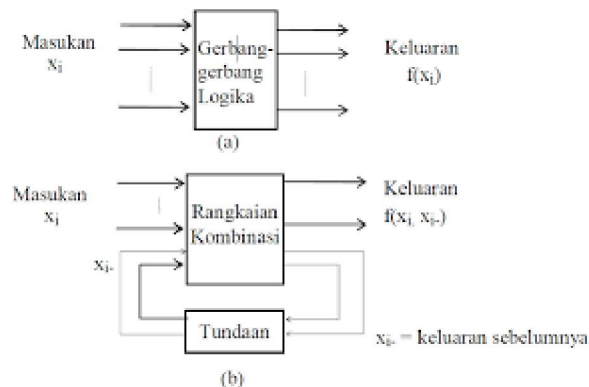


## PERTEMUAN 4

### RANGKAIAN GERBANG KOMBINASI

"Semua *rangkaian logika* dapat digolongkan atas dua jenis, yaitu rangkaian kombinasi (combinational circuit) dan rangkaian berurut (sequential circuit). Perbedaan kedua jenis rangkaian ini terletak pada sifat keluarannya. Keluaran suatu rangkaian kombinasi setiap saat hanya ditentukan oleh masukan yang diberikan saat itu. Keluaran rangkaian berurut pada setiap saat, selain ditentukan oleh masukannya saat itu, juga ditentukan oleh keadaan keluaran saat sebelumnya, jadi juga oleh masukan sebelumnya. Jadi, rangkaian berurut tetap mengingat keluaran sebelumnya dan dikatakan bahwa rangkaian ini mempunyai ingatan (memory). Kemampuan mengingat pada rangkaian berurut ini diperoleh dengan memberikan tundaan waktu pada lintasan balik (umpan balik) dari keluaran ke masukan. Secara diagram blok, kedua jenis rangkaian logika ini dapat digambarkan seperti pada Gambar 1." (Albert Paul Malvino, Ph.D.)



Gambar 3. Model Umum Rangkaian Logika

(a) Rangkaian Kombinasi

(b) Rangkaian Berurut **PERANCANGAN RANGKAIAN KOMBINASI**

"*Rangkaian kombinasi* mempunyai komponen-komponen masukan, *rangkaian logika*, dan keluaran, tanpa umpan balik. Persoalan yang dihadapi dalam perancangan (design) suatu *rangkaian kombinasi* adalah memperoleh fungsi Boole beserta diagram rangkaianannya dalam bentuk susunan gerbang-gerbang. Seperti telah diterangkan sebelumnya, fungsi Boole merupakan hubungan aljabar antara masukan dan keluaran yang

diinginkan. Langkah pertama dalam merancang setiap rangkaian logika adalah menentukan apa yang hendak direalisasikan oleh rangkaian itu yang biasanya dalam bentuk uraian kata-kata (verbal). Berdasarkan uraian kebutuhan ini ditetapkan jumlah masukan yang dibutuhkan serta jumlah keluaran yang akan dihasilkan. Masing-masing masukan dan keluaran diberi nama simbolis. Dengan membuat tabel kebenaran yang menyatakan hubungan masukan dan keluaran yang diinginkan, maka keluaran sebagai fungsi masukan dapat dirumuskan dan disederhanakan dengan cara-cara yang telah diuraikan dalam bab-bab sebelumnya.

Berdasarkan persamaan yang diperoleh ini, yang merupakan fungsi *Boole* dari pada rangkaian yang dicari, dapat digambarkan diagram rangkaian logikanya. Ada kalanya fungsi *Boole* yang sudah disederhanakan tersebut masih harus diubah untuk memenuhi kendala yang ada seperti jumlah gerbang dan jenisnya yang tersedia, jumlah masukan setiap gerbang, waktu perambatan melalui keseluruhan gerbang (tundaan waktu), interkoneksi antar bagian-bagian rangkaian, dan kemampuan setiap *gerbang* untuk mencatu (drive) *gerbang* berikutnya. Harga rangkaian logika umumnya dihitung menurut cacah gerbang dan cacah masukan keseluruhannya. Ini berkaitan dengan cacah gerbang yang dikemas dalam setiap kemasan.

***Gerbang-gerbang logika*** yang tersedia di pasaran pada umumnya dibuat dengan teknologi *rangkaian terpadu* (Integrated Circuit, IC). Pemaduan (integrasi) *gerbang-gerbang dasar seperti NOT, AND, OR, NAND, NOR, XOR* pada umumnya dibuat dalam skala kecil (Small Scale Integration, SSI) yang mengandung 2 sampai 6 gerbang dalam setiap kemasan. Kemasan yang paling banyak digunakan dalam rangkaian logika sederhana berbentuk DIP (Dual-In-line Package), yaitu kemasan dengan pen-pen hubungan ke luar disusun dalam dua baris sejajar. Kemasan gerbang-gerbang dasar umumnya mempunyai 14-16 pin, termasuk pin untuk catu daya positif dan nol (Vcc dan Ground). Setiap gerbang dengan 2 masukan membutuhkan 3 pin (1 pin untuk keluaran) sedangkan gerbang 3 masukan dibutuhkan 4 pin. Karena itu, satu kemasan 14 pin dapat menampung hanya 4 gerbang 2 masukan atau 3 gerbang 3 masukan.

Dalam praktek kita sering terpaksa menggunakan *gerbang-gerbang* yang tersedia di

pasaran yang kadang-kadang berbeda dengan kebutuhan rancangan kita. *Gerbang* yang paling banyak tersedia di pasaran adalah gerbang-gerbang dengan 2 atau 3 masukan. Umpamanya, dalam rancangan kita membutuhkan gerbang dengan 4 atau 5 masukan dan kita akan mengalami kesulitan memperoleh gerbang seperti itu. Karena itu kita harus mengubah rancangan sedemikian sehingga rancangan itu dapat direalisasikan dengan *gerbang-gerbang* dengan 2 atau 3 masukan. Kemampuan pencatutan daya masing-masing gerbang juga membutuhkan perhatian. Setiap gerbang mampu mencatu hanya sejumlah tertentu gerbang lain di keluarannya (disebut sebagai fan-out). Ini berhubungan dengan kemampuan setiap gerbang dalam menyerap dan mencatu arus listrik. Dalam perancangan harus kita yakinkan bahwa tidak ada gerbang yang harus mencatu terlalu banyak gerbang lain di keluarannya. Ini sering membutuhkan modifikasi rangkaian realisasi yang berbeda dari rancangan semula. Mengenai karakteristik elektronik *gerbang-gerbang logika* dibahas dalam Lampiran A." (Albert Paul Malvino, Ph.D.)


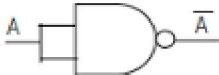

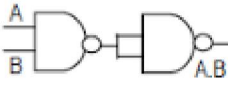

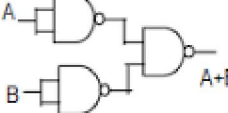
## ***IMPLEMENTASI RANGKAIAN GERBANG LOGIKA DENGAN GERBANG NAND***

### ***Gerbang NAND (NOT And)***

" *Gerbang NAND* dan NOR merupakan gerbang universal, artinya hanya dengan menggunakan jenis gerbang NAND saja atau NOR saja dapat menggantikan fungsi dari 3 gerbang dasar yang lain (*AND*, *OR*, *NOT*). Multilevel, artinya: dengan mengimplementasikan *gerbang NAND* atau NOR, akan ada banyak level / tingkatan mulai dari sistem input sampai ke sisi output. Keuntungan pemakaian NAND saja atau NOR saja dalam sebuah rangkaian digital adalah dapat mengoptimalkan pemakaian seluruh gerbang yang terdapat dalam sebuah IC, sehingga menghemat biaya

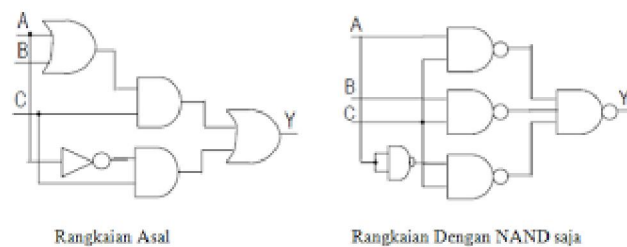
*Gerbang NAND* adalah pengembangan dari *gerbang AND*. Gerbang ini sebenarnya adalah *gerbang AND* yang pada outputnya dipasang *gerbang NOT*. Gerbang yang paling sering digunakan untuk membentuk rangkaian kombinasi adalah gerbang NAND dan NOR, dibanding dengan AND dan OR. Dari sisi aplikasi perangkat luar, *gerbang NAND* dan NOR lebih umum sehingga gerbang-gerbang tersebut dikenal sebagai gerbang yang

"universal". *Gerbang-gerbang NOT, AND dan OR* dapat di-substitusi ke dalam bentuk NAND saja, dengan hubungan seperti gambar 2.

	Bentuk Asal	Dengan NAND
<b>NOT</b>		
<b>AND</b>		
<b>OR</b>		

Gambar 4. *Substitusi Beberapa Gerbang Dasar Menjadi NAND*

Rangkaian Asal Rangkaian Dengan NAND saja



Gambar 5, implemtasi Gergang NAND

Gambar 5, impelemntasi *Gergang NAND*

Untuk mendapatkan persamaan dengan menggunakan *NAND* saja, maka persamaan asal harus dimodifikasi sedemikian rupa, sehingga hasil akhir yang didapatkan adalah persamaan dengan **NAND** saja. *Gerbang NAND* sangat banyak di pakai dalam computer modern dan mengeti pemakaiannya sangat berharga bagi kita, untuk merancang jaringan *gerbang NAND* ke **NAND**, gunakan prosedur tabel kombinasi untuk ungkapan jumlah hasil kali,

Dalam perancangan *logika, gerbang logika* siskrit tidak selalu digunakan ttapi biasanya beisi banyak gerbang, karena itu, biasanya lebih disukai untuk memanfaatkan satu jenis gerbang, dan bukan campuran beberapa gerbang untuk alasan ini konversi gerbang digunakan untuk menyatukan suatu fungsi gerbang tertentu dengan cara mengombinasikan beberapa gerbang yang bertipe sama, suatu misal implementasi *gerbang NAND* ke dalam *gerbang NO, gerbang AND dan gerbang OR* (Kf Ibrahim, "Teknik Digital")

Pertimbangan lain nya dalam impelementasi fungs *boole* berkaitan dengan jenis gate yang digunakan, seringkali di rasakan perlu nya untuk mengimplimentasikan fungsi boole dengan hanya menggunakan gate-gate **NAND** saja, walaupun mungkin tidak merupakan implementasi gate minimum, teknik tersebut memiliki keuntungan dan keteraturan yang dapat menyederhanakan proses pembuatan nya di pabrik. (wiliam steling).

### *Decoder*

" *Decoder adalah* suatu rangkaian *logika* kombinasional yang mampu mengubah masukan kode biner  $n$ -bit ke  $m$ -saluran keluaran sedemikian rupa sehingga setiap saluran keluaran hanya satu yang akan aktif dari beberapa kemungkinan kombinasi masukan. *Gamba* 2.14 memperlihatkan diagram dari decoder dengan masukam  $n = 2$  dan keluaran  $m = 4$  ( decoder 2 ke 4). Setiap  $n$  masukan dapat berisi *logika* 1 atau 0, ada  $2^N$  kemungkinan kombinasi dari masukan atau kode-kode. Untuk setiap kombinasi masukan ini hanya satu dari  $m$  keluaran yang akan aktif (berlogika 1), sedangkan keluaran yang lain adalah berlogika 0. Beberapa decoder didisain untuk menghasilkan keluaran low pada keadaan aktif, dimana hanya keluaran low yang dipilih akan aktif sementara keluaran yang lain adalah berlogika 1. Dari keadaaan aktif keluaranya, decoder dapat dibedakan atas "non inverted output" dan "inverted output". (David Bucchlah, Wayne McLahan)