, dioda dan transistor. Jenis yang populer dari IC diilustrasikan pada Gambar 9.50. Paket jenis ini oleh pabrik IC disebut sebagai paket *lipat-dua-dalam-garis* (DIP, *Dual Inline Package*). Selanjutnya, IC khusus ini akan disebut sebagai rangkaian terpadu DIP 14-kaki.

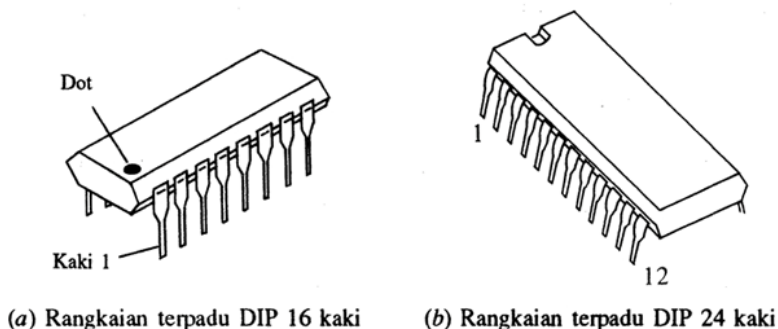


(a) Diagram kaki untuk suatu IC 7432 (b) Diagram kaki untuk suatu IC 7404

Gambar 9.54 Diagram kaki IC 7432 dan IC 7404

Perlu kita perhatikan bahwa bila kita memutar berlawanan arah jarum jam mulai dari tanda takik pada IC Gambar 9.50, maka kaki yang pertama kali kita jumpai adalah kaki nomor 1. Kaki-kaki tersebut diberi nomor berlawanan arah jarum jam mulai dari 1 sampai 14 bila dilihat dari atas IC. Pabrik IC memberikan diagram kaki yang sama dengan yang ditunjukkan untuk IC 7408 pada Gambar 9.51. Kita perhatikan bahwa IC ini berisi empat gerbang AND 2-masukan. Jadi, IC ini disebut gerbang AND 2-masukan lipat empat (*quadruple 2-input AND gate*) Gambar 9.51 melukiskan kaki IC yang diberi nomor dari 1 sampai 14 berlawanan arah jarum jam mulai dari tanda takik.

paling populer. Diberikan diagram logika pada Gambar 9.52 (a), buatlah rangkaian dengan menggunakan suatu IC 7408, Diagram rangkaian untuk rangkaian tersebut diperlihatkan pada Gambar 9.52 (b). Catu daya 5 Volt digunakan untuk semua peralatan TTL. Hubungan daya positif ( $V_{CC}$ ) dan hubungan daya negatif (GND) berturut-turut adalah kaki 14 dan 17. Saklar masukan (A dan B) dirangkai ke kaki 1 dan 2 dari IC 7408 tersebut. Perlu kita perhatikan bahwa bila saklar dalam posisi atas, maka logika (+5 V) dimasukkan pada masukan gerbang AND tersebut. Pada sebelah kanan, dioda pemancar-cahaya (LED, *Light Emitting Diode*) dan resistor pembatas 150 Ohm ( $\Omega$ ) dihubungkan

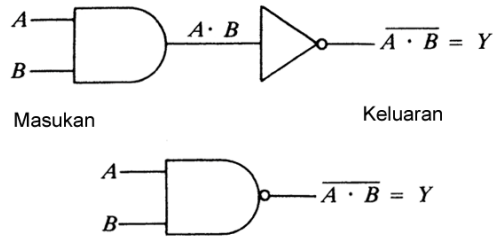


Gambar 9.55 Rangkaian terpadu DIP 16 kaki dan 24 kaki

Hubungan daya pada IC ini adalah kaki GND (kaki 7) dan kaki  $V_{CC}$  (kaki 14). Semua kaki lainnya merupakan keluaran dan masukan untuk empat gerbang AND. IC 7408 merupakan bagian dari kelompok peralatan logika. IC tersebut merupakan salah satu dari banyak peralatan dalam kelompok logika transistor-transistor (TTL, *transistor-transistor logic*) dari rangkaian logika. Akhir-akhir ini, peralatan TTL merupakan peralatan logika yang

ke tanah (ground). Bila keluaran pada kaki 3 adalah TINGGI (+5 V), maka arus akan mengalir melalui LED. Nyala LED menandakan suatu logika TINGGI atau binar 1 pada keluaran gerbang AND.

Tabel kebenaran pada Gambar 9.53 memperlihatkan hasil pengoperasian rangkaian AND 2-masukan. LED pada Gambar 9.52 (b) hanya menyala bila kedua saklar masukan (A dan B) mendapat tegangan +5 Volt.



Gambar 9.56 Gerbang NAND dan persamaan gerbang NAND lainnya

Pabrik rangkaian terpadu juga menghasilkan fungsi logika lain. Gambar 9.54 mengilustrasikan diagram kaki untuk dua IC TTL dasar. Gambar 9.54 (a) merupakan diagram kaki untuk suatu gerbang OR 2-masukan lipat empat. Dengan kata lain, IC 7432 berisi empat gerbang OR 2-masukan. IC ini dapat dirangkai dan diuji menggunakan cara yang sama dengan pengujian gerbang AND pada Gambar 9.52 (b).

IC 7404 yang ditunjukkan pada Gambar 9.54 (b), juga merupakan suatu peralatan TTL. IC 7404 tersebut berisi enam gerbang NOT atau inverter. Oleh pabrik, IC 7404 digambarkan sebagai suatu IC enam pembalik. Perlu kita perhatikan bahwa masing-masing IC mempunyai hubungan daya ( $V_{cc}$  dan GND) yang

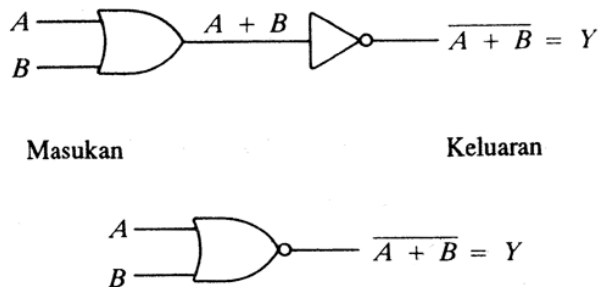
berbeda-beda. Rangkaian logika TTL selalu menggunakan catu daya 5 V dc (arus searah).

Dua variasi dari DIP ICs diilustrasikan pada Gambar 9.55. Rangkaian terpadu yang ditunjukkan pada Gambar 9.55 (a) mempunyai 16 kaki dengan kaki nomor 1 ditandai dengan titik yang terdiri dari suatu norh. IC yang ditunjukkan dalam Gambar 9.55 (b) mempunyai DIP 24 kaki dengan kaki 1 ditempatkan bersebelahan dengan takik.

IC 7408, 7432, dan 7404 yang dipelajari dalam bagian ini semuanya termasuk dalam kelompok logika TTL. Kelompok CMOS terbaru dari IC mempunyai keuntungan karena kelompok CMOS lebih populer. Jenis DIP IC CMOS 74C08 mempunyai quad 2-masukan gerbang AND, 74C04 mempunyai 6 pembalik, atau

Masukan		Keluaran	
B	A	AND	NAND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Gambar 9.57 Tabel kebenaran gerbang AND dan NAND



Gambar 9.58 Gerbang NOR

7402 quad 2 masukan gerbang OR. Jenis gerbang CMOS 74CXX tidak langsung sesuai dengan jenis TTL 7400 dari rangkaian terpadu.

Sistem digital yang paling kompleks, seperti komputer besar, disusun dari gerbang-gerbang logika dasar. Gerbang AND, OR, dan NOT adalah yang paling dasar. Empat gerbang logika lain yang bermanfaat dapat dibuat dari piranti dasar ini. Gerbang-gerbang lainnya disebut gerbang NAND, gerbang NOR, gerbang OR-eksklusif, dan gerbang NOR-eksklusif. Pada akhir bab, kita akan memahami simbol logika, tabel kebenaran, dan aljabar Boolean untuk *tujuh gerbang logika yang digunakan dalam sistem digital*.

Marilah kita perhatikan diagram simbol-logika dibagian atas Gambar 9.56. Gerbang AND dihubungkan ke suatu pembalik. Masukan  $A$  dan  $B$  di-AND-kan untuk membentuk aljabar Boolean  $A \cdot B$ , kemudian dibalik dengan gerbang NOT yang juga disebut sebagai inverter (pembalik). Pada sisi kanan dari pembalik (keluaran rangkaian) ditambahkan tanda strip di atas pada aljabar Boolean tersebut. Aljabar Boolean untuk keseluruhan rangkaian tersebut

adalah  $\overline{A \cdot B} = Y$ . Dalam hal ini dikatakan, bahwa rangkaian tersebut merupakan suatu not AND atau rangkaian NAND.

Simbol logika standar untuk gerbang NAND ditunjukkan pada diagram bagian bawah dari Gambar 9.56. Perlu kita perhatikan bahwa simbol NAND merupakan suatu simbol AND dengan gelembung kecil pada keluarannya. Gelembung tersebut kadang-kadang disebut *suatu gelembung pembalik (invert bubble)*.

Gelembung pembalik ini memberikan suatu metode yang disederhanakan untuk menyatakan gerbang NOT yang dipaparkan pada diagram-bagian-atas pada Gambar 9.56. Tabel kebenaran menggambarkan operasi yang tepat dari suatu gerbang logika. Tabel *kebenaran* untuk gerbang NAND diilustrasikan pada kolom yang tidak dihitamkan dari Gambar 9.57. Tabel kebenaran gerbang-AND juga diberikan untuk menunjukkan bagaimana setiap keluaran dibalik untuk memberikan keluaran NAND. Para siswa menganggap gerbang NAND sebagai suatu gerbang AND yang akan menghasilkan 0 bila dibuka (bila

semua masukan mendapat level logika 1).

Sekarang, fungsi NAND telah menjadi gerbang universal dalam rangkaian digital. Gerbang NAND ini penggunaannya sangat luas dalam sistem digital. Perhatikan tabel kebenaran gerbang NAND dalam Gambar 9.55. *Keluaran* khas dari gerbang NAND adalah RENDAH apabila semua masukan TINGGI.

di atas tersebut akan menghasilkan pernyataan Boolean  $\overline{A+B} = Y$ . Pernyataan ini merupakan suatu fungsi not OR. Fungsi not OR ini dapat digambarkan sebagai suatu simboi logika tunggal yang disebut *gerbang NOR*. Simbol standar untuk gerbang NOR drilustrasikan pada diagram bagian bawah pada Gambar9.58. Perlu kita perhatikan bahwa pada simbol OR tersebut telah

Masukan		Keluaran	
B	A	OR	NOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Gambar 9.59 Tabel kebenaran gerbang OR dan NOR

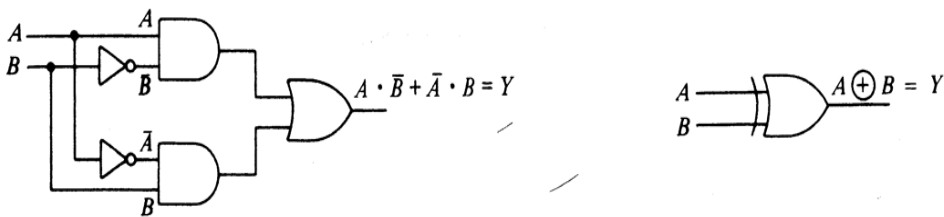
Masukan		Keluaran
B	A	Y
0	0	0
0	1	1
1	0	1
1	1	0

Gambar 9.60 Tabel kebenaran gerbang OR-Eksklusif

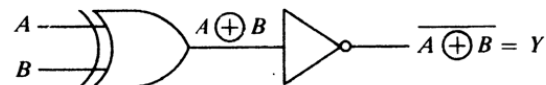
Marilah kita perhatikan diagram logika pada Gambar 9.58. Pembalik telah dihubungkan ke keluaran dari suatu gerbang OR. Pernyataan Boolean pada keluaran pembalik tersebut adalah  $\overline{A+B}$ . Kemudian pembalik tersebut mengkomplemenkan unsur yang di-OR-kan, ditunjukkan dalam pernyataan Boolean dengan tanda strip di atas. Penambahan tanda strip

ditambahkan suatu gelembung pembalik kecil untuk membentuknya menjadi simbol NOR.

Tabel kebenaran pada Gambar 9.59 melukiskan.dengan terperinci operasi gerbang NOR. Perlu kita perhatikan bahwa kolom keluaran dari gerbang NOR tersebut merupakan komplemen (kebalikan) dari kolom OR yang dihitamkan. Dengan kata lain, gerbang NOR menghasilkan keluaran 0 bila gerbang OR menghasilkan 1.

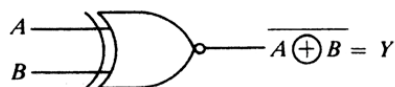


Gambar 9.61 Gerbang OR-Eksklusif (XOR)



Masukan

Keluaran



Gambar 9.62 Gerbang XNOR

Gelembung pembalik kecil pada keluaran simbol NOR seperti itu meningkatkan kita akan gagasan keluaran 0. Perhatikan tabel kebenaran gerbang NOR dalam Gambar 9.59. *Keluaran yang khas* dari gerbang NOR adalah akan TINGGI apabila semua masukan RENDAH.

*Gerbang OR-eksklusif* disebut juga sebagai gerbang "setiap tetapi tidak semua". Istilah OR-eksklusif sering kali disingkat sebagai *XOR*. Tabel kebenaran untuk fungsi XOR diilustrasikan pada Gambar 9.60. Dengan meneliti tabel kebenaran pada Gambar 9.60 tersebut, akan terlihat bahwa table kebenaran fungsi XOR tersebut sama seperti tabel kebenaran fungsi OR, kecuali bila semua masukan gerbang XOR adalah 1, keluaran gerbang XOR akan membangkitkan 0 (*low*). Ger-

bang XOR hanya akan *terbuka bila muncul satuan bilangan ganjil pada masukan*. Baris 2 dan 3 dari tabel kebenaran mempunyai satuan bilangan ganjil, dan oleh keirena itu keluaran akan terbuka dengan level logika 1. Baris 1 dan 4 dari tabel kebenaran tersebut berisi satuan bilangan genap (0, 2), dan oleh karena itu gerbang XOR tidak terbuka dan akan muncul 0 pada keluaran. Gerbang XOR dapat disebut sebagai suatu rangkaian pemeriksa bit-ganjil.

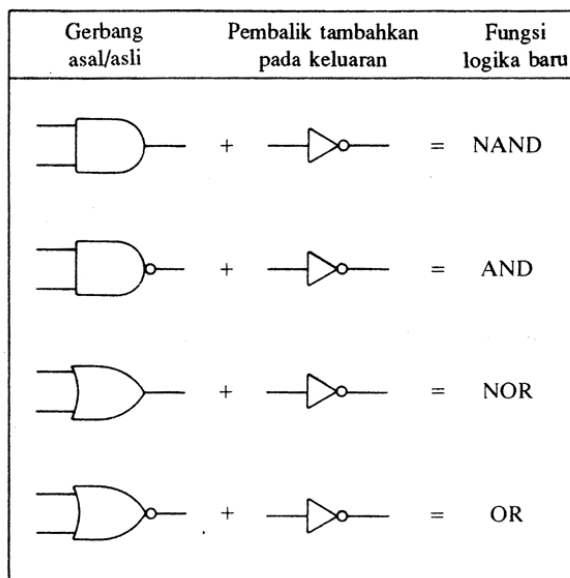
Ajibar Boolean untuk gerbang XOR dapat disusun dari label kebenaran pada Gambar 9.60. Pernyataan tersebut akan menjadi  $A \cdot \bar{B} + \bar{A} \cdot B = Y$ . Dengan pernyataan Boolean ini, rangkaian logika dapat disusun dengan menggunakan gerbang AND, gerbang OR, dan pembalik.

Rangkaian logika seperti itu digambarkan pada Gambar 9.61 (a). Rangkaian logika ini akan memberikan fungsi logika XOR. Simbol logika standar untuk gerbang XOR dipaparkan pada Gambar 9.61

tambah di dalam lingkaran menandakan fungsi XOR dalam aljabar Boolean. Simbol tersebut menyatakan bahwa masukan A dan masukan B pada Gambar 9.61 (b) di-OR-kan dengan eksklusif satu sama lain.

Masukan		Keluaran	
B	A	XOR	XNOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Gambar 9.63 Tabel kebenaran gerbang XOR dan gerbang XNOR

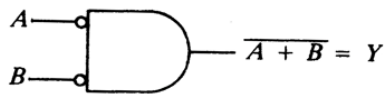


Simbol (+) berarti penambahan pada peta ini

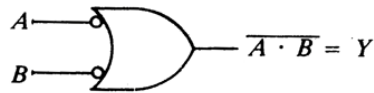
Gambar 9.64 Pengaruh penambahan keluaran gerbang

(b). Kedua diagram simbol logika pada Gambar 9.61 akan menghasilkan tabel kebenaran (XOR) yang sama. Aljabar Boolean sebelah kanan pada Gambar 9.61 (b) merupakan suatu pernyataan XOR yang disederhanakan. Simbol tanda

Pada Gambar 9.62, keluaran dari suatu gerbang XOR dibalik dengan gerbang NOT. Keluaran pembalik pada sisi kanan disebut fungsi NOR-eksklusif (XNOR). Gerbang XOR menghasilkan pernyataan  $A + B$ . Pernyataan gerbang XOR ini



(a) Simbol gerbang NOR



(b) Simbol gerbang NAND

Gambar 9.66 Simbol logika pengganti

Tambahkan pembalik pada masukan	Gerbang asal	Fungsi logika baru
	+	= NOR
	+	= NAND
	+	= OR
	+	= AND

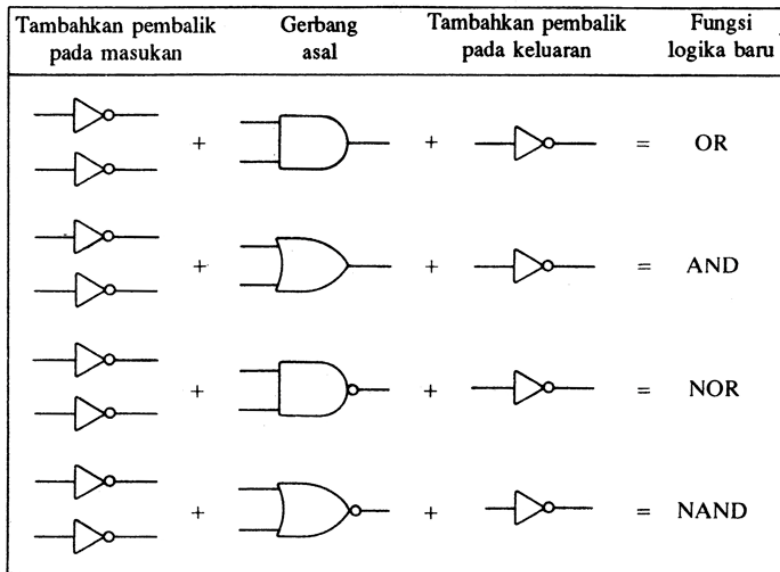
Simbol (+) berarti penambahan pada peta ini.

Gambar 9.65 Pengaruh pembalik masukan gerbang

dibalik sehingga membentuk aljabar Boolean  $\overline{A \oplus B} = Y$ . Pernyataan ini merupakan aljabar Boolean untuk gerbang XNOR. Simbol logika standar untuk gerbang XNOR diilustrasikan pada diagram bagian bawah dari Gambar 9.62. Perlu kita perhatikan bahwa simbol tersebut merupakan suatu simbol XOR dengan gelembung pembalik yang ditempelkan pada keluarannya. Kolom sebelah kanan dari tabel kebe-

nanan pada Gambar 9.63 menggambarkan dengan terperinci operasi gerbang XNOR. Perlu kita perhatikan bahwa semua keluaran gerbang XNOR tersebut merupakan komplemen dari keluaran gerbang XOR. Berbeda dengan gerbang merupakan pendeteksi satuan bilangan-ganjil. Gerbang XNOR mendeteksi *satuan bilangan genap*. Gerbang XNOR akan menghasilkan keluaran 1, bila muncul *satuan bilangan genap* pada masukan.





Simbol (+) berarti penambahan pada peta ini

Gambar 9.67 Pengaruh pembalikan baik masukan maupun keluaran gerbang

Gerbang bisa dirubah dengan menggunakan pembalik. Untuk mengubah suatu fungsi logika ke fungsi logika lainnya diperlukan gerbang logika. Metode yang mudah untuk mengubah fungsi tersebut ialah menggunakan pembalik yang ditempatkan pada keluaran atau masukan gerbang. Telah kita tunjukkan, bahwa suatu inverter yang dihubungkan pada keluaran gerbang AND akan menghasilkan fungsi NAND. Begitu pula pembalik yang ditempatkan pada keluaran gerbang OR akan menghasilkan fungsi NOR. Peta path Gambar 9.64 mengilustrasikan pengubah seperti itu dan pengubah yang lain.

Penempatan pembalik pada semua masukan gerbang logika akan menghasilkan fungsi logika yang diuraikan Gambar 9.65. Pada baris pertama, masukan pada gerbang AND dibalik (simbol tanda tambah menyatakan penambahan dalam

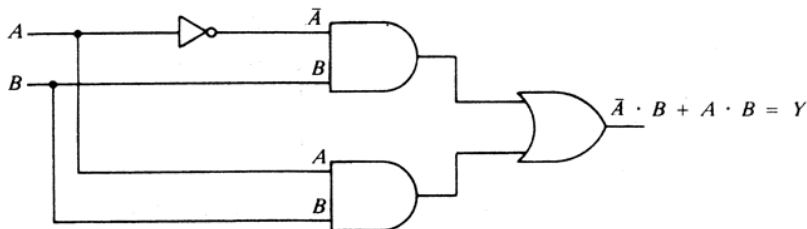
gambar ini). Hal ini akan menghasilkan fungsi NOR pada keluaran gerbang AND. Baris kedua dari Gambar 9.65 menunjukkan masukan gerbang OR yang dibalik. Hal ini akan menghasilkan fungsi NAND. Dua contoh yang pertama menyajikan simbol-simbol bare untuk fungsi NOR dan NAND. Gambar 9.66 mengilustrasikan dua simbol logika yang kadang-kadang digunakan untuk fungsi NOR dan NAND. Gambar 9.66 (a) merupakan simbol logika pengganti untuk suatu gerbang NOR. Gambar 9.66 (b) merupakan simbol logika pengganti untuk suatu gerbang NAND. Simbol-simbol ini kita jumpai dalam beberapa literatur dari pabrik.

Pengaruh pembalik baik pada masukan maupun keluaran dari suatu gerbang logika ditunjukkan pada Gambar 9.67. Sekali lagi, simbol tanda tambah adalah singkatan untuk penambahan. Teknik ini mungkin

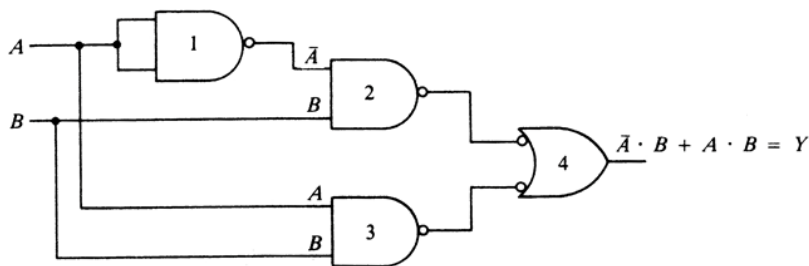
tidak terlalu sering digunakan karena membutuhkan banyak gerbang. Perlu kita perhatikan, bahwa hal ini merupakan metode pengubahan dari fungsi AND ke fungsi OR, dan kembali ke fungsi NAND lagi. Hal ini juga merupakan metode pengubahan dari fungsi NAND ke fungsi NOR, dan kembali ke fungsi NAND lagi.

dibutuhkan untuk melengkapi rangkaian pada Gambar 9.68 (a).

Telah kita sebutkan sebelumnya, bahwa gerbang NAND dianggap sebagai suatu gerbang universal. Gambar 9.68 (b) menunjukkan gerbang NAND yang digunakan untuk melengkapi logika  $X \cdot B + A \cdot B = Y$ . logika ini adalah sama dengan logika



(a) Rangkaian logis AND-OR



(b) Rangkaian logika NAND ekuivalen

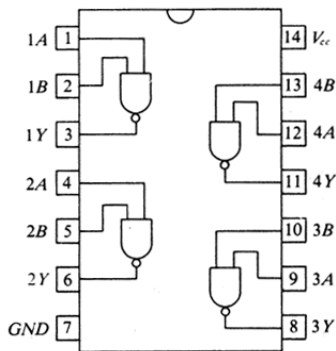
Gambar 9.68 Rangkaian Logika AND-OR dan rangkaian NAND ekuivalen

Bagaimana mengkombinasikan gerbang logika? Marilah kita lihat rangkaian logika pada Gambar 9.68 (a). Rangkaian ini disebut sebagai *gerbang-gerbang yang berpola AND-OR*. Gerbang AND dihubungkan ke dalam gerbang OR yang terakhir. Aljabar Boolean untuk rangkaian ini dinyatakan pada sebelah kanan sebagai  $A \cdot B + A \cdot \bar{B} = Y$ . Dalam menyusun rangkaian tersebut, kita membutuhkan tiga jenis gerbang yang berbeda (gerbang AND, gerbang OR, dan pembalik). Dari katalog pabrik, kita akan menjumpai bahwa tiga IC yang berbeda tersebut

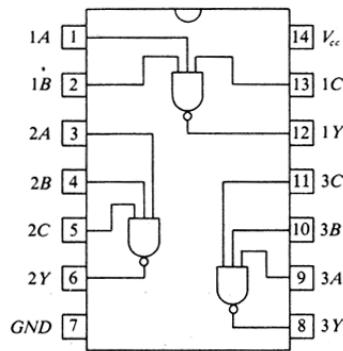
yang dihasilkan oleh rangkaian AND-OR pada Gambar 9.68(a). Ingat, bahwa gerbang yang mirip OR (gerbang 4), dengan gelembung pembalik pada masukannya, merupakan suatu gerbang NAND. Rangkaian pada Gambar 9.68 (b) adalah lebih sederhana karena semua gerbang merupakan gerbang NAND. Dan rangkaian tersebut jelas terlihat, bahwa untuk melengkapi logika NAND pada Gambar 9.68 (b), hanya dibutuhkan suatu IC (gerbang NAND 2-masukan lipat-empat). Untuk melengkapi rangkaian logika NAND, diperlukan jumlah IC yang lebih kecil.

daripada untuk melengkapi gerbang logika yang berpola AND-OR. Biasanya untuk mengubah logika

gerbang-OR 4. Gerbang NAND 1 berfungsi sebagai suatu inverter bila masukannya digabungkan satu sama



(a) Diagram kaki untuk suatu IC 7400



(b) Diagram kaki untuk suatu IC 7410

Gambar 9.69 Diagram kaki IC 7400 dan IC 7410

AND-OR ke logika NAND, mula-mula kita gambarkan poly AND-OR. Hal ini dapat dikerjakan dari aijabar Boolean. Diagram logika AND-OR akan muncul sama dengan diagram pada Gambar 9.68 (a). Selanjutnya, gerbang AND, gerbang OR, dan pembalik digantikan dengan gambar NAND. Pola logika NAND tersebut akan muncul sama dengan rangkaian pada Gambar 9.68 (b).

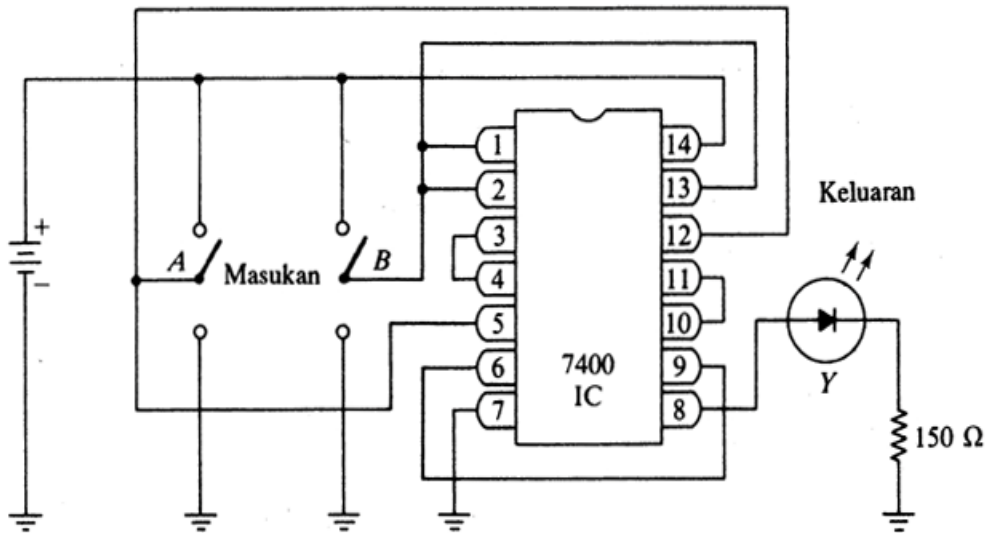
Pernyataan *mengapa* logika NAND-OR dapat digantikan dengan logika NAND, dijawab dengan petunjuk yang digambarkan pada Gambar 9.68 (b). Perlu kita perhatikan dua gelombang pembalik antara keluaran gerbang 2 dan masukan gerbang 4. *Dua gelombang pembalik tersebut saling meniadakan satu sama lain*. Hal ini akan menjadikan simbol NAND-OR menjadi seperti ditunjukkan pada Gambar 9.68 (a). Pada Gambar 9.68 (b) juga terjadi *pembalikan Banda* antara gerbang 3 dan 4.

Hal ini mengakibatkan gerbang-AND 3 memberi masukan ke

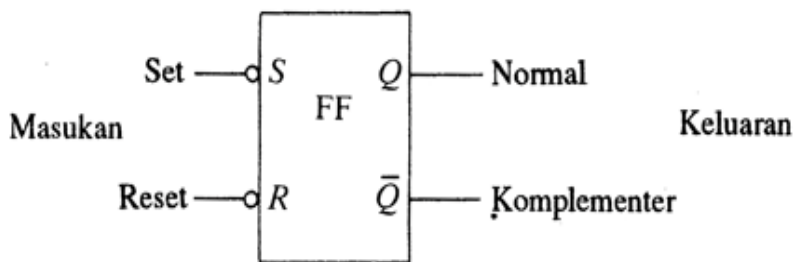
lain seperti dipaparkan pada Gambar 9.68 (b).

Bagaimana gerbang logika digunakan secara praktis? Gerbang-gerbang logika yang banyak digunakan, kebanyakan dikemas sebagai rangkain terpadu. Gambar 9.69 mengilustrasikan dua gerbang logika TTL yang dapat dibeli dalam bentuk IC. Diagram kaki dari IC7400 dipaparkan pada Gambar 9.69 (a). IC 7400 tersebut oleh pabrik digambarkan sebagai suatu IC gerbang NAND 2- masukan lipat-empat. Seperti biasanya, IC 7400 mempunyai hubungan days ( $V_{cc}$  dan GND).

Semua kaki lainnya adalah masukan dan keluaran dari empat gerbang NAND 2- masukan tersebut. Tiga gerbang NAND 3- masukan dimasukkan dalam rangkain terpadu yang berupa IC TTL 7410. Diagram kaki untuk IC 7410 ditunjukkan pada Gambar 9.69 (b). Peralatan ini oleh pabrik digambarkan sebagai IC gerbang NAND 3- masukan lipat-3. Gerbang NAND juga tersedia dengan



Gambar 9.70 Diagram rangkaian dari persoalan rangkaian logika



Gambar 9.71 Simbol logika untuk flip-flop RS

masukan lebih dari tiga kali. IC 1400 dan 7410 dari kelompok logika TTL, juga membuat jenis gerbang NAND, NOR, dan XOR dalam model ICs CMOS. Jenis gerbang NAND CMOS 74C00 mempunyai gerbang NAND quad 2 masukan, gerbang NAND 74C30 mempunyai 8 masukan dan ICs DIP gerbang NAND 4012 mempunyai awal 4 masukan. Beberapa CMOS gerbang NOR dalam bentuk IC DIP adalah 74C02 quad 2-masukan gerbang NOR dan 74C86 guard 2-masukan gerbang XOR dan 4030 guard 2-masukan

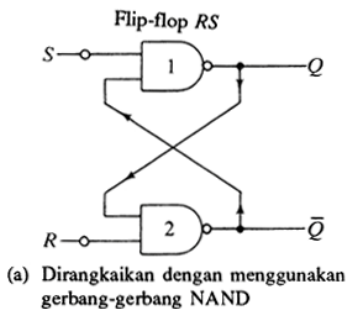
gerbang XOR. Perhatikan kedua jenis ICs CMOS seri 74C00 dan 4000. Kita harus mengingat bahwa ICs tersebut tanpa perantara khusus, *TTL dan ICs CMOS tidak kompatibel (sesuai)*.

Dari Gambar 9.70, tuliskan pernyataan boolean untuk rangkaian tersebut !

#### 9.5.2.2.3 Flip-Flop

Rangkaian logika dikelompokkan dalam dua kelompok besar. Kelompok-kelompok gerbang yang digambarkan demikian jauh, telah

dirangkakan sebagai rangkaian logika kombinasional. Pada bab ini akan kita perkenalkan jenis rangkaian yang sangat berguna, yaitu rangkaian logika sekuensial. Bentuk dasar dari logika kombinasional adalah gerbang logika. Bentuk dasar dari rangkaian logika sekuensial adalah rangkaian flip-flop. Rangkaian logika sekuensial sangat bermanfaat karena karakteristik-memorinya.



Gambar 9.72 Rangkaian ekuivalen flip-flop RS dengan menggunakan gerbang NAND dan tabel kebenarannya

Beberapa jenis flip-flop akan kita jelaskan dengan terperinci pada bab ini. Flip-flop juga disebut "kancing (*latch*)", "multivibrator", atau "biner". Dalam buku ini akan kita gunakan istilah "flip-flop". Flip-flop yang bermanfaat tersebut dapat dirangkakan dari gerbang logika seperti gerbang NAND, atau dapat juga dibeli dalam bentuk IC. Flip-flop diinterkonesikan untuk membentuk rangkaian logika sekuensial untuk penyimpanan, pewaktu, penghitungan, dan pengurutan (*sequencing*).

Kebanyakan flip-flop dasar adalah flip-flop RS. Simbol logika untuk flip-flop RS ditunjukkan pada Gambar 9.71. Simbol logika tersebut menunjukkan dua masukan, yang diberi label dengan set (*S*) dan reset (*R*) di sebelah kiri. Flip-flop RS pada simbol ini mempunyai masukan

RENDAH aktif yang ditunjukkan dengan gelembung-gelembung kecil pada masukan *S* dan *R*. Tidak seperti gerbang logika, flip-flop mempunyai dua keluaran komplementer. Keluaran tersebut diberi label dengan *Q* dan  $\bar{Q}$  (katakan "not *Q* atau *Q* not"). Keluaran *Q* dianggap merupakan keluaran "normal" dan paling sering digunakan.

Mode operasi	Masukan		Keluaran	
	<i>S</i>	<i>R</i>	<i>Q</i>	$\bar{Q}$
Larangan	0	0	1	1
Set	0	1	1	0
Reset	1	0	0	1
Tetap	1	1	tidak berubah	

(b) Tabel kebenaran (Diberikan oleh Signetics Corporatio

Keluaran lain *Q* merupakan komplemen dari keluaran *Q* dan disebut sebagai keluaran komplementer. Pada kondisi normal, keluaran-keluaran ini selalu merupakan komplementer. Dengan demikian, bila  $Q = 1$  maka  $\bar{Q} = 0$ , atau bila  $Q = 0$  maka  $\bar{Q} = 1$ .

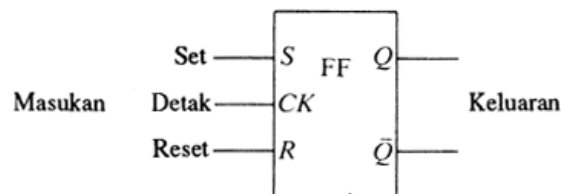
Flip-flop RS dapat disusun dari gerbang-gerbang logika. Flip-flop RS dirangkakan dari dua gerbang NAND pada Gambar 9.72 (a). Perhatikan balikan karakteristik dari keluaran satu gerbang NAND ke masukan gerbang lainnya. Sama halnya dengan gerbang logika, tabel kebenaran menentukan operasi flip-flop.

Baris 1 pada tabel kebenaran Gambar 9.72 (b) disebut keadaan terlarang dalam arti bahwa keadaan tersebut memungkinkan kedua

keluaran menjadi 1 atau TINGGI. Kondisi ini tidak digunakan pada flip-flop *RS*. Baris 2 pada tabel kebenaran tersebut menunjukkan kondisi set dari flip-flop. Di sini, level RENDAH atau logika 0 mengaktifkan masukan set (*S*). Logika 0 ini mengeset keluaran *Q* normal menjadi TINGGI atau 1, seperti ditunjukkan pada tabel kebenaran. Kondisi set ini akan terlihat bila kita menganalisa rangkaian NAND pada Gambar 9.72 (a).

keluaran *Q* menjadi 1. Begitu Pula, kondisi reset berarti di reset (clearing,) keluaran *Q* menjadi 0. Dengan demikian berarti kondisi operasi menunjuk pada keluaran normal.

Perhatikan bahwa keluaran komplementer ( $\bar{Q}$ ) adalah berlawanan dengan keluaran tersebut di atas. Oleh karena fungsi flip-flop memegang data sementara, maka flip-flop ini sering kali disebut kancing *RS*. Kancing *RS* dapat dirangkaikan



Gambar 9.73 Simbol logika untuk flip-flop *RS* yang berdetak

Logika 0 pada gerbang 1 membangkitkan 1 pada keluaran. Logika 1 ini dimasukkan kembali ke gerbang 2. Sekarang gerbang 2 mempunyai dua logika 1 yang dimasukkan pada masukannya, sehingga mendorong keluaran menjadi 0. Maka keluaran  $\bar{Q}$  menjadi 0 atau RENDAH. Baris 3 pada Gambar 9.72 (b) merupakan kondisi reset. Level RENDAH atau logika 0 mengaktifkan masukan reset tersebut. Hal ini akan mereset keluaran normal *Q* menjadi 0. Baris ke empat dari tabel tersebut menunjukkan kondisi tak-terbuka atau tetap (hold) dari flip-flop *RS*. Keluaran masih tetap seperti keadaan sebelum terjadi kondisi tetap. Jadi, tidak terdapat perubahan keluaran dari keadaan sebelumnya. Perlu kita perhatikan bahwa, bila tabel pada Gambar 9.72 menunjukkan kondisi set, hal ini berarti pengesetan

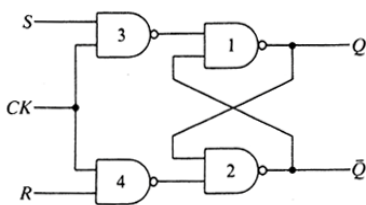
dari gerbang-gerbang atau dibeli dalam bentuk IC.

Flip-Flop *RS* berdetak, bisa dibentuk dari *latch RS* pada dasarnya merupakan suatu piranti asinkron (*asynchronous*). Peralatan seperti itu tidak beroperasi serempak dengan detak atau piranti pewaktu. Bila masukan (seperti masukan set) diaktifkan, maka keluaran normal segera diaktifkan seperti pada rangkaian logika kombinasional. Rangkaian-rangkaian penggerbangan dan kancing (*latch*) *RS* beroperasi secara asinkron. Flip-flop *RS* yang berdetak menambahkan suatu sifat sinkron yang berguna untuk kancing *RS*. Flip-flop *RS* yang berdetak akan beroperasi serempak dengan detak atau piranti pewaktu. Dengan kata lain, flip-flop tersebut beroperasi secara sinkron. Simbol logika untuk flip-flop *RS* yang berdetak diilustrasikan pada Gambar 9.73.

Flip-flop ini mempunyai masukan set (S) dan masukan detak (CK) tambahan. Flip-flop RS yang berdetak mempunyai keluaran normal (Q) dan keluaran komplementer ( $\bar{Q}$ ) seperti biasa. Flip-flop RS yang berdetak dapat dibuat dengan menggunakan gerbang-gerbang NAND. Gambar 9.74 (a) mengilustrasikan dua gerbang NAND yang ditambahkan pada Latch RS (flip-flop) untuk

ditekankan bahwa masukan-masukan R dan S adalah aktif selama keseluruhan waktu level pulsa detak TINGGI.

Level TINGGI dari pulsa detak ini dapat dianggap sebagai suatu pulsa pembuka. Tabel kebenaran pada Gambar 9.74 (b) menjelaskan dengan terperinci operasi flip-flop RS yang berdetak. Mode tetap dari operasi digambarkan pada baris 1 dari tabel



(a) Dirangkaikan dengan menggunakan gerbang NAND

Mode operasi	Masukan			Keluaran	
	CK	S	R	Q	$\bar{Q}$
Tetap		0	0	tidak berubah	
Reset		0	1	0	1
Set		1	0	1	0
Terlarang		1	1	1	1

= pulsa detak positif

(b) Tabel kebenaran

Gambar 9.74 Flip-flop RS yang berdetak

membentuk flip-flop RS yang berdetak. Gerbang NAND 3 dan 4 menambahkan sifat berdetak pada kancing RS tersebut. Perlu kita perhatikan bahwa adanya gerbang 1 dan 2 menyebabkan terbentuknya kancing RS atau flip-flop. Perhatikan juga bahwa, oleh karena pengaruh pembalikan gerbang 3 dan 4, maka sekarang masukan set (R) menjadi masukan TINGGI aktif.

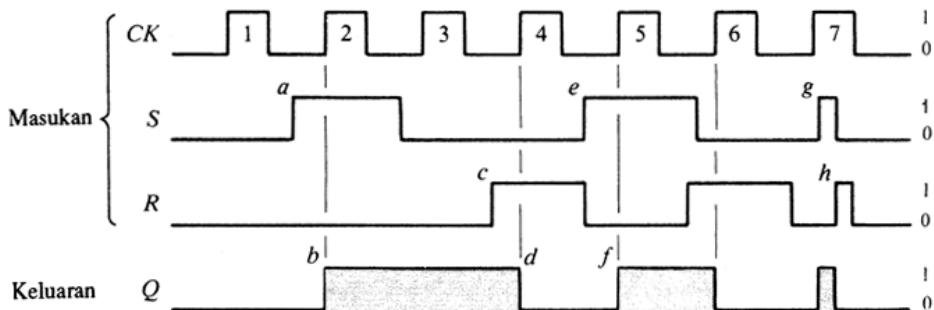
kebenaran tersebut. Bila pulsa detak datang pada masukan CK (dengan logis 0 pada masukan S dan R), maka keluaran *tidak berubah*. Keluaran-keluaran tersebut tetap sama seperti keadaan sebelum adanya pulsa detak.

Masukan detak (CK) memacu flip-flop (membuka flip-flop) bila pulsa detak menjadi TINGGI. Flip-flop RS yang berdetak dikatakan sebagai suatu peralatan *level-yang-dipacu*. Setiap kali pulsa detak menjadi TINGGI, maka informasi pada masukan data (R dan S) akan dipindahkan TINGGI, maka informasi pada masukan data (R dan S) akan dipindahkan ke keluaran. Perlu

Mode ini dapat juga digambarkan sebagai kondisi *tak terbuka* dari flip-flop. Baris 2 merupakan mode reset. Keluaran normal (Q) akan direset atau diklerkan menjadi 0 bila suatu level TINGGI mengaktifkan masukan R dan pulsa detak datang pada masukan CK. Perlu dicatat bahwa, dengan penempatan  $R = 1$  dan  $S = 0$  tidak akan segera mereset flip-flop tersebut. Flip-flop tersebut menanti sampai pula detak berubah dari RENDAH ke TINGGI dan selanjutnya flip-flop tersebut di reset. Unit ini beroperasi secara sinkron atau

serempak dengan detak. Baris 3 dari tabel kebenaran tersebut menggambarkan kondisi set dari flip-flop.

pada titik *b*, keluaran Q diset menjadi 1. Perhatikan bahwa, flip-flop menanti sampai pula detak 2 mulai berubah



Gambar 9.75 Diagram bentuk gelombang untuk flip-flop RS yang berdetak

Suatu level TINGGI (*high*) mengaktifkan masukan S (dengan  $R = 0$  dan pula detak TINGGI), sehingga akan mengeset keluaran Q menjadi 1. Baris 4 dari tabel kebenaran ini merupakan kombinasi yang terlarang (semua masukan 1), dan hal ini tidak digunakan karena untuk mengendalikan kedua keluaran TINGGI.

*Diagram waktu* atau *bentuk gelombang*, banyak dipakai dan sangat berguna untuk pekerjaan dengan flip-flop dan rangkaian logika sekuensial. Gambar 9.75 mengilustrasikan suatu diagram waktu untuk flip-flop RS yang berdetak. Tiga baris bagian atas menyatakan sinyal biner pada masukan detak, set, dan reset. Dibagian bawah hanya ditunjukkan keluaran tunggal (Q). Kita mulai dari sebelah kin, pula detak 1 tiba, tetapi tidak berpengaruh terhadap Q karena masukan S dan R berada dalam mode tetap. Maka keluaran Q tetap berada pada logis 0.

Pada titik *a* dari diagram waktu tersebut, masukan set diaktifkan menjadi TINGGI. Sesaat sesudah itu,

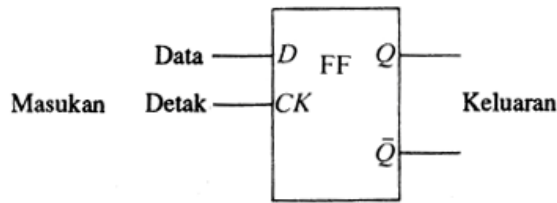
dari RENDAH ke TINGGI sebelum keluaran Q diset. Pulsa 3 tidak akan mengubah keluaran karena masukan-masukan (R dan S) berada dalam mode tetap. Pada titik *c*, masukan reset diaktifkan dengan level TINGGI. Sesaat kemudian, pada titik *d*, keluaran Q direset atau diubah menjadi 0. Sekali lagi, hal ini terjadi pada transisi pulsa detak RENDAH-ke-TINGGI. Pada titik *e* dari diagram waktu tersebut, masukan set diaktifkan sehingga akan mengeset keluaran Q menjadi logika 1 pada titik *f*. Masukan S dinonaktifkan dan R diaktifkan sebelum pula 6, yang menyebabkan keluaran Q menjadi RENDAH atau kondisi reset. Pulsa 7 menunjukkan bahwa selama detaknya TINGGI maka keluaran Q mengikuti masukan S dan R. Pada titik *g* dari diagram waktu Gambar 9.76 tersebut, masukan set (S) menjadi TINGGI (*high*) dan keluaran Q mengikuti menjadi TINGGI (*high*) pada titik *h*.

Hal ini menyebabkan keluaran Q reset atau menjadi RENDAH. Kemudian, masukan R menjadi RENDAH. Kemudian masukan R



kembali menjadi RENDAH, dan akhirnya pulsa detak 7 mengakhirinya dengan transisi TINGGI-ke-RENDAH.

sebelah kanan simbol tersebut. Flip-flop *D* sering kali disebut sebagai *flip-flop*

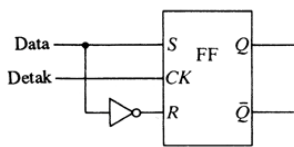


Gambar 9.76 Simbol logika untuk flip-flop D

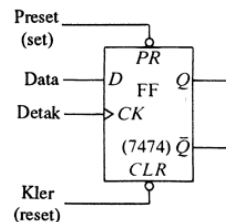
Selama pulsa detak, keluaran *di-set* menjadi TINGGI dan kemudian *reset* menjadi RENDAH.

Perhatikan bahwa antara pulsa 5 dan 6 terjadi keadaan dimana baik masukan *S* maupun *R* berada pada logika 1. Kondisi masukan *R* dan *S* yang kedua-duanya berlevel TINGGI ini, umumnya dianggap sebagai keadaan terlarang untuk flip-flop tersebut. Dalam kasus ini, masukan *R* dan *S* kedua-duanya diperbolehkan berlevel TINGGI karena pulsa detak adalah RENDAH, sehingga flip-flop tidak diaktifkan.

*tunda*. Nama ini menggambarkan dengan tepat operasi unit ini. Apapun bentuk masukan pada masukan data (*D*), masukan tersebut akan *tertunda selama satu pulsadetak* untuk mencapai keluaran normal (*Q*). Data dipindahkan ke keluaran pada transisi pulsa detak RENDAH-ke-TINGGI. Flip-flop *RS* yang berdetak dapat dirubah menjadi flip-flop *D* dengan menambahkan suatu pembalik. Pengubahan ini ditunjukkan pada diagram Gambar 9.77 (a). Kita perhatikan bahwa masukan *R* pada flip-flop *RS* yang berdetak tersebut



(a) FLip-flop *D* yang dirangkaikan dari flip-flop *RS* yang berdetak



(b) Simbol logika untuk flip-flop *D* 7474 dengan masukan asinkron

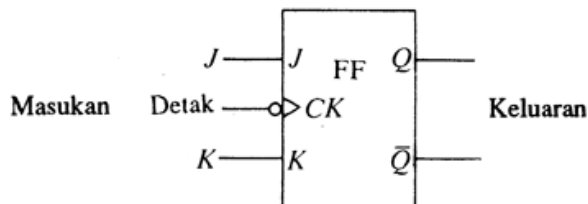
Gambar 9.77 Flip-flop *D* terangkai dr flip-flop *RS* yg berdetak dan Simbol logika flip-flop *D*7474

Simbol flip-Flop *D* secara umum diilustrasikan seperti terlihat pada Gambar 9.76. *Flip-flop D* tersebut hanya mempunyai masukan *data* tunggal (*D*) dan masukan (*CK*). Keluaran *Q* dan *Q* ditunjukkan pada

telah dibalik. Flip-flop *D* komersial ditunjukkan pada Gambar 9.77 (b). Flip-flop *D* pada Gambar 7-9b ini merupakan suatu peralatan TTL, yang oleh pabriknya disebut sebagai IC 7474. Simbol logika untuk flip-flop

Mode operasi	Masukan				Keluaran	
	Asinkron		Sinkron		Q	Q̄
	PR	CLR	CK	D		
Set asinkron	0	1	X	X	1	0
Reset asinkron	1	0	X	X	0	1
Terlarang	0	0	X	X	1	1
Set	1	1	↑	1	1	0
Reset	1	1	↑	0	0	1

Gambar 9.78 Model tabel kebenaran untuk flip-flop D 7474



Gambar 9.79 Simbol logika untuk flip-flop JK

D 7474 tersebut menunjukkan masukan  $D$  dan  $CK$  yang biasa. Masukan-masukan ini disebut masukan *sinkron* karena beroperasi serempak dengan detak. Dua masukan ekstra adalah masukan-masukan asinkron yang beroperasi seperti flip-flop  $RS$  yang telah dibahas sebelumnya.

Masukan-masukan asinkron tersebut dilabelkan dengan reset ( $PR$ ) dan Clear ( $CLR$ ) dapat diaktifkan oleh level RENDAH, seperti ditunjukkan oleh gelembung kecil pada simbol logika. Bila masukan clear pada flip-flop  $D$  diaktifkan, maka keluaran  $Q$  direset atau dikierkan menjadi 0. Masukan-masukan asinkron *mengesampingkan* masukan sinkron pada flip-flop  $D$ . Tabel kebenaran untuk flip-flop  $D$  7474 dipaparkan pada Gambar 9.78. Mode operasi

diberikan di sebelah kiri dan tabel kebenaran di sebelah kanan.

Tiga baris pertama merupakan operasi asinkron (masukan preset dan kler). Baris 1 menunjukkan masukan preset ( $PR$ ) yang diaktifkan dengan suatu level RENDAH. Hal ini akan mengeset keluaran  $Q$  menjadi 1. Kita perhatikan  $X$  di bawah masukan sinkron ( $CK$  dan  $D$ ).  $X$  berarti bahwa masukan ini adalah tidak relevan karena masukan asinkron menolaknya. Baris 2 menunjukkan masukan  $CLR$  yang diaktifkan dengan suatu level RENDAH. Hal ini menyebabkan keluaran  $Q$  direset atau dikierkan menjadi 0.

Baris 3 menunjukkan masukan asinkron yang terlarang (baik  $PR$  maupun  $CLR$  berada pada logika 0). Masukan sinkron ( $D$  dan  $CK$ ) akan beroperasi bila semua masukan

Mode operasi	Masukan			Keluaran	
	<i>CK</i>	<i>J</i>	<i>K</i>	<i>Q</i>	<i>Q̄</i>
Tetap		0	0	tidak berubah	
Reset		0	1	0	1
Set		1	0	1	0
Togel		1	1	keadaan berlawanan	

Gambar 9.80 Model tabel kebenaran untuk flip-flop JK yang dipacu pulsa

asinkron terbuka ( $PR = 1$ ,  $CLR = 1$ ). Baris 4 menunjukkan logika 1 pada masukan data ( $D$ ) dan pulsa detak yang naik (ditunjukkan dengan anak panah ke atas). Logika 1 pada masukan  $D$  ini dipindahkan ke keluaran  $Q$  pada pulsa detak tersebut. Baris 5 ( $D$ ) yang dipindahkan ke keluaran  $Q$  pada transisi detak RENDAH-ke-TINGGI. Bila flip-flop  $D$  tersebut tidak mempunyai masukan asinkron, maka hanya diperlukan dua baris bagian bawah dari tabel kebenaran pada Gambar 9.78. Flip-flop  $D$  banyak digunakan dalam penyimpanan data.

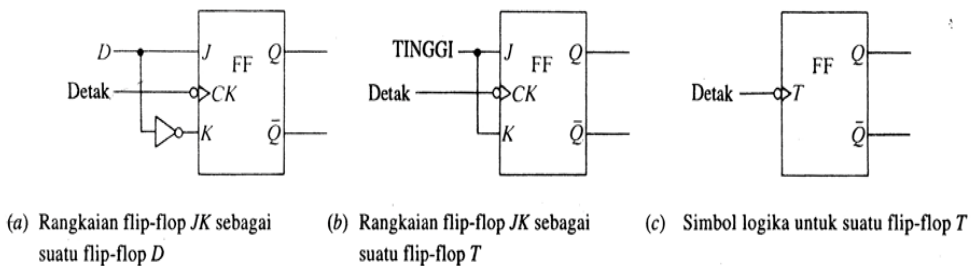
Marilah kita simak simbol flip-flop  $D$  pada Gambar 9.76 dan 9.77 (b). Perhatikan bahwa masukan detak ( $CK$ ) pada Gambar 9.77 (b) mempunyai tanda kecil  $>$  di dalam simbol, yang berarti bahwa flip-flop tersebut merupakan peralatan yang dipacu-di-ujung. Flip-flop yang dipacu-di-ujung memindahkan data dari masukan  $D$  ke keluaran  $Q$  pada transisi pulsa detak RENDAH-ke-TINGGI.

Pada pemacuan ujung, yang memindahkan data adalah perubahan detak dari RENDAH ke TINGGI. Pada flip-flop yang dipacu-di-ujung, sekali

pulsa detak berlevel TINGGI maka perubahan masukan  $D$  tidak akan berpengaruh terhadap keluaran. Gambar 9.76 dan 9.77 (a) mengilustrasikan suatu flip-flop  $D$  yang dipacu level (berlawanan dengan dipacu ujung). Tidak adanya tanda kecil  $>$  pada masukan detak di dalam simbol ini, menyatakan bahwa peralatan tersebut merupakan peralatan yang dipacu-level. Pada flip-flop yang dipacu-level, level tegangan tertentu akan menyebabkan data pada masukan  $D$  berpindah ke keluaran  $Q$ .

Persoalan yang menyangkut peralatan yang dipacu-level adalah, keluaran yang mengikuti masukan bila masukan berubah pada saat pulsa detak berlevel TINGGI. Pemacuan level atau pendekatan dapat menjadi persoalan bila data masukan berubah pada saat pulsa detak berlevel TINGGI.

Simbol flip-flop  $JK$  ditunjukkan seperti terlihat pada Gambar 9.79. Piranti ini dapat dianggap sebagai flip-flop universal. Dikatakan universal karena flip-flop jenis lain dapat dibuat dari flip-flop  $JK$  ini. Simbol logika  $JK$  seperti terlihat pada Gambar 9.79 mengilustrasikan mengenai tiga



Gambar 9.81 Rangkaian flip-flop JK dan simbol logika untuk flip-flop T

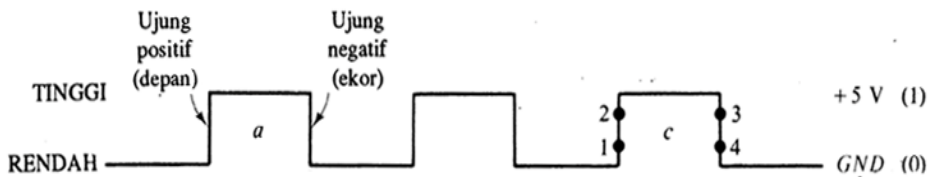
masukan sinkron ( $J$ ,  $K$ , dan  $CK$ ). Masukan  $J$  dan  $K$  merupakan masukan data, dan masukan detak memindahkan data dari masukan ke keluaran. Simbol logika pada Gambar 9.79 juga menunjukkan keluaran normal ( $Q$ ) dan keluaran komplementer ( $\bar{Q}$ ). Tabel kebenaran untuk flip-flop  $JK$  dilukiskan pada Gambar 9.80. Mode operasi diberikan di sebelah kiri dan tabel kebenaran di sebelah kanan. Baris 1 dari tabel kebenaran tersebut menunjukkan kondisi tetap atau kondisi terbuka. Perhatikan bahwa semua data masuk ( $J$  dan  $K$ ) adalah RENDAH. Kondisi reset atau kler dari flip-flop tersebut ditunjukkan pada baris 2 dari tabel kebenaran. Bila  $J = 0$  dan  $K = 1$  serta pulsa detak datang pada masukan  $CK$ , maka flip-flop tersebut direset ( $Q = 0$ ). Baris 3 menunjukkan kondisi set dari flip-flop  $JK$ . Bila  $J = 1$  dan  $K = 0$  serta terdapat pulsa detak, maka keluaran  $Q$  diset menjadi 1. Baris 4 mengilustrasikan kondisi yang sangat berguna dari flip-flop  $JK$ . Kondisi ini disebut posisi *togel* (toggle). Bila masukan  $J$  dan  $K$  kedua-duanya TINGGI, maka keluaran akan berlawanan dengan keadaan pada waktu pulsa tiba pada masukan  $CK$ . Dengan pulsa detak yang berulang, keluaran  $Q$  dapat menjadi RENDAH,

TINGGI, RENDAH, TINGGI, RENDAH, dan sebagainya. Gagasan RENDAH-TINGGI-RENDAH-TINGGI ini disebut *pentogelan*. Istilah "pentogelan" berasal dari kenyataan sifat HIDUP-MATI dari saklar togel.

Marilah kita perhatikan tabel kebenaran pada Gambar 9.80. Keseluruhan pulsa detak ditunjukkan di bawah kolom masukan detak ( $JK$ ) dari tabel kebenaran tersebut. Kebanyakan flip-flop  $JK$  adalah *pulsa trigger*. Pada flip-flop seperti itu, diperlukan keseluruhan pulsa untuk memindahkan data dari masukan ke keluaran. Dengan adanya masukan detak pada tabel kebenaran tersebut, jelas bahwa flip-flop  $JK$  merupakan suatu flip-flop sinkron.

Flip-flop  $JK$  dianggap merupakan flip-flop universal. Gambar 9.81 (a) mengilustrasikan bagaimana flip-flop  $JK$  dan pembalik pada dirangkai untuk membentuk suatu flip-flop  $D$ . Perhatikan masukan  $D$  tunggal pada ujung sebelah kiri dan masukan detak. Flip-flop  $D$  yang dirangkai ini akan memacu pada transisi pulsa detak TINGGI-ke-RENDAH, seperti ditunjukkan oleh gelembung pada masukan  $CK$ .

*Flip-flop togel* yang banyak digunakan (flip-flop jenis-T) ditunjukkan dalam rangkaian Gambar



Gambar 9.82 Pulsa-pulsa detak

9.81 (b). Gambar tersebut menunjukkan flipflop *JK* yang digunakan dalam mode togel. Perhatikan bahwa masukan *J* dan *K* digabungkan sedemikian rupa sehingga menjadi level TINGGI, dan detak dimasukkan pada masukan *CK*. Begitu pulsa detak yang berulang dimasukkan pada masukan *CK*, keluaran akan mentogel secara sederhana.

Operasi togel banyak digunakan dalam rangkaian logika sekuensial. Oleh karena penggunaannya yang luas, maka kadang-kadang digunakan simbol khusus untuk flipflop togel (jenis-*T*) tersebut. Gambar 9.81 (c) mengilustrasikan simbol logika untuk flipflop togel. Masukan tunggal (yang dilabelkan dengan *T*) merupakan masukan detak. Keluaran *Q* dan  $\bar{Q}$  ditunjukkan di sebelah kanan dari simbol tersebut. Flipflop *T* hanya mempunyai mode operasi togel.

Flipflop *JK* komersial dapat juga mempunyai sifat masukan asinkron, disamping masukan *J*, *K*, dan *CK* yang normal. Khususnya, flipflop tersebut dapat mempunyai masukan preset (*PR*) untuk pengesetan secara asinkron keluaran *Q* menjadi 1. Flipflop ini dapat juga mempunyai masukan kler (*CLR*) untuk mengklerkan atau mereset keluaran *Q* menjadi 0. Kebanyakan flipflop *JK* adalah pulsa trigger; namun flipflop

ini dapat juga dibeli berupa unit yang dipacu-ujung (edge-triggered unit).

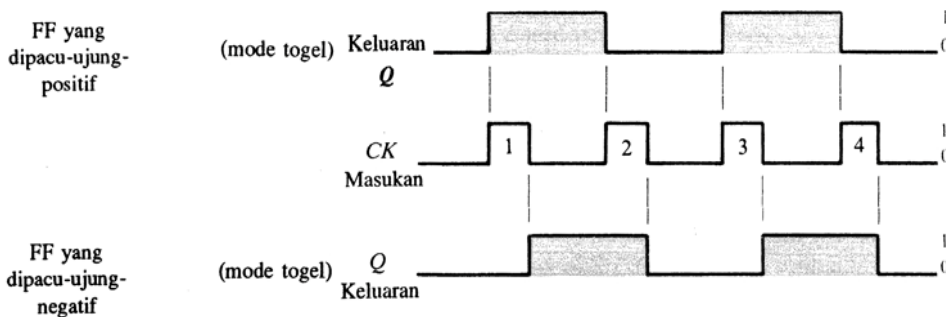
Flipflop adalah dasar pembentuk blok dari urutan rangkaian logika. Oleh karena, itu, pembuatan IC jenisnya bervariasi dalam dua flipflop dalam kelompok TTL dan CMOS. Pada umumnya flipflop CMOS adalah jenis 7428 8-bit alamat Latch, 40125 quad flipflop D, dan 14C76 dual flipflop J-K dengan Clear dan preset.

Pemacu diperlukan pada flipflop. Kebanyakan perlengkapan digital yang kompleks beroperasi sebagai suatu sistem sekuensial sinkron. Hal ini menyatakan bahwa suatu sinyal detak master dikirimkan kepada semua bagian sistem tersebut untuk mengkoordinasikan operasinya. Deretan pulsa-detak khusus ditunjukkan pada Gambar 9.82. Ingat bahwa jarak horisontal pada bentuk gelombang tersebut adalah *waktu* dan jarak vertikal adalah *tegangan*. Pulsa detak yang dipaparkan pada gambar ini adalah untuk peralatan TTL karena adanya tegangan +5 V dan *GND*. Rangkaian digital yang lain menggunakan detak tetapi tegangannya dapat berbeda, bisa dilihat dari datasheet komponen.

Kita mulai dan sebelah kiri pada bentuk gelombang pada Gambar 9.82. Mula-mula pulsa berada pada tegangan *GND* atau level RENDAH. Keadaan ini disebut juga suatu logis

0. Pulsa *a* menunjukkan *ujung depan* (*ujung positif*) dari bentuk gelombang yang berubah dari tegangan *GND* ke

*flops*). Operasi flip-flop yang dipacu-ujung-negatif 'ditunjukkan pada dua bentuk gelombang bagian bawah dari



Gambar 9.83 Pemacuan flip-flop ujung positif dan ujung negatif

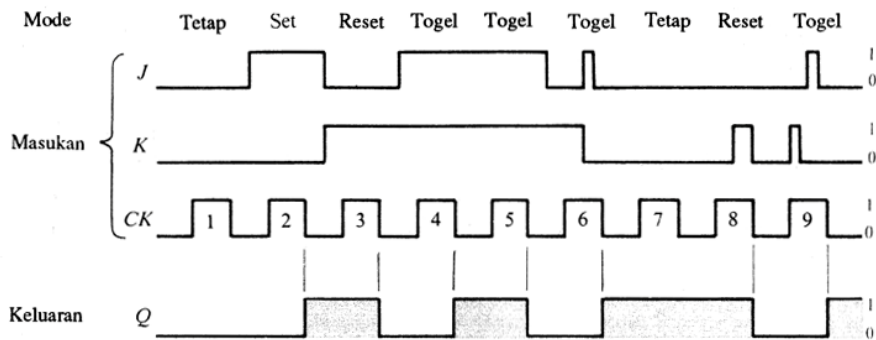
+ 5 V. Ujung bentuk gelombang ini dapat juga disebut ujung RENDAH-ke-TINGGI (R-ke-T) dari bentuk gelombang. Pada sisi kanan dari pulsa *a*, bentuk gelombang turun dari + 5 V ke tegangan *GND*. Ujung ini disebut ujung TINGGI-ke-RENDAH (T-ke-R) dari pulsa detak. Hal ini juga disebut ujung yang *menuju-negatif* atau ujung *ekor* dari pulsa detak.

Beberapa flip-flop memindahkan data dari masukan ke keluaran pada ujung positif 1 (depan) dari pulsa detak. Flip-flop ini disebut sebagai flip-flop yang dipacu ujung positif (*positive-edge-triggered flip-flop*). Contoh flip-flop seperti itu diilustrasikan pada Gambar 9.83. Masukan detak ditunjukkan dengan bentuk gelombang di tengah. Bentuk gelombang bagian atas menunjukkan keluaran *Q*, bila flip-flop yang dipacu-ujung-positif tersebut berada pada mode togel. Perhatikan bahwa masing-masing ujung depan (ujung menuju-positif) dari detak tersebut mentogel flip-flop.

Flip-flop lain dikelompokkan sebagai *flip-flop yang dipacu-ujung-negatif* (*negative-edge-triggered flip-*

Gambar 9.83. Bentuk gelombang bagian tengah merupakan masukan detak. Bentuk gelombang bagian bawah adalah keluaran *Q* bila flip-flop berada pada mode togel. Perhatikan bahwa, flip-flop ini mentogel menjadi keadaannya yang berlawanan, hanya pada ujung ekor (ujung menuju-negatif) dari pulsa detak. hal ini penting untuk menentukan perbedaan waktu dari flip-flop yang dipacu-ujung-positif dan flip-flop yang dipacu-ujung-negatif pada Gambar 9.83. Perbedaan waktu ini sangat penting dalam beberapa aplikasi. Banyak flip-flop *JK* yang berupa unit yang dipacu-pulsa. Peralatan yang dipacu-pulsa ini merupakan *flip-flop JK master-budak* (*master-slave JK flip-flops*).

Flip-flop *JK* master-budak sebenarnya merupakan beberapa gerbang dan flip-flop yang dirangkaikan satu sama lain dengan menggunakan pulsa-pulsa detak untuk memindahkan data dari masukan ke keluaran. Pulsa *c* pada Gambar 9.82 akan digunakan untuk membantu menjelaskan bagaimana pemacuan pulsa bekerja dengan suatu flip-flop *JK* master-budak.



Gambar 9.84 Diagram bentuk gelombang untuk suatu flip-flop JK

Kejadian dibawah ini terjadi selama urutan pemacuan-pulsa pada titik-titik yang diberi nomor pada Gambar 9.82 adalah :

1. Masukan dan keluaran dari flip-flop diisolasi.
2. Data dimasukkan dari masukan  $J$  dan  $K$ , tetapi tidak (dipindahkan ke keluaran).
3. Masukan  $J$  dan  $K$  terbuka.
4. Data yang dimasukkan sebelumnya, dari  $J$  dan  $K$ , dipindahkan ke keluaran.

Perhatikan bahwa data sebenarnya muncul pada keluaran titik 4 (ujung ekor) pada bentuk gelombang Gambar 9.82. Simbol logika untuk flip-flop yang dipacu-pulsa mempunyai gelembung kecil yang diletakkan pada masukan detak ( $CK$ ) untuk menunjukkan bahwa pemindahan data yang sebenarnya ke keluaran terjadi pada transisi pulsa detak T-ke-R.

Bentuk gelombang pada Gambar 9.84 akan membantu memahami operasi flip-flop  $JK$  master-budak dan pemacuan pulsa. Kita mulai dari sebelah kiri pada diagram bentuk gelombang tersebut. Tiga bentuk ge-

lombang bagian atas adalah masukan sinkron  $J$ ,  $K$ , dan  $CK$ . Baris bagian atas menggambarkan mode operasi selama pulsa detak. Baris bagian bawah pada Gambar 9.84 merupakan keluaran yang dihasilkan oleh flip-flop  $JK$  pada keluaran  $Q$ .

Marilah kita amati pulsa detak ( $CK$ ) pada Gambar 9.84. Baik masukan  $J$  maupun masukan  $K$  berlevel RENDAH. Hal ini merupakan kondisi tetap, sehingga keluaran  $Q$  tetap diam pada logis 0 seperti sebelum adanya pulsa 1. Sekarang, marilah kita lihat pulsa detak ( $CK$ ) 2. Masukkan  $J$  dan  $K$  berada pada mode set ( $J = 1, K = 0$ ). Pada ujung ekor pulsa 2, keluaran  $Q$  menuju logis 1 atau TINGGI. Pulsa 3 datang pada waktu masukan-masukan berada dalam keadaan mode reset ( $J = 0, K = 1$ ). Pada ujung ekor dari pulsa detak 3, keluaran  $Q$  direset atau diklerkan menjadi 0. Pulsa 4 datang pada saat masukan-masukan dalam keadaan mode togel ( $J = 1, K = 1$ ). Pada ujung ekor dari pulsa detak 4, keluaran  $Q$  mentogel ke logis 1 atau TINGGI. Pulsa 5 datang pada waktu masukan-masukan dalam keadaan mode togel kembali. Pada ujung ekor

dari pulsa 5, keluaran Q mentogel ke logis 0 atau RENDAH.

Pulsa detak 6 pada Gambar 9.84 akan menunjukkan sifat yang luar biasa dari flip-flop JK master-budak. Perhatikan bahwa pada ujung depan dari pulsa 6, masukkan  $K = 1$  dan  $J = 0$ . Kemudian, pada waktu pulsa 6 sedang berlevel TINGGI, masukkan  $K$  berubah dari 1 ke 0, sedangkan masukan  $J$  dari 0 ke 1 dan ke 0 lagi. Pada ujung ekor dari pulsa 6 kedua masukan dan  $K$ ) berlevel RENDAH. Sepintas lalu, kondisi seperti ini seolah-olah seperti kondisi tetap. Namun, inilah keanehan yang terlihat pada flip-flop JK master-budak; ternyata flip-flop ini masih mentogel level TINGGI. Flip-flop JK master-budak *mengingat setiap atau semua masukan TINGGI pada waktu pulsa detak sedang berlevel TINGGI*. Selama pulsa 6, baik masukan  $J$  maupun  $K$  berlevel TINGGI untuk sesaat, pada waktu masukan detak sedang berlevel TINGGI. Maka flip-flop tersebut tetap menganggap hal ini sebagai kondisi togel.

Selanjutnya, perhatikan pulsa detak 76 pada Gambar 9.84. Pulsa 7 muncul pada waktu masukan JK berada pada mode tetap ( $J = 0, K = 0$ ). Keluaran Q tetap berada pada keadaannya yang sekarang (Logis 1). Selama pulsa 8 muncul, masukan  $K$  berlevel TINGGI untuk sesaat dan masukan  $J$  berada pada logis 0. Flip-flop menerima hal ini sebagai mode reset. Maka pada ujung ekor dari pulsa detak 8, keluaran Q akan reset menjadi 0.

Sekarang, perhatikan pulsa detak 9 pada Gambar 9.84. Pada waktu muncul ujung positif pulsa detak 9 dari flip-flop JK master-budak tersebut, baik masukan  $J$  maupun  $K$  kedua-duanya berlevel RENDAH.

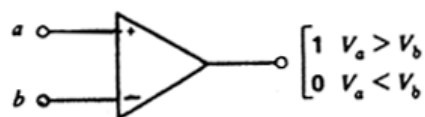
Selama pulsa 9 berlevel TINGGI, masukan  $K$  menjadi TINGGI untuk sesaat, dan kemudian menyusul masukan  $C^y$  juga menjadi TINGGI pada saat yang sama. Pada ujung ekor dari pulsa detak 9, kedua masukan tersebut ( $J$  dan  $K$ ) berlevel RENDAH. Flip-flop menganggap hal ini sebagai mode togel. Maka keluaran Q berubah keadaan dari logika 0 ke 1.

Sebagai catatan, perlu kita ketahui bahwa tidak semua flip-flop JK berjenis master-budak. Beberapa flip-flop JK adalah dipacu-ujung. Manual data dari pabrik akan menentukan apakah flip-flop tersebut dipacu-ujung atau dipacu-pulsa.

#### 9.5.2.2.4 Konverter

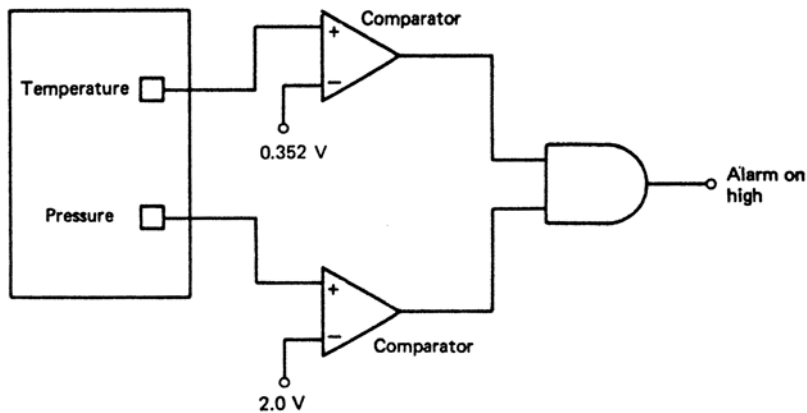
Komponen digital yang sangat penting dalam teknologi sistem kontrol salah satunya adalah mengubah informasi digital ke analog dan sebaliknya. Pengukuran yang sangat baik dari variabel proses bisa dilihat dari peralatan yang mengubah informasi dari suatu variabel sinyal tegangan elektrik.

Untuk menghubungkan sinyal ini dengan sebuah komputer atau rangkaian digital logic, yang pertama diperlukan adalah memakai suatu komponen yang mengkonversi dari sinyal analog menjadi digital, yang disebut *Analog to Digital Converter*



Gambar 9.85 Komparator dasar



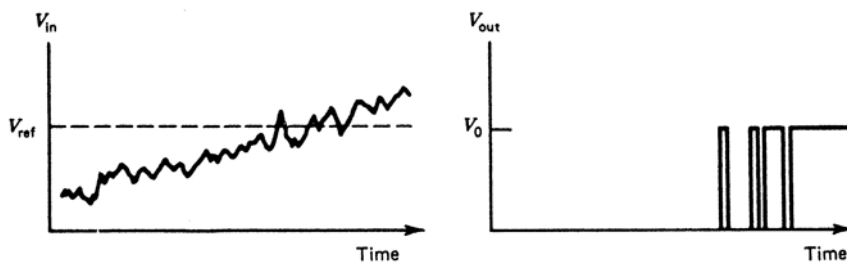


Gambar 9.86 Contoh aplikasi rangkaian komparator

(ADC). Spesifikasi dari konversi ini harus bisa diketahui dengan baik secara detail, yaitu mengetahui hubungan yang ada antara sinyal analog dan digital. Seringkali terjadi situasi yang berkebalikan, dimana sinyal digital yang diperlukan untuk menggerakkan peralatan yang analog. Dalam kasus ini, diperlukan suatu komponen elektronika yang mampu mengubah sinyal digital menjadi sinyal analog, yang disebut *Digital to Analog Converter* (DAC).

Komponen elektronika yang merupakan bentuk mendasar

yang banyak digunakan dalam suatu komunikasi antara analog dan digital adalah menggunakan suatu komponen (biasa berbentuk *IC-Integrated Circuit*) disebut komparator. Komponen ini secara skematis ditunjukkan pada Gambar 9.85, membandingkan dua tegangan analog secara sederhana pada terminal inputnya. Tergantung tegangan mana yang lebih besar, output akan menjadi sinyal digital 1 (*high*) atau 0 (*low*). Komparator secara luas dipakai sebagai sinyal alarm atau sinyal *trigger* atau sinyal



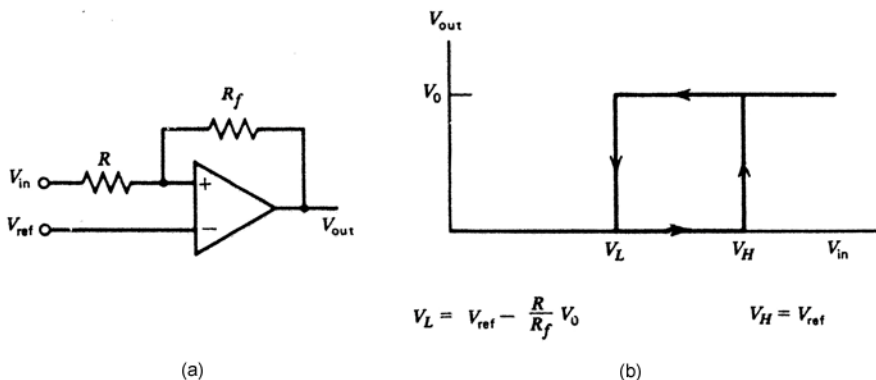
Gambar 9.87 Output keluaran akan bergoncang ketika sinyal mempunyai noise didaerah level tegangan referensi

picu untuk suatu komputer atau sistem pemrosesan digital. Elemen ini juga terintegrasi menjadi bagian dari *analog to digital (A/D)* dan *digital to analog (D/A) converter*.

Salah satu tegangan dari input komparator,  $V_a$  or  $V_b$  pada Gambar 9.85 akan menjadi input yang bervariasi (input variabel) dan input lainnya yang tidak berubah disebut tegangan referensi. Nilai dari tegangan referensi ini dihitung dari spesifikasi masalah kemudian diaplikasikan secara tepat pada terminal input komparator. Sebagai ilustrasi, ditunjukkan dalam Gambar 9.86. Tegangan referensi disediakan dari rangkaian pembagi yang diambil dari sumber tegangan yang tersedia.

Output komparator yang berubah-ubah ini bisa menyebabkan masalah pada peralatan yang didesain untuk menafsirkan sinyal output dari komparator.

Masalah ini seringkali dapat diatasi oleh pemberian histerisis pada sekitar daerah level referensi dimana sering terjadi perubahan output. Suatu waktu, komparator mendapatkan pemicu dengan level high, level referensi ini secara otomatis akan berkurang sehingga sinyal yang berkurang drastis pada beberapa nilai tertentu di bawah level referensi yang lalu, sebelum komparator menghasilkan kondisi output yang low. Ada banyak cara dengan histerisis bisa disediakan, te-



Gambar 9.88 Komparator dengan histerisis.

Komparator dengan histerisis seringkali diperlukan ketika memakai rangkaian komparator, seringkali terjadi masalah jika tegangan sinyal mempunyai noise. Noise adalah sinyal yang tidak diharapkan. Output komparator kemungkinan bisa "bergoncang" kembali dan seterusnya antara high dan low. Di daerah level tegangan referensi. Efek ini ditunjukkan dalam Gambar 9.87.

tapi gambar 9.88 (a) menunjukkan salah satu dari teknik tersebut. Resistor *feedback*  $R_f$  dipasang antara output dan salah satu input komparator dan input dipisah dari sinyal oleh resistor yang lain,  $R$ . Untuk kondisi  $R_f \gg R$ , respon dari komparator ditunjukkan dalam Gambar 9.88 (b). Kondisi dimana output akan menjadi high ( $V_0$ ) didefinisikan oleh kondisi :

$$V_{in} \geq V_{ref}$$

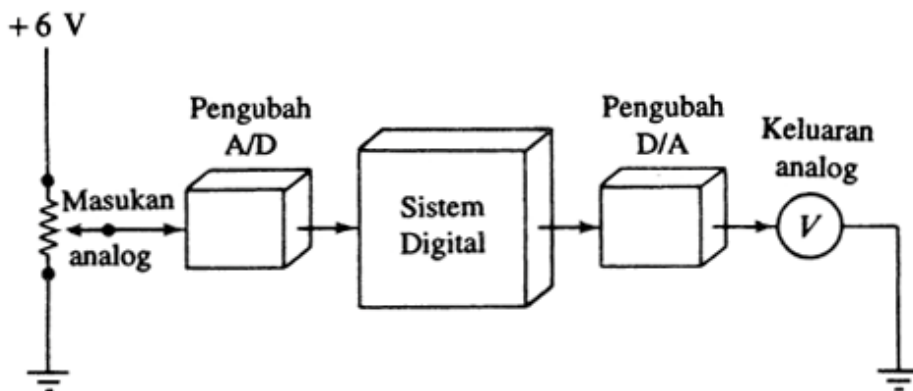
Suatu ketika, menjadi high, kondisi dari output akan kembali menjadi low (0 V) diberikan oleh hubungan :

$$V_{in} \leq V_{ref} - (R/R_f)V_o$$

Histerisis diberikan oleh  $(R/R_f)V_o$  dan hal ini bisa berubah dengan mengatur harga resistor yang ada, selama hubungannya masih memenuhi. Respon komparator ditunjukkan pada Gambar 9.88 (b). Anak panah menindikasikan penambahan atau pengurangan tegangan input.

lain, pengubah tersebut merumuskan dengan baik masukan-masukan dan keluaran-keluaran yang diharapkan dari sistem. Tabel kebenaran pada Gambar 9.90 (b) merinci beberapa kemungkinan masukan dan keluaran untuk pengubah D/A.

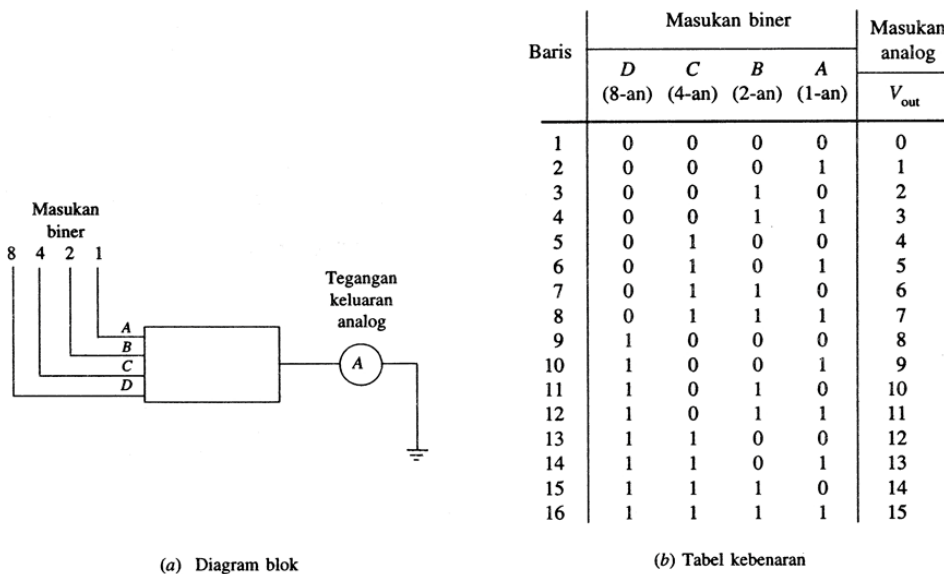
Marilah kita tinjau tabel kebenaran pada Gambar 9.90 (b) untuk pengubah D/A tersebut. Bila masing-masing masukan RENDAH, maka tegangan keluaran ( $V_{out}$ ) menjadi 0 V seperti dirumuskan pada baris 1 dari tabel tersebut. Baris 2 sekedar memaparkan masukan 1 (A) yang diaktifkan oleh suatu level TINGGI (level TINGGI ini kira-kira



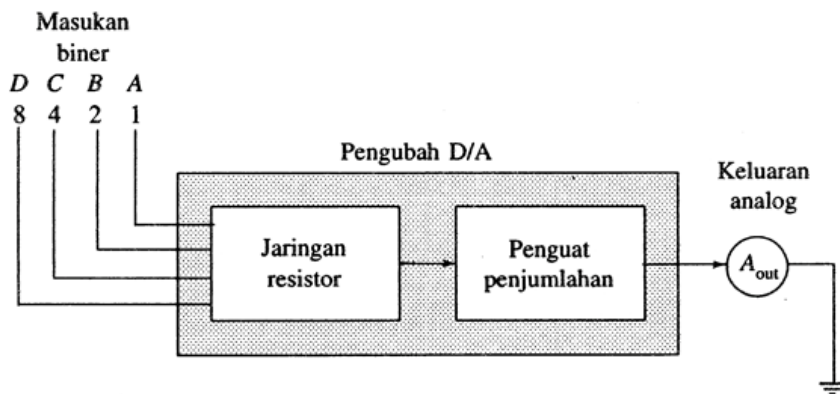
Gambar 9.89 Penggunaan konverter A/D dan D/A dalam suatu sistem elektronika

Digital to Analog Converter (DAC) berfungsi untuk mengubah/mentransformasikan masukan digital menjadi keluaran analog. Gambar 9.90 (a) mengilustrasikan fungsi dari pengubah D/A. Bilangan biner dimasukkan pada masukan-masukan di sebelah kiri, sedangkan tegangan keluaran yang bersangkutan berada di sebelah kanan. Seperti halnya tugas-tugas dalam elektronika yang

ra 3,75 V). Bila masukan adalah (0001), maka keluaran dari pengubah D/A tersebut menjadi 1 V. Baris 3 hanya menunjukkan masukan B yang diaktifkan (0010). Hal ini akan membangkitkan keluaran 4 V. Baris 9 hanya menunjukkan masukan D (1000) yang diaktifkan, yang menghasilkan keluaran 8 V dari pengubah D/A tersebut. Perhatikan, bahwa masukan-masukan tersebut



Gambar 9.90 Pengubah D/A

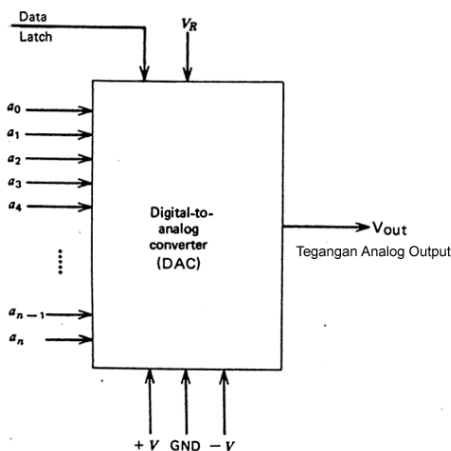


Gambar 9.91 Bagian dari suatu pengubah D/A

(D, C, B, A) adalah *berbobot* demikian rupa, sehingga level TINGGI (sekitar 3,75 V) pada masukan D akan membangkitkan 8 V, sedangkan level TINGGI pada masukan A hanya akan menghasilkan keluaran 1 V. Pembobotan yang relatif dari masing-masing masukan diberikan pada

Gambar 9.90 (a), yaitu : 8 untuk masukan D, 4 untuk masukan C, 2 untuk masukan B, dan 1 untuk masukan A. Pengubah D/A terdiri atas dua bagian fungsional. Gambar 13-3 mengilustrasikan diagram blok dari suatu pengubah D/A. Pengubah tersebut dibagi menjadi *jaringan resistor* dan *penguat penjumlahan*.

Jaringan resistor membebani masukan 1, 2, 4, dan 8 dengan tepat, sedangkan penguat penjumlahan memberikan Skala tegangan keluaran yang sesuai dengan tabel kebenaran. Sebagai penguat penjumlahan dalam suatu penguah D/A khusus kita gunakan IC jenis *penguat operasional* (operational amplifier). Penguat operasional sering disebut sebagai *op amp*. Perlu kita perhatikan dari Gambar 9.89, bahwa penguah D/A merupakan piranti perantara antara sistem digital dan lingkungan luar.



Gambar 9.92 Diagram dasar DAC, menunjukkan input dan output sinyal.

Pada aplikasi modern, kebanyakan DAC terintegrasi dalam bentuk IC, digambarkan sebagai "kotak hitam" yang mempunyai beberapa karakteristik input dan output. Pada gambar 9.92, kita lihat elemen yang diperlukan pada DAC dalam hubungannya dengan input dan output yang diperlukan. Karakteristik dari DAC tersebut, dapat disimpulkan di bawah ini, antara lain :

1. *Input digital (Digital Input)*. Berupa data biner secara parallel dari

sejumlah bits tertentu, dan level logic TTL diperlukan, kecuali jika ditentukan dalam IC DAC tertentu.

2. *Sumber tegangan (Power Supply)*. Menggunakan sumber bipolar pada level  $\pm 12$  sampai  $\pm 18$  Volt, diperlukan untuk amplifier internal. Beberapa DAC beroperasi dengan supply yang unipolar (0 sampai 18 Volt).
3. *Tegangan referensi (Reference Supply)*. Diperlukan untuk menetapkan *range* dari tegangan output dan resolusi dari konverter. Nilainya harus stabil, mempunyai riak yang kecil. Dalam beberapa unit, tegangan referensi sudah ada didalamnya.
4. *Keluaran (Output)*. Sebuah tegangan yang merepresentasikan input digitalnya. Tegangan ini berubah dalam setiap langkah sebagai perubahan input digitalnya dalam setiap bit-nya. Output sebenarnya bisa jadi bipolar jika konverter didesain untuk merepresentasikan input digital yang negatif.
5. *Offset*. Karena DAC biasanya diimplementasikan dengan menggunakan op-amp, ada tegangan offset outputnya dengan input zero-nya. Koneksi akan memberikan kemudahan untuk membuat nol dari output DAC dengan input nolnya.
6. *Data terkunci (Data Latch)*. Banyak DAC mempunyai membuat data terkunci pada inputnya. Ketika sebuah perintah logic diberikan untuk mengunci suatu data, apapun data pada terminal input akan terkunci dalam DAC dan analog output akan berubah berdasarkan data input tersebut. Output akan tetap nilainya selama data digital yang baru terkunci

sebagai input. Dengan cara ini, input DAC dapat disambungkan secara langsung pada data bus dari komputer, tetapi ini akan berubah datanya hanya ketika perintah mengunci diberikan oleh sebuah komputer.

7. *Waktu konversi (Conversion Time)*. Sebuah DAC melaksanakan konversi digital input menjadi analog output dengan cepat. Waktu konversi tersebut biasanya terjadi dalam beberapa mikrodetik.

Output dari DAC dapat didefinisikan ke dalam persamaan :

$$V_{out} = V_R [b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n} +]$$

dimana,

$V_{out}$  = tegangan analog output.

$V_R$  = tegangan referensi.

$b_1 b_2 \dots b_n$  = data biner n-bit.

Nilai minimum dari  $V_{out}$  adalah nol, dan maksimumnya tergantung dari ukuran n-bit data binernya. Sebagai contoh data 4 bit mempunyai nilai maksimum :

$$V_{max} = V_R [2^{-1} + 2^{-2} + 2^{-3} + 2^{-4}] = 0.9375 V_R$$

Dan data 8-bit mempunyai nilai maksimum :

$$V_{max} = V_R [2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8}] = 0.9961 V_R$$

Alternatif persamaan di atas, yang seringkali lebih mudah dipakai. Yaitu :

$$V_{out} = \frac{N}{2^n} V_R$$

dimana, N = input DAC dengan data berbasis 10 (desimal). Diketahui konverter 8 bit dengan tegangan

referensi 5 Volt, mempunyai input  $10100111_2$ , atau A7H. Jika input disini dikonversi ke basis 10 (desimal), kita dapatkan  $N = 167_{10}$  dan  $2^8 = 256$ . Dari persamaan di atas maka kita dapatkan output DAC sebesar

$$V_{out} = \frac{167}{256} 5 = 3.2617 \text{ Volt.}$$

*Analog to Digital Converter* (ADC) sering diperlukan dalam perancangan suatu rangkaian elektronika. Meskipun banyak sensor yang menghasilkan output sinyal digital secara langsung, baik yang sudah ada atau masih dalam pengembangan, masih banyak variabel terukur dalam bentuk sinyal tegangan elektrik yang perlu dikonversi. Dengan semakin banyaknya pemakaian rangkaian logika digital dan komputer dalam sistem kontrol, maka diperlukan pemakaian suatu ADC untuk menghasilkan sinyal yang terkodekan secara digital pada komputer. Persamaan ADC adalah sebagai berikut :

$$V_{in} = V_R [b_1 2^{-1} + b_2 2^{-2} + \dots + b_n 2^{-n} +]$$

dimana,

$V_{in}$  = tegangan analog input.

$V_R$  = tegangan referensi.

$b_1 b_2 \dots b_n$  = output data biner n-bit.

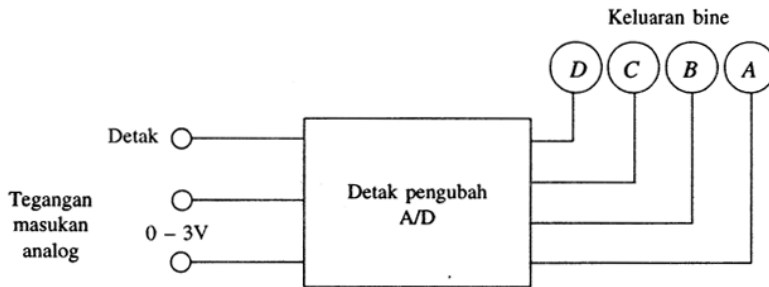
Dengan setiap langkah untuk 1 bit data outputnya mempunyai nilai sebesar :

$$\Delta V = V_R 2^{-n}$$

Untuk itu, Ada suatu ketidakpastian yang menjadi sifatnya dari  $\Delta V$  dalam banyak kasus konversi tegangan analog menjadi sinyal digital.

Ketidaktepatan ini harus bisa dibawa ke dalam nilai yang diharapkan dalam sebuah desain aplikasi. Jika masalah dibawah pertimbangan yang sudah

bersangkutan. Keluaran biner tersebut akan berbanding lurus dengan masukan analog. Diagram blok dari suatu pengubah A/D dipa-



Gambar 9.93 Diagram blok dari suatu pengubah A/D 4 bit

Baris	Masukan analog	Keluaran biner				Baris	Masukan analog	Keluaran biner			
		8s	4s	2s	1s			8s	4s	2s	1s
$V_{in}$	D	C	B	A	$V_{in}$		D	C	B	A	
1	0	0	0	0	0	9	1,6	1	0	0	0
2	0.2	0	0	0	1	10	1,8	1	0	0	1
3	0.4	0	0	1	0	11	2,0	1	0	1	0
4	0.6	0	0	1	1	12	2,2	1	0	1	1
5	0.8	0	1	0	0	13	2,4	1	1	0	0
6	1.0	0	1	0	1	14	2,6	1	1	0	1
7	1.2	0	1	1	0	15	2,8	1	1	1	0
8	1.4	0	1	1	1	16	3,0	1	1	1	1

Gambar 9.94 Tabel kebenaran untuk suatu pengubah A/D 4 bit

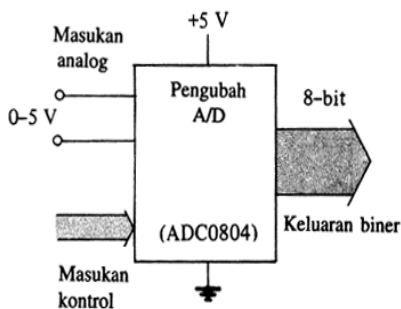
ditentukan dari sebuah resolusi tertentu dalam tegangan analog, maka ukuran data dan tegangan referensi harus dipilih untuk mendapatkan data digital yang telah dikonversi.

Pengubah analog ke digital (pengubah A/D) membalik proses dari pengubah D/A. Tegangan analog yang tak diketahui dimasukkan ke dalam pengubah A/D, dan akan muncul keluaran biner yang

perhatikan pada Gambar 9.92. Pengubah A/D ini mempunyai tegangan masukan analog yang berkisar dari 0 sampai 3 V. Kemudian, keluaran biner tersebut akan terbaca dalam bentuk biner dari 0000 sampai 1111. Perhatikan bahwa pengubah A/D tersebut juga mempunyai masukan detak. Tabel kebenaran pada Gambar 9.93 menjelaskan dengan terinci operasi pengubah A/D. Perlu kita perhatikan bahwa sisi masukan

dalam tabel kebenaran tersebut memaparkan tegangan masukan analog, sedangkan sisi keluaran memberikan pembacaan biner yang bersangkutan. Perhatikan Pula bahwa untuk masing-masing perubahan 0,2 V dalam tegangan masukan hitungan keluaran biner akan bertambah dengan 1. Tabel kebenaran pengubah A/D ini (Gambar 9.93) merupakan kebalikan dari pengubah D/A pada Gambar 9.90.

Ada banyak IC yang dipakai sebagai rangkaian ADC. Pabrik menghasilkan ratusan pengubah analog ke digital yang berbeda dalam bentuk IC. Dengan menggunakan teknologi CMOS modem, banyak gambaran yang dapat ditambahkan terhadap perubahan A/D IC sementara yang

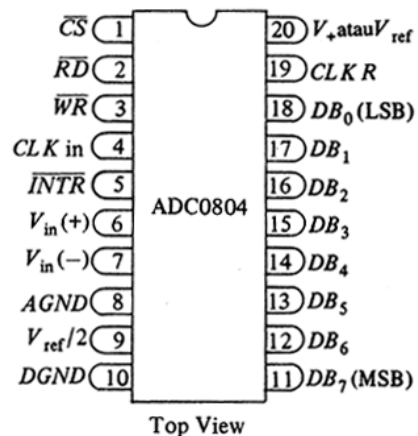


Gambar 9.95 Penyederhanaan blok diagram dari pengubah A/D IC ADC0804 b bit

penghamburan daya tertahan dan pembayarannya dengan level yang rendah.

Perubahan IC A/D 8-bit akan digambarkan dalam bagian ini. Mikroprosesor ADC0804 8-bit pengubah A/D penyesuai 20-pin IC DIP yang dihasilkan oleh pabrik

seperti National Semiconductor, Signetics, dan Intersil. Sebuah blok sederhana dari pengubah fungsi A/D ADC0804 ditunjukkan dalam Gambar 9.95. Pertama baris pengontrol pengubah A/D secara langsung disampling dan digitalkan dengan tegangan analog pada masukan. Kedua, baris pengontrol pengubah A/D secara langsung membangkitkan 8-bit keluaran biner. Keluaran biner 8-bit adalah langsung disesuaikan terhadap masukan tegangan analog. Jika tegangan masukan 5 V maka keluaran biner seharusnya 11111111. Apabila tegangan masukan 0 V, keluaran biner akan 00000000.



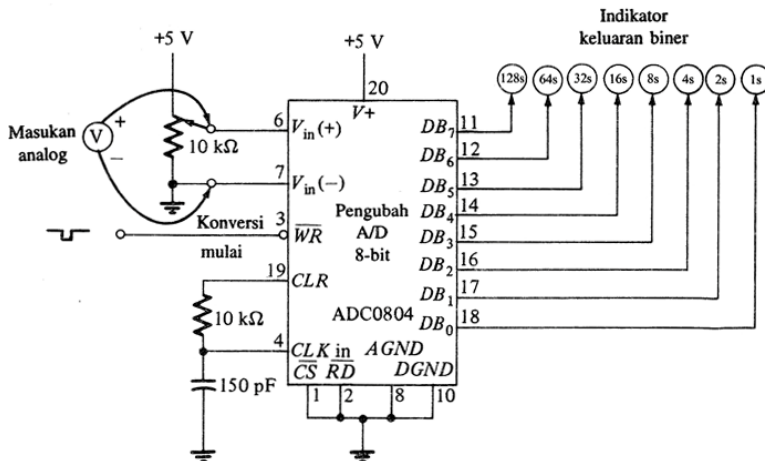
Gambar 9.96 Diagram pin dari ADC 0804 8 bit pengubah A/D.

Sebuah diagram pin dari pengubah A/D IC ADC0804 ditunjukkan dalam Gambar 9.96. IC ADC0804 adalah sebuah CMOS 8-bit berurut diperkirakan pengubah A/D di many perancangannya dioperasikan dengan mikroprosesor 8080A tanpa tambahan interface. IC pengubah waktu ADC0804 adalah di bawah 100 mikro detik, dan semua masukan dan



keluaran adalah TTL yang lengkap. Pengoperasian dengan catu daya 5 V, dan akan dapat menangani secara penuh jarak 0 s/d 5 V masukan analog di antara pin 6 dan 7. IC

hanya beberapa dolar. Jika tegangan masukan analog diukur dengan teliti melalui sebuah DMM (digital multimeter), pengubah A/D ADC0804 didapatkan dengan ketelitian yang



Gambar 9.97 Sebuah pengetest rangkaian yang menggunakan pembalik IC ADC0804 8 bit

ADC0804 mempunyai chip pendetak pembangkit di mana hanya memerlukan resistor luar dan kapasitor (perhatikan Gambar 9.97).

Sebuah contoh lab setup digunakan ADC0804 A/D pengubah adalah ditunjukkan dalam Gambar 9.97. Masukan tegangan analog dikembangkan menrberang pengelap dan ground dari 10 kΩ potensiometer. Hasil dare A/D pengubah adalah  $1/255$  ( $2 - 1$ ) dare skala penuh tegangan analog (5 V dalam contoh ini). Untuk setiap penambahan 0,02 V ( $1/255 \times 5 \text{ V} = 0,02 \text{ V}$ ), merupakan keluaran baris yang ditambah 1. Jika masukan analog sama dengan 0,1 V, keluaran biner akan 00000101 ( $0,1 \text{ V}/0,02 \text{ V} = 5$ , dan desimal 5 = 00000101 dalam biner).

Rangkaian yang ditunjukkan dalam Gambar 9.97 dapat diperoleh

baik atas masukan tegangan tertentu. Pabrik memberikan *total kesalahan* untuk ADC0804  $\pm 1$  LSB. Perpindahan H-ke-L pada pulsa detak pada masukan WR terhadap IC ADC0804 ditunjukkan dalam Gambar 9.97 awal proses perubahan. Keluaran biner timbul 100 1.1s pada indikator (petunjuk di sebelah kanan) keluaran-keluaran tiga keadaan bufer, juga semuanya dapat dihubungkan secara langsung terhadap data bus pada sebuah dasar sistem mikroprosesor. pengubah A/D ADC0804 mempunyai *interrupt keluaran* (INTR, lihat kaki 5, Gambar 9.96) di mana sinyal sistem mikroprosesor arus analog ke digital pengubah selesai. Interup diperlukan di dalam sistem mikroprosesor di mana interface sangat "pelan". Setiap piranti asinkronos pengubah A/D ke

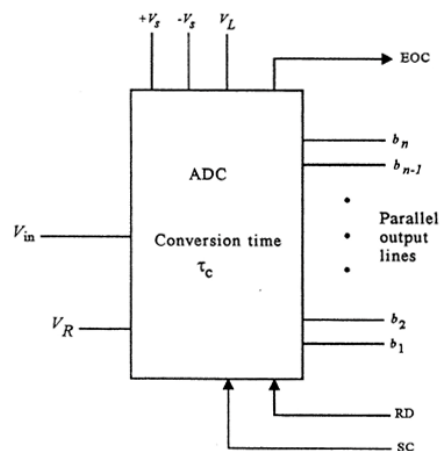
"sangat cepat" piranti sinkronous untuk sebuah mikroprosesor. Karakteristik yang penting dari ADC dapat diringkas sebagai berikut :

1. *Tegangan analog input (Analog voltage input)*. Ini merupakan yang disambungkan dengan tegangan yang akan dikonversi. Yang penting adalah bahwa nilai tegangan ini harus konstant selama proses konversinya.
2. *Sumber tegangan (power supply)*. Secara umum, sebuah ADC memerlukan sumber tegangan bipolar untuk op-amp internal dan sumber digital logic-nya.
3. *Tegangan referensi (reference voltage)*. Tegangan referensi harus stabil, dengan memakai sumber yang teregulasi dengan baik.
4. *Digital output*. Pengubah akan mempunyai  $n$  output untuk koneksi dengan rangkaian *interfacing* digital. Secara umum, level dengan nilai TTL untuk mendefinisikan kondisi high dan low.
5. *Control lines*. ADC mempunyai sejumlah *control line* dengan 1 bit digital input dan output, didesain untuk operasi kontrol ADC dan mengizinkan untuk berhubungan dengan sebuah komputer. *Common line* tersebut adalah :

- a. *SC (Start Convert)*. Ini adalah digital input ke ADC yang memulai konverter untuk proses pencarian digital output yang benar untuk diberikan input tegangan analog. Secara khusus, konversi dimulai pada tegangan yang berubah dari high ke low.
- b. *EOC (End of Convert)*. Ini merupakan digital output dari ADC untuk diterima oleh suatu

peralatan, misalnya komputer. Saluran ini akan bernilai high selama proses konversi. Ketika proses konversi sudah selesai, maka saluran akan menjadi low. Dimana perubahan dari level high menjadi low mengindikasikan bahwa proses konversi sudah selesai.

- c. *RD (Read)*. Ketika output di buffer dengan tri-state, terjadi proses konversi yang lengkap, hasil digital yang benar tidak di keluarkan ke saluran output. Peralatan penerima harus menjadikan saluran RD menjadi low untuk membolehkan tri-state dan mengirim data ke saluran output.



Gambar 9.98 Diagram ADC, menunjukkan sinyal input dan output dan gambaran waktu konversi

6. *Waktu konversi (conversion time)*. Ini bukan input ataupun output, tetapi sangat penting dalam karakteristik suatu ADC. ADC tidak menghasilkan output digital secara terus-menerus ketika tegangan analog dibeikan pada terminal inputnya. ADC bisa secara

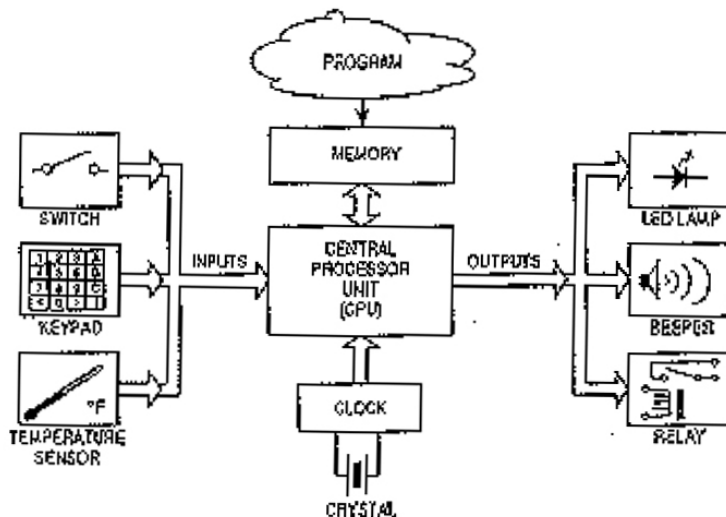
sekuensial memproses untuk mendapatkan output digitalnya dan proses ini memerlukan waktu.

### 9.5.3 Sistem Mikroprosesor dan Mikrokontroler

Mikrokontroler adalah salah satu dari bagian dasar dari suatu sistem komputer. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan komputer mainframe, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Secara sederhana, komputer akan menghasilkan output spesifik berdasarkan inputan yang diterima dan program yang dikerjakan.

melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks yang diinginkan oleh programmer.

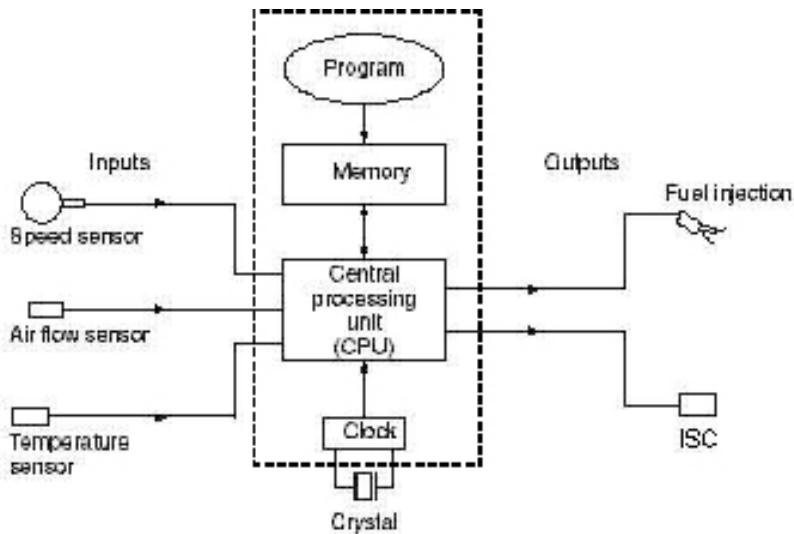
Piranti input (pada sistem input komputer) menyediakan informasi kepada sistem komputer dari dunia luar. Dalam sistem komputer pribadi, piranti input yang paling umum adalah *keyboard* dan *mouse*. Komputer mainframe menggunakan *keyboard* dan pembaca kartu berlubang sebagai piranti inputnya. Sistem dengan mikrokontroler umumnya menggunakan piranti input yang jauh lebih kecil seperti saklar atau *keypad* kecil, menghasilkan tegangan ON/OFF saja.



Gambar 9.99 Elemen dasar komputer

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 yang dibuat oleh seorang programmer. Program ini menginstruksikan komputer untuk

Hampir semua input mikrokontroler hanya dapat memproses sinyal input digital dengan tegangan yang sama dengan tegangan logika dari sumber. Level 0 disebut dengan VSS dan tegangan positif sumber (VDD) umumnya adalah 5 volt. Padahal dalam dunia nyata terdapat banyak sinyal analog atau sinyal dengan



Gambar 9.100 Sensor, sistem komputer dan aktuator

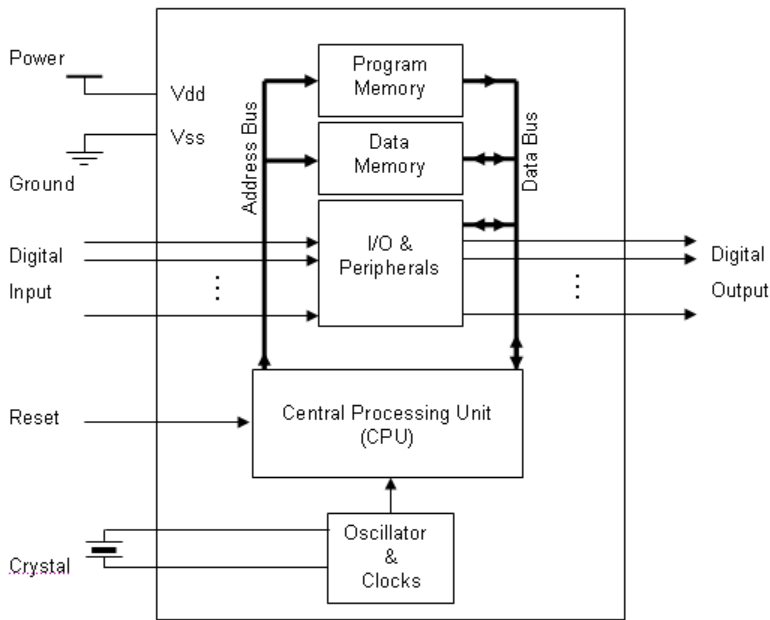
tegangan level yang bervariasi. Karena itu ada piranti input yang mengkonversikan sinyal analog menjadi sinyal digital sehingga komputer bisa mengerti dan menggunakannya. Ada beberapa mikrokontroler yang dilengkapi dengan piranti konversi ini, yang disebut dengan ADC, dalam satu rangkaian terpadu.

Piranti output (pada sistem output Komputer) digunakan untuk berkomunikasi informasi maupun aksi dari sistem komputer dengan dunia luar. Dalam sistem komputer pribadi (PC), piranti output yang umum adalah monitor CRT, tetapi sekarang semakin banyak orang yang menyukai monitor LCD. Sedangkan sistem mikrokontroler mempunyai output yang jauh lebih sederhana seperti lampu indikator atau beeper. Frasa kontroler dari kata mikrokontroler memberikan penegasan bahwa alat ini mengontrol sesuatu.

Mikrokontroler mengolah sinyal secara digital, sehingga untuk dapat memberikan output analog diperlukan proses konversi dari sinyal digital menjadi analog. Piranti yang dapat melakukan konversi ini disebut dengan DAC (*digital to analog converter*).

CPU (*Central processing unit*) adalah otak dari sistem komputer. Pekerjaan utama dari CPU adalah mengerjakan program yang terdiri atas instruksi-instruksi yang dibuat oleh programmer. Programmer membuat logika berpikir yang dirubah ke bahasa pemrograman. Program komputer akan menginstruksikan CPU untuk membaca informasi dari piranti input, membaca dan menulis informasi dari/ke memori, dan untuk menulis informasi ke output.

Dalam mikrokontroler umumnya hanya ada satu program yang bekerja dalam suatu aplikasi. CPU AT89S51 mengenali hanya beberapa puluh



Gambar 9.101 Blok diagram sistem mikrokontroler

instruksi yang berbeda. Karena itu sistem komputer ini sangat cocok dijadikan model untuk mempelajari dasar dari operasi komputer karena dimungkinkan untuk menelaah setiap operasi yang dikerjakan.

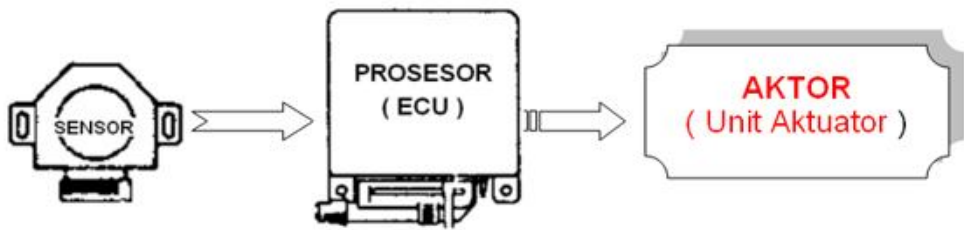
Sistem komputer menggunakan osilator *clock* untuk memicu CPU untuk mengerjakan satu instruksi ke instruksi berikutnya dalam alur yang berurutan. Setiap langkah kecil dari operasi mikrokontroler memerlukan waktu satu atau beberapa *clock* untuk melakukannya.

Ada beberapa macam tipe dari memori komputer yang digunakan untuk beberapa tujuan yang berbeda dalam sistem komputer. Tipe dasar yang sering ditemui dalam mikrokontroler adalah ROM (*read only memory*) dan RAM (*random access memory*). ROM digunakan

sebagai media penyimpanan program dan data permanen yang tidak boleh berubah meskipun tidak ada tegangan yang diberikan pada mikrokontroler. RAM digunakan sebagai tempat penyimpanan data sementara dan hasil kalkulasi selama proses operasi. Beberapa mikrokontroler mengikutsertakan tipe lain dari memori seperti EPROM (*erasable programmable read only memory*) dan EEPROM (*electrically erasable programmable read only memory*).

Program komputer adalah hasil imajinasi seorang programmer. Komponen utama dari program adalah instruksi-instruksi dari instruksi set CPU. Program disimpan dalam memori dalam sistem komputer di mana mereka dapat secara berurutan dikerjakan oleh CPU.

listrik, yang dikendalikan oleh media



Gambar 9.102 Blok diagram hubungan antara sensor, mikroprosesor dan aktuator

Setelah dipaparkan bagian-bagian dari suatu sistem komputer, sekarang akan dibahas mengenai mikrokontroler. Digambarkan sistem komputer dengan bagian yang dikelilingi oleh garis putus-putus. Bagian inilah yang menyusun mikrokontroler. Bagian yang dilingkupi kotak bagian bawah adalah gambar lebih detail dari susunan bagian yang dilingkupi garis putus-putus. Kristal tidak termasuk dalam sistem mikrokontroler tetapi diperlukan dalam sirkuit osilator *clock*.

Suatu mikrokontroler dapat didefinisikan sebagai sistem komputer yang lengkap termasuk sebuah CPU, memori, osilator *clock*, dan I/O dalam satu rangkaian terpadu. Jika sebagian elemen dihilangkan, yaitu I/O dan memori, maka chip ini akan disebut sebagai mikroprosesor.

#### 9.5.4 Aktuator

Aktuator adalah sebuah peralatan listrik atau mekanis untuk menggerakkan atau mengontrol sebuah mekanisme atau sistem. Aktuator diaktifkan dengan menggunakan lengan mekanis yang biasanya digerakkan oleh motor

diantaranya mikrokontroler. Macam-macam aktuator adalah sebagai berikut :

- Aktuator tenaga listrik, biasanya digunakan solenoid, motor arus searah, mempunyai sifat mudah diatur dengan torsi kecil sampai sedang.
- Aktuator tenaga hidrolik. Torsi yang besar, konstruksinya sukar.
- Aktuator tenaga pneumatik. Sukar dikendalikan
- Aktuator Lainnya : *piezoelectric*, magnetik, ultra sound dan lain-lain.

Bentuk umum dari aktuator adalah sebagai berikut :

1. *Relay*, adalah peralatan yang dioperasikan secara elektrik yang secara mekanik akan menswitch sirkuit elektrik. Relay merupakan bagian yang penting dalam sistem kontrol, karena kegunaannya dalam kendali jarak jauh, dan mengendalikan listrik tegangan tinggi dengan menggunakan listrik tegangan rendah. Ketika tegangan mengalir ke dalam elektromagnet pada sistem kontrol *relay*, maka magnet akan menarik lengan logam pada arah magnet, dengan demikian kontak

- terjadi. Relay bisa memiliki jenis NO atau NC ataupun dua-duanya.
2. *Solenoida*. Solenoida adalah alat yang digunakan untuk mengubah sinyal listrik atau arus listrik menjadi gerakan mekanis linear. Terbentuk dari kumparan dengan inti besi yang dapat bergerak, besarnya gaya tarikan atau dorongan yang dihasilkan adalah ditentukan dengan jumlah lilitan kumparan tembaga dan besar arus yang mengalir melalui kumparan.
  3. *Stepper*. *Stepper* adalah alat yang mengubah pulsa listrik yang diberikan menjadi gerakan rotor diskrit (tidak kontinyu) yang disebut *step* (langkah). Satu putaran motor memerlukan 360 derajat dengan jumlah langkah yang tertentu perderajatnya. Ukuran kerja dari *stepper* biasanya diberikan dalam jumlah langkah per-putaran per-detik. Motor *stepper* mempunyai kecepatan dan torsi yang rendah namun memiliki kontrol gerakan posisi yang cermat, hal ini dikarenakan memiliki beberapa segmen kutub kumparan.
  4. *Motor DC*. Motor DC adalah alat yang mengubah energi listrik menjadi gerak, mempunyai prinsip dasar yang sama dengan motor *stepper* namun gerakannya bersifat kontinyu atau berkelanjutan. Motor DC dibagi menjadi 2 jenis yaitu ; Motor DC dengan sikat (mekanis komutasi), yaitu motor yang memiliki sikat karbon berfungsi sebagai pengubah arus pada kumparan sedemikian rupa sehingga arah tenaga putaran motor akan selalu sama. Motor

DC tanpa sikat, menggunakan semi konduktor untuk merubah maupun membalik arus sehingga layaknya pulsa yang menggerakkan motor tersebut. Biasa digunakan pada sistem servo, karena mempunyai efisiensi tinggi, umur pemakaian lama, tingkat kebisingan suara listrik rendah, karena putarannya halus seperti *stepper* namun putarannya terus menerus tanpa adanya *step*.

## 9.6 Sistem Kontrol Analog dan Digital (Elektronik)

Pada sistem kontrol elektronik, kontroler yang digunakan merupakan suatu unit yang terdiri dari komponen elektronika. Unit elektronika disini merupakan rangkaian yang terintegrasi dari banyak komponen elektronika, yaitu resistor, kapasitor, induktor, dioda, transistor, op-amp, IC dan masih banyak komponen elektronika yang lain. Minimum disini maksudnya adalah dimana kontroler yang berupa rangkaian elektronika terdiri dari komponen-komponen elektronika baik resistor, kapasitor, induktor, dioda, transistor, op-amp sampai dengan IC tertentu yang terangkai menjadi satu-kesatuan yang utuh dengan jumlah komponen yang seminimal mungkin tetapi sudah mampu untuk membentuk sistem kontrol tertentu. Dengan pengertian lain pada bagian ini akan diberikan konsep sistem kontrol yang sederhana dengan rangkaian elektronika, tetapi dari segi fungsi sudah bisa tercapai. Sistem kontrol elektronika ini bisa berupa sistem kontrol loop terbuka ataupun sistem kontrol loop tertutup. Pada dasarnya ada dua jenis sistem kontrol

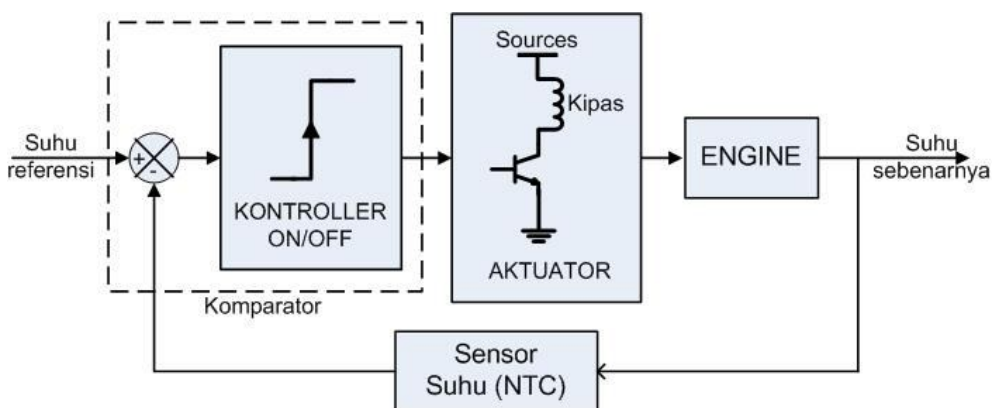
elektronika yang bisa dibentuk, yaitu sistem kontrol analog dan sistem kontrol digital.

Dalam sistem kontrol analog, pengontrol tersusun dari piranti dan rangkaian analog yang tradisional, kebanyakan memakai penguat linear (linear amplifier). Banyak yang memakai Op-Amp karena dengan komponen ini kita bisa membuat rangkaian yang merepresentasikan persamaan yang kita buat. Bagaimana merancang sistem kontrol analog, di bawah akan kita bahas dari perancangan sampai terbentuk sistem kontrol yang sudah siap diterapkan. Pada awalnya, sistem kontrol yang dibuat bersifat analog karena teknologi analog adalah satu-satunya yang tersedia pada saat itu. Dalam sistem kontrol analog, setiap perubahan, baik pada referensi maupun pada umpanbalik, dapat terindra secara segera, dan langsung penguat menyesuaikan outputnya (kepada aktuator). Hal ini yang membedakan dengan sistem kontrol digital. Karena pada sistem kontrol analog, sinyal yang diperoleh secara langsung dan kontinyu

diproses tanpa memerlukan waktu dan langsung diproses ke piranti berikutnya.

Sebagai contohnya, mari kita membuat sistem kontrol suhu mesin pada mobil. Pertama kali harus dibuat adalah membuat blok diagram sistem kontrolnya. Blok diagram ini bisa dijelaskan sesuai pada gambar 9.103. Tampak bahwa sistem ini merupakan sistem kontrol loop tertutup. Hal ini bisa diamati dengan adanya umpan balik yang diberikan dari keluaran engine (berupa besaran suhu) ke pembanding (+/-) sebelum masuk ke kontroler. Tentunya agar besaran mekanis bisa diolah secara elektronik, maka diperlukan sebuah sensor yang mampu merubah besaran suhu ( $^{\circ}\text{C}$ ) menjadi besaran listrik, yaitu tegangan dengan satuan Volt.

Bagaimana strategi kontrol yang diterapkan dalam sistem ini. Hal ini akan dijelaskan sebagai berikut. Suhu referensi yang menjadi tujuan dari sistem ini, mempunyai harga yang tetap. Sebagai misal kita tentukan nilai suhu referensi adalah  $80^{\circ}\text{C}$ . Nilai ini adalah harga yang disaran-



Gambar 9.103 Blok diagram sistem kontrol suhu engine



kan pada engine yang bekerja, atau disebut suhu kerja engine. Dengan kata lain, diharapkan bahwa suhu sebenarnya pada engine nanti adalah disekitar  $80^{\circ}\text{C}$ , tidak boleh kurang maupun lebih dari  $80^{\circ}\text{C}$ . Ada lingkaran dengan tanda +/-, ini disebut pembanding. Fungsi dari komponen ini adalah membandingkan suhu engine yang terbaca oleh sensor dengan suhu referensi yang diberikan. Sehingga dengan memakai persamaa yang sederhana yaitu pengurangan suhu referensi dengan suhu terbaca, maka akan didapatkan nilai error tertentu.

$$\text{error} = \text{suhu ref.} - \text{suhu real}$$

*(error = tujuan - output)*

Tampak pada blok diagram gambar 9.103, bahwa kontroler yang dipakai adalah kontroler ON/OFF. Sehingga nilai error yang didapatkan dari kalkulasi di atas, maka akan kita dapatkan ada tiga kemungkinan yang ada, yaitu :

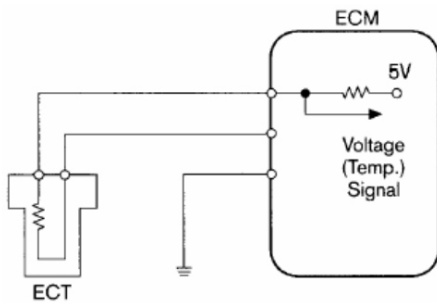
1. *Error bernilai negatif.*
2. *Error bernilai positif.*
3. *Error bernilai nol.*

Dari tiga kemungkinan ini, maka kontroler akan mengambil keputusan, apakah aktuator akan di ON ataukah di OFF kan. Aktuator berfungsi untuk merubah *sinyal perintah* yang berasal dari kontroler menjadi *sinyal aksi* yang akan diberikan ke engine. Aktuator diperlukan karena sinyal perintah yang berasal dari kontroler tidak mampu untuk menggerakkan kipas secara langsung, karena memang sinyal yang berasal dari kontroler merupakan sinyal yang lemah. Sehingga dalam hal ini

memerlukan transistor power agar mampu meng-ON kan kipas. Disamping transistor power maka agar suhu engine bisa menurun, maka diperlukannya angin yang berhembus ke sistem radiator untuk mendinginkan air yang tentunya mengalir ke dalam engine dan menyerap panas engine untuk dikeluarkan lagi ke radiator. Maka diperlukan kipas untuk merubah sinyal listrik menjadi hembusan angin. Kipas merupakan salah satu bentuk aktuator juga.

Kemudian bagaimana yang terjadi dengan temperatur engine. Suhu sebenarnya yang dihasilkan oleh engine bisa bervariasi. Bisa dingin, sedang ataupun tinggi. Hal inilah, kenapa sistem kontrol ini diperlukan. Apabila sistem kontrol ini berfungsi dengan baik, tentunya suhu sebenarnya akan berkisar didaerah  $80^{\circ}\text{C}$ . Seperti sudah diketahui bahwa kenapa suhu kerja engine harus di sekitar  $80^{\circ}\text{C}$ . Pada kondisi temperatur rendah maka kerja dari engine tidak bisa menghasilkan power yang optimal, begitu juga jika temperature tinggi. Jika temperature sangat tinggi, terjadi *overheating*, maka akibatnya akan sangatlah fatal. Hal ini bisa menyebabkan kerusakan pada engine. Sehingga dengan adanya sistem kontrol ini, diharapkan temperatur pada engine selalu berada di suhu kerjanya.

Setelah dibuat blok diagram, maka mari kita buat rangkaian elektroniknya. Pada umpan balik, ada sensor suhu dengan tipe NTC. Tentunya agar bisa dirubah menjadi sinyal tegangan, maka diperlukan rangkaiannya. Rangkaian tersebut bisa ditunjukkan pada gambar 9.104. Tampak pada gambar tersebut bahwa rangkaian yang ada adalah

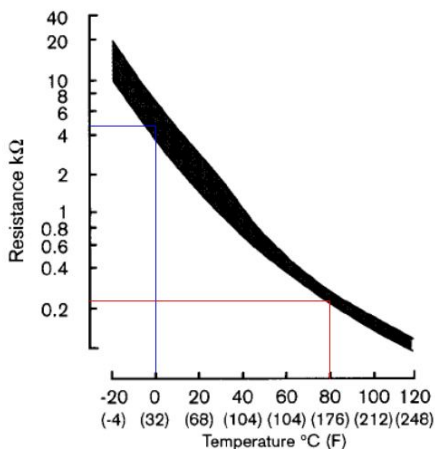


Gambar 9.104 Rangkaian sensor temperature yang sederhana

rangkaian pembagi tegangan atau disebut rangkaian *divider*. Sehingga tegangan yang terbaca oleh *Electronic Control Unit (ECU)* atau *Electronic Control Modul (ECM)* didapatkan persamaan

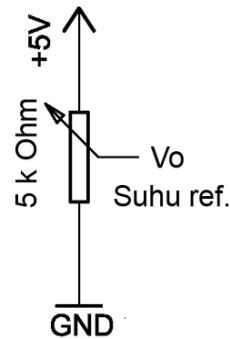
$$V_o = (R_{ECT} / (R_{ECT} + R_{ECU})) \cdot 5 \text{ Volt}$$

Harga  $R_{ECU}$  tergantung dari desain yang dibuat. Bisa dibuat dengan nilai 1 k $\Omega$  - 5 k $\Omega$ .



Gambar 9.105 Hubungan antara temperatur dengan nilai hambatan yang dihasilkan

Semakin besar temperatur, maka semakin kecil hambatan  $R_{ECT}$ , sehingga akan didapatkan nilai tegangan  $V_o$  yang semakin kecil. Sebaliknya jika temperature engine semakin kecil, maka semakin besar hambatan  $R_{ECT}$  sehingga semakin besar pula tegangan  $V_o$  yang dihasilkan. Dengan kata lain ada hubungan berbanding terbalik antara besaran temperatur dengan tegangan listrik yang dihasilkan. Hubungan antara temperatur dengan nilai hambatan yang dihasilkan bisa diperhatikan pada gambar 9.105.

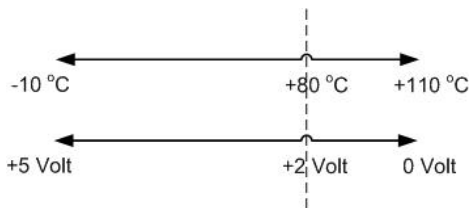


Gambar 9.106 Rangkaian divider dengan potensiometer untuk membentuk suhu referensi.

Sedangkan suhu referensi bisa kita buat dengan memakai rangkaian divider. Tegangan keluaran dari rangkaian divider (suhu referensi), harus stabil. Sehingga power supply 5 Volt yang dilewatkan regulator diperlukan agar didapatkan harga yang tidak berubah-ubah. Dengan menghubungkan kedua ujung potensiometer ke sumber 5 Volt dan ground, maka terminal tengah dari potensiometer sudah menghasilkan tegangan tertentu yang kita inginkan. Tegangan inilah yang menjadi nilai dari besaran suhu referensi pada blok

diagram sistem kontrol di atas. Nilai dari  $V_o$  pada gambar 9.106 diatur supaya harganya merepresentasikan nilai temperatur  $80^\circ\text{C}$ .

Kita asumsikan bahwa relasi antara temperatur sebenarnya dengan tegangan temperatur terukur sesuai dengan gambar 9.107. Tampak bahwa pada suhu  $80^\circ\text{C}$  didapatkan  $V_o$  sebesar 2 Volt.



Gambar 9.107 Hubungan antara suhu sebenarnya dengan tegangan terukur

Sehingga suhu referensi harus kita atur agar bernilai 2 Volt. Tentunya mengatur nilai ini dengan merubah nilai dari kedua resistor yang berada di dalam potensiometer dengan cara memutar pegangan potensiometer.

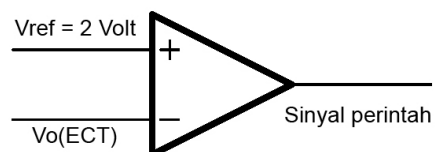
Kemudian bagaimana dengan rangkaian pembanding dan kontrolernya. Tampak pada gambar 9.103 kontroler dan pembanding di selubungi oleh garis putus-putus. Dengan menggunakan rangkaian komparator (lihat bagian 9.6.2.2.4) maka pembanding dan kontroler (ON/OFF) bisa dijadikan menjadi 1 rangkaian komparator dengan menggunakan Op-Amp. Dengan melihat gambar 9.85, maka dengan adanya tiga kemungkinan dari nilai error, yaitu positif, negatif dan nol, maka bisa kita tentukan dimana harus kita sambungkan suhu referensi, suhu terbaca ke masing-masing terminal input dari Op-Amp. Pada kondisi awal sebelum engine *running* maka suhu temperature adalah dingin

dimana tegangan suhu yang terukur masih di atas 2 Volt. Sehingga kipas harus dalam kondisi OFF (kondisi  $V_o > 2$  Volt). Sehingga sinyal perintah dari kontroler yang berupa rangkaian komparator Op-Amp harus menghasilkan nilai 0 Volt. Jika mobil distarter, maka engine akan *running*, tentunya karena terjadi pembakaran maka timbul panas. Semakin lama, temperatur akan panas sehingga suhu engine akan mendekati  $80^\circ\text{C}$  dimana tegangan  $V_o$  akan mendekati 2 Volt. Selama suhu engine belum di atas  $80^\circ\text{C}$ , maka kondisi kipas akan tetap OFF (sinyal perintah = low). Sampai jika suhu temperatur di atas  $80^\circ\text{C}$  maka kipas harus ON (sinyal perintah = high). Hal ini mengandung maksud agar terjadi proses pendinginan. Kejadian ini akan terus berlangsung terus menerus sampai posisi kunci kontak mobil di OFF kan.

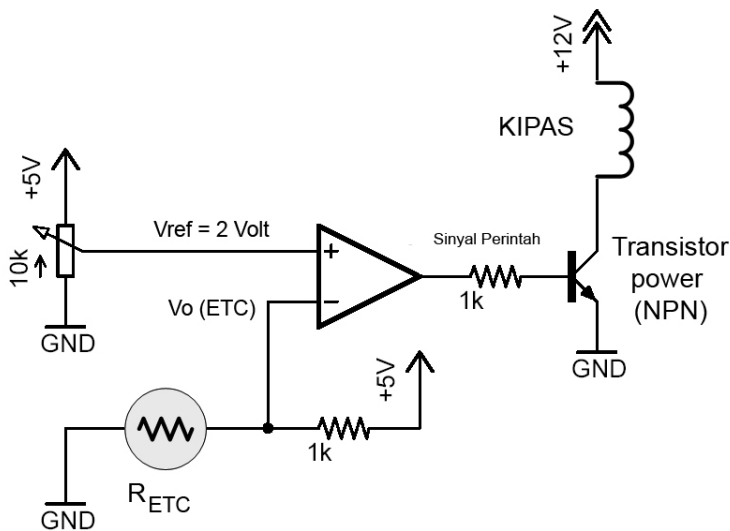
Dari keterangan tersebut bisa kita simpulkan, yaitu sebagai berikut :

1.  $Temperature < 80^\circ\text{C}$ .  
 $V_o > 2$  Volt, maka kipas OFF
2.  $Temperature > 80^\circ\text{C}$ .  
 $V_o < 2$  Volt, maka kipas ON
3.  $Temperature = 80^\circ\text{C}$ .  
 $V_o = 2$  Volt, maka kondisi kipas sama dengan sebelumnya.

Dari ketiga kemungkinan ini, maka bisa dibuat rangkaian komparator seperti terlihat pada gambar 9.108.



Gambar 9.108 Rangkaian komparator Op-amp dengan permasalahan pada sistem kontrol suhu engine



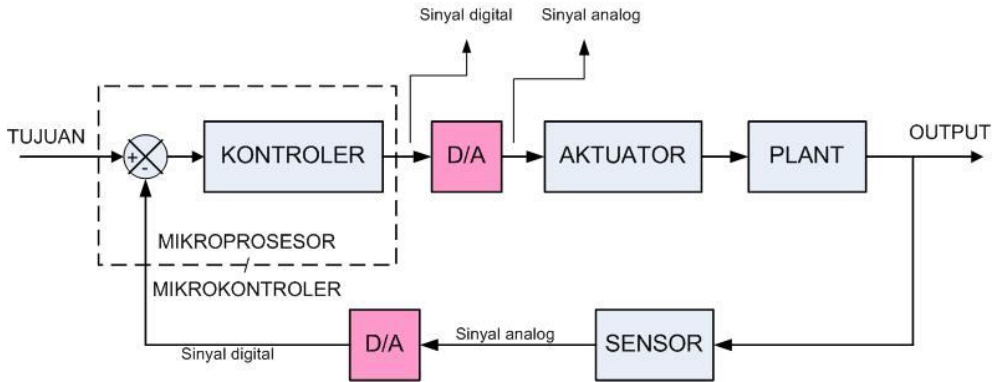
Gambar 9.109 Rangkaian elektronik untuk sistem kontrol suhu engine

Dari masing-masing penjelasan di atas, maka bisa kita gabungkan rangkaian lengkap dari sistem kontrol analog suhu engine seperti terlihat pada gambar 9.109.

Dalam sistem kontrol digital, pengontrol menggunakan rangkaian digital. Kerap kali, rangkaian ini sesungguhnya adalah komputer yang biasanya berbasis mikroprosesor atau mikrokontroler. Komputer tersebut melaksanakan program yang berulang berkali-kali (setiap perulangan disebut *iterasi* atau *scan*).

Program memerintahkan kepada komputer untuk membaca data dari sensor yang sebelumnya diolah dulu dari sinyal tegangan analog ke sinyal digital, lalu menggunakan bilangan-bilangan ini untuk menghitung output pengontrol (yang kemudian dikirim kepada aktuator). Program tersebut lalu memutar balik ke permulaan dan memulai lagi. Waktu total untuk satu kali melintasi program mungkin kurang dari 1 milidetik (ms). Sistem

digital hanya "melihat" inputnya pada saat tertentu dalam suatu scan dan memperbarui outputnya pada saat yang lain. Jika input berubah sejenak setelah komputer melihatnya, perubahan itu masih tidak terdeteksi sampai waktu berikutnya melintasi scan. Hal ini secara mendasar berbeda daripada sistem analog, yang bersifat kontinu dan menanggapi setiap perubahan secara segera. Meskipun demikian, pada kebanyakan sistem kontrol digital, waktu scan sedemikian singkat dibandingkan waktu tanggapan proses yang dikontrol sehingga, untuk semua tujuan praktis, tanggapan pengontrol terasa seketika. Dunia fisik pada dasarnya adalah "alam analog". Gejala-gejala alamiah membutuhkan waktu untuk terjadi, dan biasanya mereka bergerak secara sinambung dari satu posisi ke posisi berikutnya. Oleh karena itu, kebanyakan sistem kontrol akan mengendalikan proses-proses



Gambar 9.110 Blok diagram sistem kontrol digital dengan loop tertutup dengan menggunakan mikroprosesor atau mikrokontroler

analog. Hal ini berarti bahwa, pada banyak kasus, sistem kontrol digital mula-mula harus mengubah data input analog dari dunia-nyata menjadi bentuk digital sebelum data tersebut dapat dipergunakan. Begitu pula, output dari pengontrol digital harus diubah dari bentuk digital kembali menjadi bentuk analog. Gambar 9.110 memperlihatkan diagram blok dari sistem kontrol kalang-tertutup digital. Perhatikan dua blok tambahan: pengubah digital-ke-analog (digital to analog converter atau DAC) dan pengubah analog-ke-digital (analog to digital converter atau ADC). (Piranti-piranti ini, yang mengubah data di antara format digital dan analog, dibahas dalam bagian 9.6.2.2.4). Juga perhatikan garis umpan-balik diperlihatkan langsung menuju ke pengontrol. Hal ini menekankan kenyataan bahwa komputer, bukan rangkaian pengurangan yang terpisah, yang melakukan perbandingan di antara sinyal referensi dan sinyal umpan-balik.

## 9.7 Pemrograman Sistem Kontrol Elektronik

Konsep pemrograman terstruktur memegang peran penting dalam merancang, menyusun, memelihara dan mengembangkan suatu program, khususnya program aplikasi yang besar dan kompleks.

Konsep ini diungkapkan pertama kali pada tahun 1960-an oleh Profesor Edsger Dijkstra dari Universitas Eindhoven. Profesor Dijkstra mengungkapkan bahaya dari penggunaan instruksi peralihan proses tanpa syarat tertentu (GOTO, salah satu sintaks bahasa pascal) dalam pembuatan segala bentuk program karena akan menjadikan program tidak terstruktur dengan baik. Oleh sebab itu mulailah dikembangkan teknik pemrograman terstruktur. Namun demikian penerapan teknik ini tidak semata-mata hanya untuk menghindari penggunaan instruksi GOTO, tetapi untuk menciptakan program yang terstruktur dan sistematis.

Sebelum mempelajari bahasa pemrograman terstruktur lebih lanjut, ada beberapa istilah mendasar yang perlu dipahami lebih dahulu, yaitu:

1. *Program* adalah kata, ekspresi, pernyataan atau kombinasinya yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman sehingga dapat dieksekusi oleh komputer.
2. *Bahasa pemrograman* merupakan prosedur/tata cara penulisan program. Pada bahasa pemrograman terdapat dua faktor penting, yaitu sintax dan semantik. *Sintay* (sintaks) adalah aturan-aturan gramatikal yang mengatur tata cara penutisan kata, ekspresi dan pernyataan, sedangkan *semantik* adalah aturan-aturan untuk menyatakan suatu arti.
3. *Pemrograman* merupakan proses menaimplementasikan urutan langkah untuk menyelesaikan suatu masalah dengan menggunakan suatu Bahasa pemrograman. Algoritma berasal dari kata algoris dan ritmis, yang pertama kali diungkapkan oleh Abu Ja'far Mohammed Ibn Musa al Khowarismi (825 M) dalam buku *Al-Jabr Wa-al Muqabla*. Dalam bidang pemrograman algoritma didefinisikan sebagai suatu metode khusus yang tepat dan terdiri dari serangkaian langkah yang terstruktur dan dituliskan secara sistematis yang akan dikerjakan untuk menyelesaikan suatu masalah dengan bantuan komputer.

4. *Pemrograman terstruktur* merupakan proses mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program yang memiliki rancang bangun yang terstruktur dan tidak berbelit-belit sehingga mudah ditelusuri, dipahami dan dikembangkan oleh siapa saja.

Teknik pemrograman terstruktur harus memiliki ciri atau karakteristik sebagai berikut :

1. mengandung algoritma pemecahan masalah yang tepat, benar, sederhana, standar dan efektif;
2. memiliki struktur logika dan struktur program yang benar dan mudah dipahami serta menghindari penggunaan instruksi GOTO.
3. membutuhkan biaya testing, pemeliharaan dan pengembangan yang rendah;
4. memiliki dokumentasi yang baik;

Standar bahasa pemrograman yang baik dibutuhkan untuk menciptakan suatu program yang baik yang memiliki portabilitas yang tinggi sehingga memudahkan dalam merancang dan merawat program serta meningkatkan efektivitas penggunaan peralatan komputer. Untuk menentukan standar program yang baik dibutuhkan beberapa standar sebagai dasar penilaian, seperti:

1. teknik pemecahan masalah,
2. penyusunan program,
3. perawatan program,
4. standar prosedur.

Standar-standar tersebut sering dilihat oleh pemrogram sebagai batasan kreativitas dan kemampuan untuk menuangkan berbagai ide ke

dalam bentuk program. Namun standar tersebut akan membuat program menjadi konsisten dan mudah untuk dikembangkan.

Komputer adalah mesin yang dapat melaksanakan seperangkat perintah dasar (*instruction set*). Komputer hanya dapat diberi perintah yang terdiri dari perintah-perintah dasar tersebut. Perintah-perintah yang lebih rumit (misalnya mengurutkan suatu daftar sesuai abjad) harus diterjemahkan menjadi serangkaian perintah-perintah dasar yang dapat dimengerti komputer (perintah-perintah yang termasuk dalam *instruction set* komputer tersebut) yang pada akhirnya dapat menyelesaikan tugas yang diinginkan, meskipun dijalankan dengan beberapa operasi dasar, bukan satu operasi rumit.

### 9.7.1 Teknik Merancang Program

Dalam merancang sebuah program, seringkali para pembuat program menganggap sebuah program rancangannya sudah selesai jika program tersebut telah berjalan dengan baik. Namun dalam aplikasinya, seringkali sebuah program yang dirancang, perlu ditelusuri lagi untuk keperluan pengembangan lebih lanjut ataupun digabungkan dengan program lain dalam kerja sama tim.

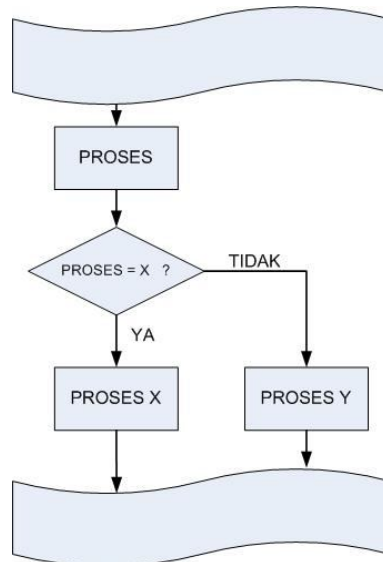
Pihak lain dalam tim seringkali membutuhkan cara kerja dari program rancangan tersebut untuk menggabungkan atau mengembangkan lebih lanjut program rancangan yang telah dianggap selesai. Untuk itu, sebuah program yang baik tidak hanya dapat berjalan dengan baik saja, namun program tersebut harus

dapat ditelusuri kembali dengan mudah. Bahkan pembuat program sendiri seringkali menemui kesulitan dalam menelusuri program tersebut jika program rancangan itu tidak dirancang dengan struktur yang teratur.

Teknik merancang sebuah program yang mempunyai struktur yang baik, biasanya diawali dengan pembuatan *flowchart* (diagram alir), pseudo code dan program assembly yang dirancang. Diagram alir digunakan untuk menggambarkan terlebih dahulu mengenai "apa yang harus dikerjakan" sebelum mulai merancang program.

Hal yang penting diketahui sebelum membuat suatu program adalah *algoritma*. *Algoritma* berasal dari kata *algoris* dan *ritmis*, yang pertama kali diungkapkan oleh Abu Ja'far Mohammed Ibn Musa al Khwarismi (825 M) dalam buku *Al-Jabr Wa-al Muqabla*. Dalam bidang pemrograman algoritma didefinisikan sebagai suatu metode khusus yang tepat dan terdiri dari serangkaian langkah yang terstruktur dan dituliskan secara sistematis yang akan dikerjakan untuk menyelesaikan suatu masalah dengan bantuan komputer.

Algoritma merupakan pola pikir terstruktur yang berisi tahap-tahap penyelesaian masalah yang dapat disajikan dengan teknik tulisan maupun dengan gambar. Penyajian algoritma dalam bentuk tulisan biasanya menggunakan metode *structure english* atau *pseudo code* (kode semu). Kebanyakan orang menyajikan algoritma dalam bentuk gambar. Penyajian algoritma dalam bentuk gambar biasanya menggunakan *flowchart* (diagram alir). Selain flow chart sebenarnya



Gambar 9.111 Contoh diagram alir atau flowchart

masih ada beberapa metode untuk menggambarkan algoritma yaitu structure chart, HIPO (*hierarchy input process output*), *data flow diagram*, dan lain-lain.

Istilah *pseudo code* (kode semu) merupakan metode yang cukup efisien untuk menggambarkan suatu algoritma. *Pseudo code* dituliskan dengan menggunakan bahasa yang mudah dipahami (boleh menggunakan bahasa Indonesia) agar alur logika yang digambarkan dapat dimengeti oleh orang awam sekalipun yang tidak mengerti masalah keteknikan, terutama tentang bahasa pemrograman itu sendiri..

*Pseudo code* (kode semu) disusun dengan tujuan untuk menggambarkan tahap-tahap penyelesaian suatu masalah dengan kata-kata (teks). Metode ini mempunyai kelemahan, dimana penyusunan algoritma dengan kode

semu sangat dipengaruhi oleh tata bahasa pembuatnya, sehingga kadang-kadang sulit dipahami oleh orang lain. Oleh karena itu kemudian dikembangkan suatu metode lain yang dapat menggambarkan suatu algoritma program secara lebih mudah dan sederhana yaitu dengan menggunakan flowchart (diagram alir). Terdapat 2 jenis flowchart yaitu sistem flowchart dan program flowchart.

Sistem flowchart merupakan diagram alir yang menggambarkan suatu sistem peralatan komputer yang digunakan dalam proses pengolahan data serta hubungan antar peralatan tersebut. Sistem flow chart tidak digunakan untuk menggambarkan urutan langkah untuk memecahkan masalah, tetapi hanya untuk menggambarkan prosedur dalam sistem yang dibentuk.



Seperti terlihat contoh *flowchart* pada gambar 9.111, bisa dijelaskan sebagai berikut : setelah proses selesai dijalankan, hasil dari proses diperiksa. Jika hasil dari proses adalah X, jalankan proses X. Namun, jika hasil dan proses bukan X, proses Y yang dijalankan.

Pseudo code biasanya digambarkan dalam Bahasa C atau Pascal yang lebih manusiawi daripada bahasa assembly seperti pada contoh berikut di mana pseudo code digambarkan dalam Bahasa Pascal :

```
If Hasil = X then Begin
  ProsesX End
Else Begin ProsesY End;
```

Struktur dari sebuah program biasanya terdiri atas:

### 1) Pernyataan

Pernyataan menggambarkan dasar mekanisme dari melakukan sesuatu, seperti pemberian nilai pada variabel atau memanggil subroutine berikut:

- ✓ Mov A,#00H ;memberi nilai pada akumulator
- ✓ Call SerialOut ;memanggil subroutine Serialout

### 2) Putaran

Struktur putaran digunakan untuk melakukan sebuah operasi yang dilakukan berulang-ulang.

### 3) Pilihan

Dua buah pernyataan yang seringkali digunakan untuk memilih dari dua kemungkinan yang ada.

## 9.5.2 Bahasa Pemrograman

Bahasa pemrograman adalah bahasa yang dapat diterjemahkan

menjadi kumpulan perintah-perintah dasar tersebut. Penerjemahan dilakukan oleh program komputer yang disebut kompilator (*compiler*). Setiap bahasa pemrograman mempunyai kompilatornya sendiri. Contoh dari kompilator ini adalah kompilator C++ tidak akan mengerti program yang ditulis dengan bahasa Java. Sintaks dari bahasa pemrograman lebih mudah dipahami oleh manusia daripada sintaks perintah dasar. Namun tentu saja komputer hanya dapat melaksanakan perintah dasar itu. Maka di sinilah peran penting kompilator sebagai perantara antara bahasa pemrograman dengan perintah dasar.

Kegiatan membuat program komputer dengan menggunakan bahasa pemrograman disebut pemrograman komputer. Contoh bahasa pemrograman adalah bahasa Asembler, FORTRAN, COBOL, BASIC, JAVA, dan C++.

Bahasa pemrograman berfungsi sebagai media untuk menyusun dan memahami suatu program komputer serta sebagai alat komunikasi antara programmer dengan komputer. Bahasa pemrograman dapat digolongkan menjadi beberapa tingkatan yaitu :

### 1. Bahasa tingkat rendah (Low Level Language)

Bahasa tingkat rendah merupakan bahasa pemrograman yang berorientasi pada mesin. Disebut tingkat rendah karena bahasa ini lebih dekat ke bahasa mesin daripada bahasa manusia. Yang tergolong dalam bahasa tingkat rendah adalah bahasa assembly. Kelemahan bahasa tingkat rendah antara lain :

- a. Sulit dipelajari karena programmer harus mengetahui seluk beluk perangkat keras yang digunakan
- b. Bahasa assembly untuk satu jenis mikroprosesor satu dengan yang lain sangat jauh berbeda karena belum ada standardisasi. Contoh : bahasa *assembly* untuk mikroprosesor Intel 8088 dengan bahasa *assembly* untuk mikroprosesor Z-80 sangat jauh berbeda.
- c. Fungsi-fungsi yang tersedia sangat terbatas, misalnya tidak ada fasilitas untuk pemrograman grafik, fungsi-fungsi numerik & string, dll.

Selain memiliki kelemahan seperti tersebut di atas, bahasa assembly memiliki beberapa kelebihan yang tidak dimiliki oleh bahasa lain yaitu :

- a. Kecepatan eksekusi dari program yang ditulis dengan bahasa assembly sangat tinggi, paling cepat dibanding dengan program-program yang dibuat menggunakan bahasa yang lain
- b. *Executable file* yang dihasilkan oleh bahasa assembly ukurannya paling kecil. Buktinya : hampir semua program virus yang banyak beredar adalah dibuat dengan menggunakan bahasa assembly karena ukurannya paling kecil, sehingga kehadiran virus tersebut menjadi lebih sulit terdeteksi.

## 2. Bahasa tingkat tinggi (*High Level Language*)

Bahasa tingkat tinggi lebih dekat ke bahasa manusia dari pada bahasa mesin. Bahasa tingkat tinggi merupakan bahasa pemrograman yang memiliki aturan-aturan gramatikal dalam penulisan ekspresi

atau pernyataan dengan standar yang mudah dipahami oleh manusia.

Yang tergolong bahasa tingkat tinggi antara lain : BASIC, Fortran, COBOL, Pascal, Prolog, C, dll. Ada sebagian para pakar yang menyebut bahasa C sebagai bahasa tingkat menengah (*middle level language*), karena dianggap bahasa C adalah bahasa tingkat tinggi yang mempunyai kelebihan hampir menyamai bahasa assembly karena kelengkapan fungsinya dalam mengakses perangkat keras. Kelebihan bahasa tingkat tinggi antara lain :

- a. Mudah dipelajari.
- b. Mempunyai fasilitas *trace & debug* untuk mendeteksi adanya kesalahan (*error*).
- c. Mempunyai fungsi/library yang lengkap sehingga dapat mempermudah dan mempercepat pembuatan program.

### 9.5.3 Pemrograman dengan Asembler untuk mikroprosesor/mikrokontroler

Secara fisik, kerja dari sebuah mikrokontroler dapat dijelaskan sebagai siklus pembacaan instruksi yang tersimpan di dalam memori. Mikrokontroler menentukan alamat dari memori program yang akan dibaca, dan melakukan proses baca data di memori. Data yang dibaca diinterprestasikan sebagai instruksi. Alamat instruksi disimpan oleh mikrokontroler di register, yang dikenal sebagai program counter. Instruksi ini misalnya program aritmatika yang melibatkan 2 register. Sarana yang ada dalam program assembly sangat minim, tidak seperti dalam bahasa pemrograman tingkat atas (*high level language programming*) semuanya sudah siap

pakai. Penulis program *assembly* harus menentukan segalanya, menentukan letak program yang ditulisnya dalam memori-program, membuat data konstan dan tabel konstan dalam memori-program, membuat variabel yang dipakai kerja dalam memori-data dan lain sebagainya.

Program sumber *assembly* (*assembly source program*) merupakan kumpulan dari baris-baris perintah yang ditulis dengan program penyunting teks (*text editor*) sederhana, misalnya program EDIT.COM dalam DOS, atau program NOTEPAD dalam Windows atau MIDE-51. Kumpulan baris-perintah tersebut biasanya disimpan ke dalam file dengan nama ekstensi \*.ASM dan lain sebagainya, tergantung pada program Assembler yang akan dipakai untuk mengolah program-sumber *assembly* tersebut. Setiap baris perintah merupakan sebuah perintah yang utuh, artinya sebuah perintah tidak mungkin dipecah menjadi lebih dari satu baris. Satu baris perintah bisa terdiri atas 4 bagian, bagian pertama dikenali sebagai label atau sering juga disebut sebagai symbol, bagian kedua dikenali sebagai kode operasi, bagian ketiga adalah operand dan bagian terakhir adalah komentar. Antara bagian-bagian tersebut dipisahkan dengan sebuah spasi atau tabulator.

Bagian label dijelaskan seperti dibawah ini. Label dipakai untuk memberi nama pada sebuah baris-perintah, agar bisa mudah menyebitnya dalam penulisan program. Label bisa ditulis apa saja asalkan diawali dengan huruf, biasa panjangnya tidak lebih dari 16 huruf. Huruf-huruf berikutnya boleh merupakan angka atau tanda titik dan

tanda garis bawah. Kalau sebuah baris-perintah tidak memiliki bagian label, maka bagian ini boleh tidak ditulis namun spasi atau tabulator sebagai pemisah antara label dan bagian berikutnya mutlak tetap harus ditulis.

Dalam sebuah program sumber bisa terdapat banyak sekali label, tapi tidak boleh ada label yang kembar. Sering sebuah baris perintah hanya terdiri dari bagian label saja, baris demikian itu memang tidak bisa dikatakan sebagai baris perintah yang sesungguhnya, tapi hanya sekedar memberi nama pada baris bersangkutan. Bagian label sering disebut juga sebagai bagian symbol, hal ini terjadi kalau label tersebut tidak dipakai untuk menandai bagian program, melainkan dipakai untuk menandai bagian data.

Berikutnya adalah bagian kode operasi. Kode operasi (*operation code* atau sering disingkat sebagai *OpCode*) merupakan bagian perintah yang harus dikerjakan. Dalam hal ini dikenal dua macam kode operasi, yang pertama adalah kode operasi untuk mengatur kerja mikrokontroler. Jenis kedua dipakai untuk mengatur kerja program *assembler*, sering dinamakan sebagai *assembler directive*. Kode operasi ditulis dalam bentuk mnemonic, yakni bentuk singkatan-singkatan yang relatif mudah diingat, misalnya adalah MOV, ACALL, RET dan lain-lain. Kode-operasi ini ditentukan oleh pabrik pembuat mikrokontroler, dengan demikian setiap prosesor mempunyai kode operasi yang berlainan. Kode operasi berbentuk *mnemonic* tidak dikenal mikrokontroler, agar program yang ditulis dengan kode *mnemonic* bisa dipakai untuk mengendalikan prosesor, program

```

1  ===== PROGRAM BY TRIGAS BAUMANTO =====
2  ===== PHONE +628563154194 =====
3  ===== INSTRUCTOR OF VEHICLE CONTROL SYSTEM =====
4  ===== AT AUTOTRONICS, VEDC MALANG =====
5
6
7  LABEL  OP-CODE  OPERAND  KOMENTAR
8
9
10
11
12          ORG      0H
13
14  START :
15          MOV      A, #11111110B ; ISI AKUMULATOR
16          MOV      R0, #7D      ; ISI R0 DENGAN 7
17
18  KIRI :
19          MOV      P0, A        ; COPY A KE P0
20          CALL     DELAY       ; PANGGIL DELAY
21          RL       A           ; PUTAR KE KIRI DARI NILAI ACC 1 BIT
22          DEC      R0          ; R0 = R0 - 1
23          CJNE    R0, #0, KIRI ; DIBANDINGKAN, JIKA R0 TDK SM DGN 0, KE START
24          SJMP    START       ; LONCAT KE LABEL START
25
26  DELAY: ; SUBROUTINE DELAY
27          MOV      R1, #255D
28  ULANG1:
29          MOV      R2, #255D
30  ULANG2:
31          DJNZ    R2, ULANG2
32          DJNZ    R1, ULANG1
33          RET
34

```

Gambar 9.112 Contoh bentuk program assembly yang umum

semacam itu diterjemahkan menjadi program yang dibentuk dari kode-operasi kode-biner, yang dikenali oleh mikroprosesor/mikrokontroler.

Tugas penerjemahan tersebut dilakukan oleh program yang dinamakan sebagai Program Assembler. Di luar kode-operasi yang ditentukan pabrik pembuat mikroprosesor/mikrokontroler, ada pula kode-operasi untuk mengatur kerja dari program assembler, misalnya dipakai untuk menentukan letak program dalam memori (ORG), dipakai untuk membentuk variabel (DS), membentuk tabel dan data konstan (DB, DW) dan lain sebagainya.

Bagian lainnya adalah bagian operand. Operand merupakan pelengkap bagian kode operasi, namun tidak semua kode operasi memerlukan operand, dengan demikian bisa terjadi sebuah baris perintah hanya terdiri dari kode operasi tanpa operand. Sebaliknya ada pula kode operasi yang perlu lebih dari satu operand, dalam hal ini antara operand satu dengan yang lain dipisahkan dengan tanda koma.

Bentuk operand sangat bervariasi, bisa berupa kode-kode yang dipakai untuk menyatakan Register dalam prosesor, bisa berupa nomor-memori (alamat memori) yang dinyatakan dengan bilangan atau pun

nama label, bisa berupa data yang siap di-operasi-kan. Semuanya disesuaikan dengan keperluan dari kode-operasi.

Untuk membedakan operand yang berupa nomor-memori atau operand yang berupa data yang siap di-operasi-kan, dipakai tanda-tanda khusus atau cara penulisan yang berlainan.

Di samping itu operand bisa berupa persamaan matematis sederhana atau persamaan Boolean, dalam hal semacam ini program Assembler akan menghitung nilai dari persamaan-persamaan dalam operand, selanjutnya merubah hasil perhitungan tersebut ke kode biner yang dimengerti oleh prosesor. Jadi perhitungan di dalam operand dilakukan oleh program assembler bukan oleh prosesor.

Bagian yang terakhir adalah bagian komentar. Bagian komentar merupakan catatan-catatan penulis program, bagian ini meskipun tidak mutlak diperlukan tapi sangat membantu masalah dokumentasi. Membaca komentar-komentar pada setiap baris-perintah, dengan mudah bisa dimengerti maksud tujuan baris bersangkutan, hal ini sangat membantu orang lain yang membaca program.

Pemisah bagian komentar dengan bagian sebelumnya adalah tanda spasi atau tabulator, meskipun demikian huruf pertama dari komentar sering-sering berupa tanda titik-koma, merupakan tanda pemisah khusus untuk komentar.

Untuk keperluan dokumentasi yang intensip, sering-sering sebuah baris yang merupakan komentar saja, dalam hal ini huruf pertama dari baris bersangkutan adalah tanda titik-koma.

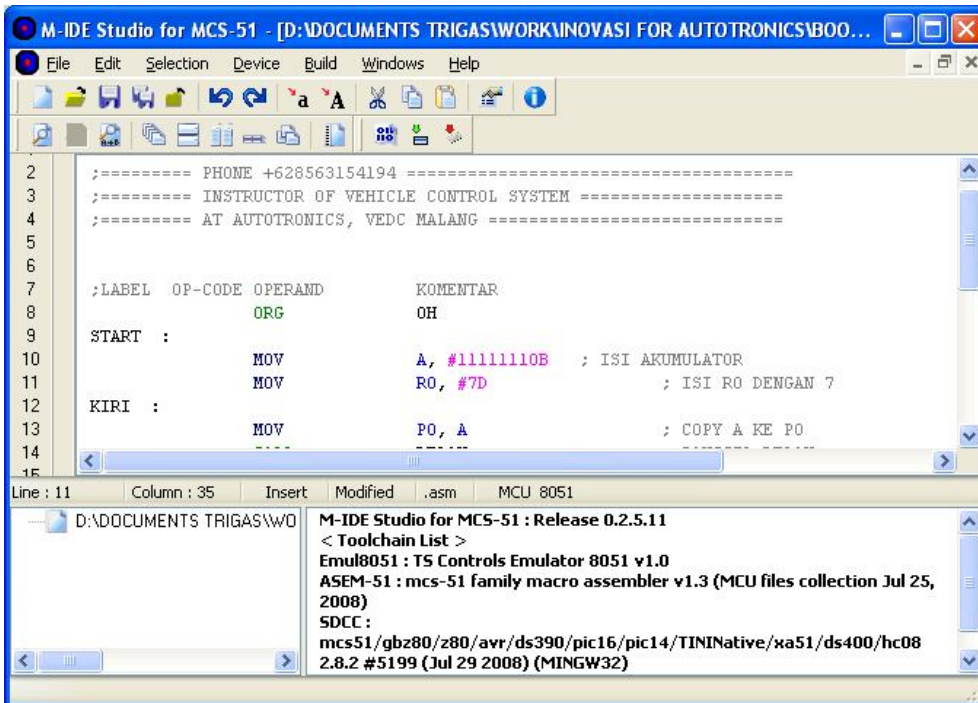
AT89S51 memiliki sekumpulan instruksi yang sangat lengkap. Instruksi MOV untuk byte dikelompokkan sesuai dengan mode pengalamatan (addressing modes). Mode pengalamatan menjelaskan bagaimana operand dioperasikan. Berikut penjelasan dari berbagai mode pengalamatan. Bentuk program assembly yang umum ialah seperti terlihat pada gambar 9.112.

Isi memori ialah bilangan heksadesimal yang dikenal oleh mikrokontroler kita, yang merupakan representasi dari bahasa assembly yang telah kita buat. Mnemonic atau opcode ialah kode yang akan melakukan aksi terhadap operand. Operand ialah data yang diproses oleh opcode. Sebuah opcode bisa membutuhkan 1, 2 atau lebih operand, kadang juga tidak perlu operand. Sedangkan komentar dapat kita berikan dengan menggunakan tanda titik koma (;). Berikut contoh jumlah operand yang berbeda beda dalam suatu *assembly*:

- CJNE R5,#22H,label1; 3 operand.
- MOVX @DPTR,A; 2 operand;
- RL A ; 1 buah operand
- NOP ;tidak memerlukan operand

#### 9.5.4 Perangkat Lunak Pemrograman Mikrokontroler

Program yang telah selesai dibuat disimpan dengan ekstension .asm. Lalu kita dapat membuat program objek dengan ekstension HEX dengan menggunakan compiler MIDE-51. MIDE-51 merupakan software freeware yang bisa diambil dengan bebas di internet. Tampilan software ini bisa dilihat seperti pada gambar 9.113.



Gambar 9.113 Tampilan software MIDE-51 sebagai editor assembler dan compilernya.pada desktop komputer

Program sumber *assembly* di atas, setelah selesai ditulis diserahkan ke program assembler (kompilator) untuk diterjemahkan. Setiap prosesor mempunyai program assembler (kompilator) tersendiri, bahkan satu macam prosesor bisa memiliki beberapa macam program Assembler buatan pabrik perangkat lunak yang berlainan.

Hasil utama pengolahan program Assembler adalah program-obyek. Program-obyek ini bisa berupa sebuah file tersendiri, berisikan kode-kode yang siap dikirimkan ke memori program mikroprosesor atau mikrokontroler, tapi ada juga program-obyek yang disisipkan pada program-sumber *assembly* seperti

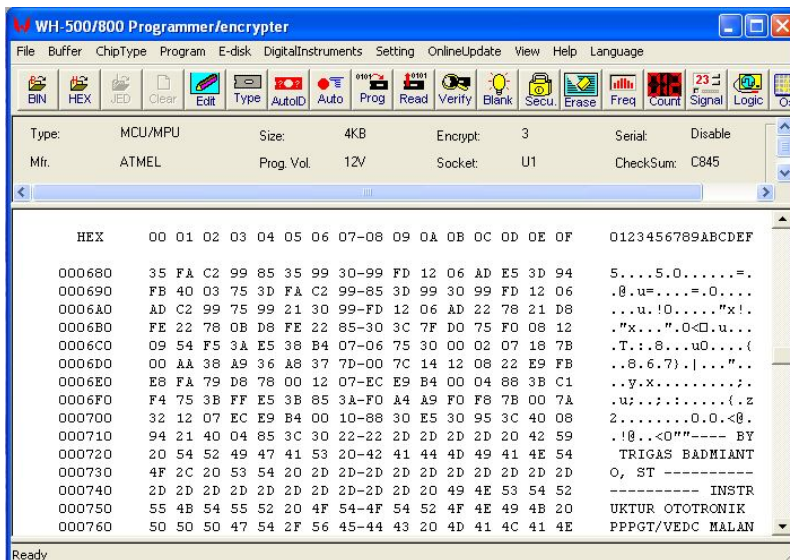
terlihat dalam *Assembly Listing* di Gambar 9.113.

Bagian kanan Gambar 9.113 merupakan program sumber *assembly* karya asli penulis program, setelah diterjemahkan oleh program Assembler kode-kode yang dihasilkan berikut dengan nomor-nomor memori tempat penyimpanan kode-kode tadi, disisipkan pada bagian kiri setiap baris perintah, sehingga bentuk program ini tidak lagi dikatakan sebagai program-sumber *assembly* tapi dikatakan sebagai *Assembly Listing*.

Membaca *Assembly Listing* bisa memberikan gambaran yang lebih jelas bagi program yang ditulis, bagi pemula *Assembly Listing* memberi



Gambar 9.114 Contoh hardware downloader yang dijual dipasaran



Gambar 9.115 Tampilan software dari downloader Brightek WH500-A

pengertian yang lebih mendalam tentang isi memori-program, sehingga bisa lebih dibayangkan bagaimana kerja dari sebuah program. Dan dengan membaca file assembly listing (berekstensi .lst) maka bisa kita lihat pada bagian mana program kita yang terjadi kesalahan.

Setelah didapatkan file \*.hex, maka diperlukan proses untuk memasukkan file tersebut ke dalam memori mikrokontroler. Antara software dan hardware, harus merupakan satu paket, dimana komunikasi antara software dengan hardware harus sesuai. Ada banyak

peralatan yang dijual di pasaran. Tetapi ada juga yang bisa kita dapatkan secara cuma-cuma dari internet. Contoh peralatan yang dijual dipasaran adalah Brightek WH-500A. Bentuk hardware dan tampilan software, bisa diamati pada gambar 9.114 dan gambar 9.115.

## 9.8 Minimum Sistem Mikrokontroler dan *Electronic Control Unit (ECU)*

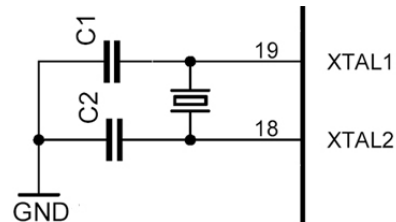
Setelah proses seperti dijelaskan pada bagian 9.8 selesai, yaitu proses memasukkan file \*.hex, biasa disebut proses download dari komputer ke mikrokontroler, maka langkah selanjutnya adalah menjalankan rangkaian yang telah kita buat. Untuk itu diperlukan rangkaian yang bisa menjalankan program yang telah ditanam di dalam memori mikrokontroler. Mikrokontroler yang kita bahas mempunyai tipe AT89Sxx. Buatan dari ATMEL dan masih masuk kelompok MCS-51.

Rangkaian dengan komponen yang paling minimal yang mampu menjalankan program di dalam memori sesuai dengan apa yang dikehendaki, disebut dengan minimum sistem mikrokontroler. Ada 3 kelompok utama yang harus ada ketika kita membuat minimum sistem ini, yaitu :

### 1. Rangkaian oscillator

Komponen yang diperlukan adalah kristal dengan nilai tidak lebih dari 33 MHz. Sedangkan C1 dan C2 bernilai  $30 \text{ pF} \pm 10 \text{ pF}$ . Rangkaian seperti ditunjukkan pada gambar 9.116. Dirangkai dengan mikrokontroler pada pin nomor 18 (XTAL2) dan 19 (XTAL1). Rangkaian ini untuk membangkitkan clock, kalau

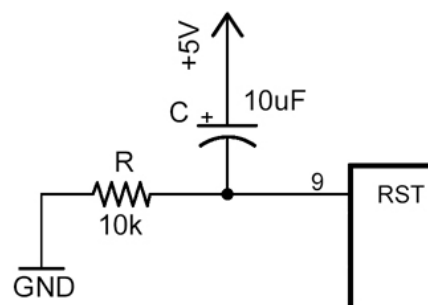
pada *personal komputer* maka fungsinya sama dengan prosessor (pentium 1, 2, 3 atau pentium 4). Semakin besar harga kristalnya, maka akan semakin cepat pula eksekusi programnya.



Gambar 9.116 Rangkaian oscillator

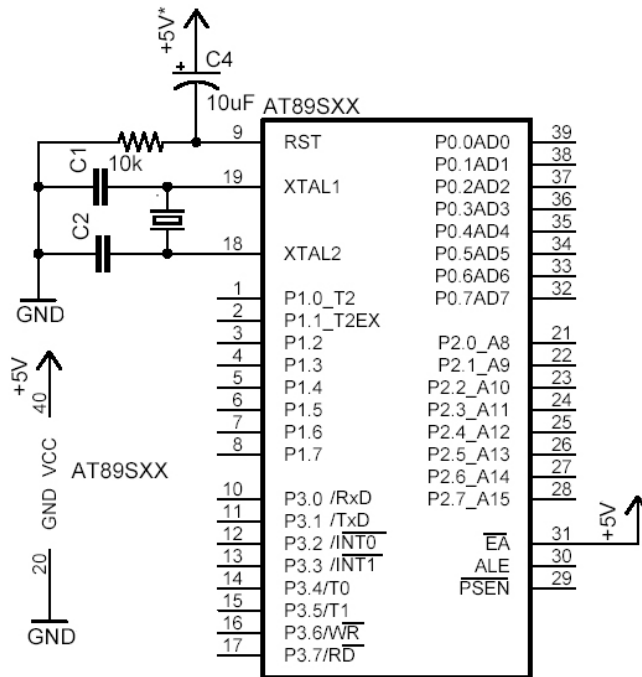
### 2. Rangkaian Reset

Semua kaki input/output (I/O) akan reset ke kondisi awal jika kaki RST (kaki 9) dibuat high. Penahanan kaki RST pada kondisi high selama 2 cycle mesin selama oscillator sedang jalan, akan membuat reset mikrokontroler. Pengaruh ke program adalah eksekusi program akan kembali ke yang paling awal. Agar mikrokontroler bisa reset secara otomatis pada saat awalnya, maka diperlukan komponen resistor dan kapasitor yang dirangkai dengan RST (pin 9), Vcc dan ground, seperti diperlihatkan pada gambar 9.117.



Gambar 9.117 Rangkaian reset otomatis





Gambar 9.118 Minimum sistem mikrokontroler MCS-51

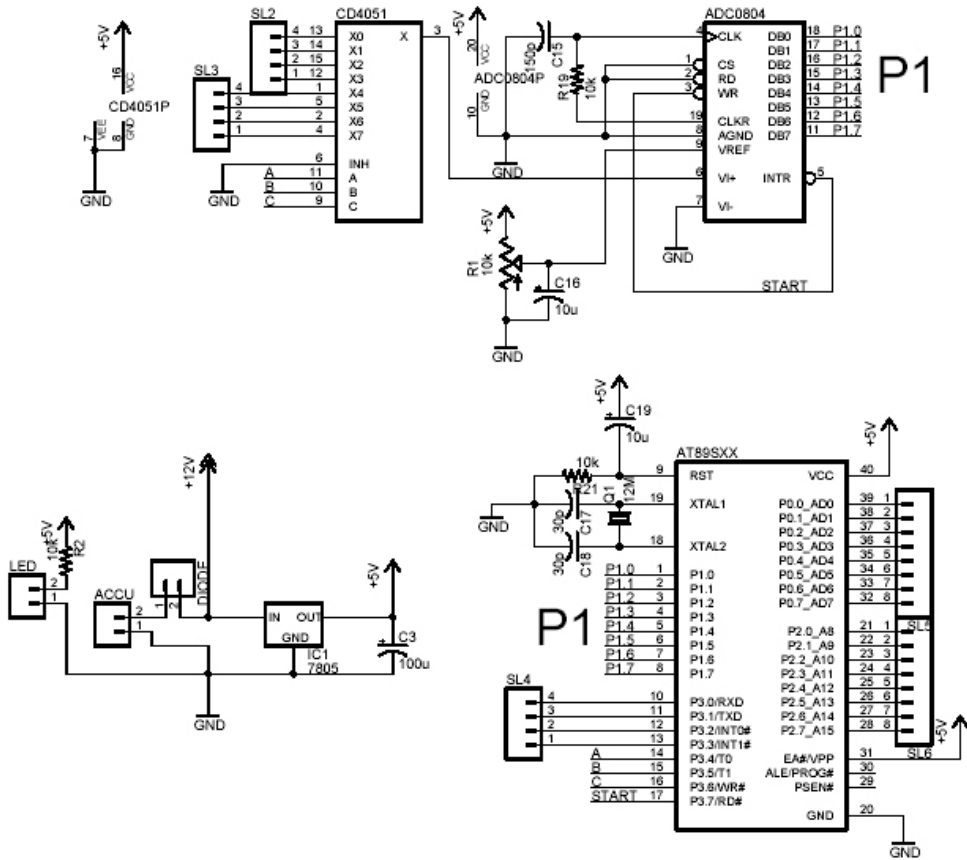
3. *Power supply dan rangkaian untuk memakai memory eksternal atau tidak.*

Pada pin 40 diberi tegangan 5 Volt. Mikrokontroler bisa berfungsi dengan baik jika diberi sumber tegangan sebesar 4 Volt s/d 5.5 Volt. Sedangkan pada pin 20 disambungkan dengan ground rangkaian. Jika pin 31 (EA) diberi tegangan low, berarti dipakai untuk megekskusi program yang berada diluar, misalnya ditaruh di memori eksternal. Sedangkan apabila menjalankan program internal, maka pin ini harus diberi tegangan high, atau disambung langsung dengan Vcc.

Secara lengkap, minimum sistem mikrokontroler ditunjukkan oleh gambar 9.118. Dengan merangkai mikrokontroler MCS-51 dengan 1 bu-

ah resistor, 3 buah kapasitor dan 1 buah kristal dan tentunya diberi sumber tegangan, maka mikrokontroler ini sudah bisa berfungsi dan bisa difungsikan sebagai kontroler pada sistem kontrol tertentu.

Untuk membentuk minimum sistem *Electronic Control Unit (ECU)*, maka perlu ditambahi dengan rangkaian pengkondisi sinyal digital. Dalam hal ini perlu ditambahi ADC sebagai masukan digital ke mikrokontroler. Gambar 9.119 menunjukkan rangkaian minimum sistem *ECU*. Tampak bahwa di gambar 9.119 terdapat ADC untuk menghubungkan sensor-sensor analog dengan mikrokontroler. Terdapat juga IC7805 yang berfungsi untuk menghasilkan tegangan 5 Volt



Gambar 9.119 Minimum sistem *Electronic Control Unit (ECU)* pada kendaraan

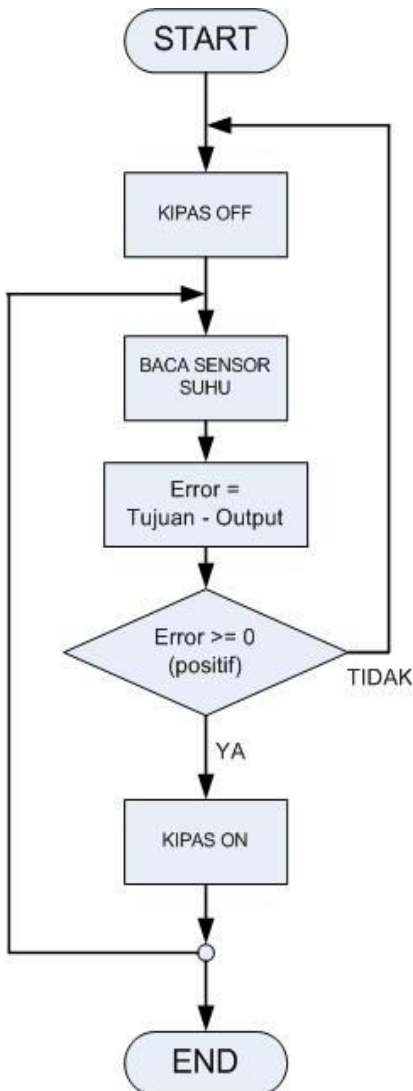
yang konstan untuk men-suplay rangkaian. Agar mikrokontroler bisa membaca sensor lebih dari satu, maka perlu ditambahkan multiplexer analog (IC-CD4051), yang berfungsi untuk menyalurkan banyak saluran ke satu keluaran dengan adanya selector yang berfungsi untuk memilih saluran input mana yang akan diteruskan.

Aplikasi sistem kontrol suhu engine dijelaskan pada bagian di bawah. Seperti pada bagian 9.7 di atas maka bisa dibuat blok diagram sistemnya seperti pada gambar 9.103, hanya saja yang membedakan

dengan sistem kontrol analog adalah terletak pada kontrolernya. Kalau sistem kontrol analog menggunakan op-amp sebagai kontrolernya, maka untuk sistem digital digunakan minimum sistem *ECU* untuk menyelesaikan masalah ini secara digital.

Output sensor suhu disambungkan dengan salah satu input dari multiplexer (IC-CD4051). Dimana keluaran IC-CD4052 (pin 3) disambungkan dengan input ADC, kemudian diolah menjadi sinyal digital 8 bit yang bisa langsung dibaca oleh mikrokontroler. Sedangkan untuk kipas, memerlukan 1 bit digital out,

tu dipakai P2.0. Flowchart untuk sistem ini bisa diamati pada gambar 9.120. Sedangkan program asemblernya bisa dilihat pada gambar 9.121. Dengan membuat flowchart terlebih dahulu, maka seorang programer akan lebih mudah untuk membuat program asemblernya. Demikian juga apabila suatu saat, orang lain ingin



Gambar 9.120 Flowchart sistem kontrol suhu engine

```

1  ;==PROGRAM SISTEM KONTROL SUHU==
2
3
4  ;-----BY TRIGAS BADMIANTO, ST -----
5  ;-----INSTRUKTUR OTOTRONIK PPPGT/VEDC MALANG--
6  ;-----HP : +628563154194-----
7
8
9  ;KETERANGAN DENGAN HARDWARE   :
10 ;1. P1      : INPUT ADC
11 ;2. P3.7    : START ADC
12 ;3. P3.6    : SELEKTOR C (MUX)
13 ;4. P3.5    : SELEKTOR B (MUX)
14 ;5. P3.4    : SELEKTOR A (MUX)
15 ;6. P2.0    : DIGITAL OUT KE KIPAS
16 ;7. X0(MUX) : SENSOR SUHU
17
18
19 SUHU EQU 30H
20
21 ORG 0H
22
23 MAIN1 :
24 CLR P2.0 ; KIPAS OFF
25
26 MAIN2 :
27 CALL BACA_SENSOR
28 MOV A,#102D ; 2 V = 102D (TUJUAN)
29 SUBB A,SUHU ; A = A - SUHU
30 JC MAIN1 ; IF NEGATIF, KE MAIN1
31 SETB P2.0 ; KIPAS ON
32 JMP MAIN2
33
34 BACA_SENSOR :
35 CLR P3.6 ; C = LOW, XO TERUS
36 CLR P3.5 ; B = LOW, XO TERUS
37 CLR P3.4 ; B = LOW, XO TERUS
38 CLR P3.7 ;
39 SETB P3.7 ; START ADC
40 MOV R0,#200D ; DELAY KONVERSI ADC
41 DJNZ R0,$ ; DELAY KONVERSI ADC
42 MOV SUHU,P1 ; SUHU = P1
43 RET
44
45 END
    
```

Gambar 9.121 Asembler dibuat berdasarkan flowchart sistem kontrol suhu engine.

mengembangkan program tersebut, maka orang lain itu akan lebih mudah dengan membaca *flowchart* untuk memahami sistem daripada langsung membaca assembler yang sudah jadi.

*Asembler* yang nampak pada gambar 9.121 ini, ditulis di editor dengan memakai MIDE-51 atau notepad, kemudian di compile agar didapatkan file hex. Kemudian dengan memakai downloader, maka file hex ini dimasukkan ke memori mikrokontroler. Dengan membuat rangkaian seperti pada gambar 9.119, maka terbentuklah sistem kontrol suhu engine secara digital. Rangkaian ini yang sering disebut sebagai *Electronic Control Unit (ECU)*.