

DATA & INFORMASI DIGITAL

F. Rizal Batubara

 **USU**press
Art Design, Publishing & Printing

2013

USUpress

Art Design, Publishing & Printing

Jl. Universitas No. 9, Gedung Pusat Sistem Informasi (Gedung F)
Kampus USU, Medan 20155
Indonesia

Telp. 061-8213737

Kunjungi kami di:

<http://usupress.usu.ac.id>

Terbitan pertama 2013

© USUpress 2013

Hak cipta dilindungi oleh undang-undang; dilarang memperbanyak,
menyalin, merekam sebagian atau seluruh bagian buku ini dalam
bahasa atau bentuk apapun tanpa izin tertulis dari penerbit.

ISBN 979 458 708 7

Perpustakaan Nasional: Katalog Dalam Terbitan (KDT)

Batubara, F Rizal

Data dan Informasi Digital/F. Rizal Batubara

Cet. 1. Medan : USUpress, 2013.

iv, 145 p. ; illus. : 29,7 cm

Bibliografi, Indeks

ISBN : 979-458-708-7

1. DATA, INFORMASI

I. Judul

003'.54 - dc22

Dicetak di Medan, Indonesia

PENGANTAR

Data merupakan fakta dan angka-angka yang relative tidak berarti, yang diubah menjadi informasi oleh suatu pengolah informasi. Informasi memiliki arti bagi pemilikinya. Pengolah informasi menyediakan informasi dalam bentuk lisan maupun tertulis. Informasi berasal dari sumber-sumber internal maupun lingkungan dan digunakan untuk membuat keputusan untuk memecahkan masalah.

Informasi adalah data yang telah diberi makna melalui konteks. Informasi merupakan pesan (ucapan atau ekspresi) atau kumpulan pesan yang terdiri dari order sekuens dari simbol, atau makna yang dapat ditafsirkan dari pesan atau kumpulan pesan. Informasi dapat direkam atau ditransmisikan. Hal ini dapat dicatat sebagai tanda-tanda, atau sebagai sinyal berdasarkan gelombang. Informasi adalah jenis acara yang mempengaruhi suatu negara dari sistem dinamis. Informasi bisa di katakan sebagai pengetahuan yang didapatkan dari pembelajaran, pengalaman, atau instruksi. Dalam beberapa hal pengetahuan tentang peristiwa-peristiwa tertentu atau situasi yang telah dikumpulkan atau diterima melalui proses komunikasi, pengumpulan intelejen, ataupun didapatkan dari berita juga dinamakan informasi. Informasi yang berupa koleksi data dan fakta seringkali dinamakan informasi statistik. Dalam bidang ilmu komputer, informasi adalah data yang disimpan, diproses, atau ditransmisikan.

Buku ini membahas bentuk data dan bagaimana menyusunnya sehingga menjadi informasi dalam format digital serta beberapa hal yang terkait dalam prosesnya. Diharapkan buku ini dapat membantu pemahaman data dan informasi digital, sehingga dapat digunakan dalam kehidupan sehari-hari dalam era teknologi informasi saat ini.

Medan, Januari 2013
Penulis,

F. Rizal Batubara, ST, MTI

DAFTAR ISI

I. BIT	1
1.1 BIT MATEMATIKA	2
1.2 BIT FISIK	5
1.3 BIT KLASIK.....	5
1.4 BIT KUANTUM	6
1.5. KEUNGGULAN QUBIT	8
II. KODE	9
2.1 UKURAN RUANG SIMBOL.....	10
2.2 PENGGUNAAN KAPASITAS CADANGAN	11
2.3 ISU STRATEGIS ADALAH PERANCANGAN KODE.....	15
2.4 KODE DENGAN PANJANG TERTENTU DAN PANJANG VARIABEL	17
2.5 DETAIL: ASCII.....	19
2.6 DETAIL: KODE INTEGER.....	22
2.7 DETAIL: KODE GENETIK.....	24
2.8 DETAIL: <i>IP ADDRESS</i>	28
III. KOMPRESI	30
KOMPRESI LZW.....	31
IV. ERROR	35
4.1 PERLUASAN MODEL SISTEM	35
4.2 BAGAIMANA DAPAT TERJADI KESALAHAN ?.....	36
4.3 DETEKSI VS KOREKSI.....	36
4.4 JARAK HAMMING	37
4.5 BIT TUNGGAL	37
4.6 BIT GANDA.....	39
4.7 KODE BLOK	41
4.8 KODE LANJUTAN	42
V. PROBABILITAS	43
5.1 RUANG KEADAAN	44
5.2 HASIL YANG DIKETAHUI	46
5.3 HASIL YANG TIDAK DIKETAHUI	46
5.4 <i>JOINT EVENT</i> DAN PROBABILITAS BERSYARAT	47
5.5 NILAI RATA-RATA	49
5.6 INFORMASI.....	50
5.7 SIFAT-SIFAT INFORMASI	52
5.8 <i>CODING</i> DENGAN SUMBER EFISIEN	53
5.9 DETAIL: KODE SUMBER YANG EFISIEN	54
5.10 DETAIL : CONTOH TINGKAT KEMATIAN	59
VI. KOMUNIKASI	60
6.1 MODEL SUMBER	61

6.2	ENTROPI SUMBER.....	62
6.3	TEOREMA PENGKODEAN SUMBER	64
6.4	MODEL KANAL.....	65
6.5	TEORI KANAL TANPA GANGGUAN.....	66
6.6	KANAL <i>NOISY</i> (RAMAI)	67
6.7	DALIL KAPASITAS KANAL BER- <i>NOISE</i>	71
6.8	PEMBALIKAN (REVERSIBILITAS)	72
6.9	RINCIAN : KEPERLUAN SISTEM KOMUNIKASI.....	72
VII.	PROSES	74
7.1	MODEL PROSES	75
7.2	INFORMASI, RUGI-RUGI DAN NOISE.....	78
7.3	CONTOH DETERMINISTIK	79
7.4	KAPASITAS	80
7.5	OBSERVASI.....	81
7.6	PROSES YANG DI- <i>CASCADE</i>	81
VIII.	KESIMPULAN.....	82
8.1	PENILAIAN (ESTIMASI).....	83
8.2	CONTOH	85
8.3	STRATEGI PENGAMBILAN KESIMPULAN.....	89
IX.	PRINSIP ENTROPI MAKSIMUM: BENTUK SEDERHANA	91
9.1	SUSUNAN MASALAH	92
9.2	PROBABILITAS.....	93
9.3	ENTROPI	94
9.4	PEMBATAS	94
9.5	ENTROPI MAKSIMUM: BENTUK ANALISA	95
X.	PRINSIP-PRINSIP ENTROPI MAKSIMUM.....	98
10.1	RUMUSAN MASALAH.....	98
10.2	PROBABILITAS	101
10.3	ENTROPI.....	101
10.4	PEMBATASAN.....	102
10.5	ENTROPI MAKSIMUM: BENTUK ANALITIK	103
10.6	ENTROPI MAKSIMUM: BENTUK PENGALI LAGRANGE	103
10.7	CONTOH- CONTOH	107
XI.	SISTEM FISIK	109
11.1	SIFAT MEKANIKA KUANTUM	110
11.2	PENGALAN MEKANIKA KUANTUM.....	111
11.3	KEADAAN SEIMBANG	114
11.4	MODEL MULTI KEADAAN	115
11.5	SISTEM ENERGI.....	117
11.6	SISTEM INFORMASI.....	117
XII.	ENERGI	118
12.1	MODEL MAGNET <i>DIPOLE</i>	119
12.2	PRINSIP ENTROPY MAKSIMUM UNTUK FISIK SISTEM	119
12.3.	SISTEM DAN LINGKUNGAN	124

XIII. TEMPRATUR	134
13.1 MESIN PEMANAS.....	135
13.2 SIKLUS KONVERSI-ENERGI.....	137
XIV. INFORMASI QUANTUM	140
14.1 PENYIMPANAN INFORMASI QUANTUM	140
14.2 MODEL 1: BIT KECIL SEDERHANA.....	142
14.3 MODEL 2: DAERAH SUPERPOSISI (QUBIT).....	142
14.4 MODEL 3: PERKALIAN QUBIT DENGAN BELITAN	142
REFERENSI.....	145



BIT

Informasi diukur dalam bit, seperti halnya panjang diukur dalam meter dan waktu diukur dalam detik. Tentunya mengetahui sejumlah informasi tidak sama dengan memahami informasi itu sendiri, apa yang dimaksud ataupun apa kegunaannya. Dalam catatan ini kita tidak akan memikirkan isi atau makna dari informasi, tetapi hanya jumlah.

Skala panjang yang berbeda dibutuhkan dalam keadaan yang berbeda. Kadang kala kita ingin mengukur panjang dalam kilometer atau inci, bahkan terkadang dalam *ångström*. Serupa dengan hal tersebut, skala lain untuk informasi selain menggunakan skala bit kadang dibutuhkan; dalam hal sistem fisika, informasi sering diukur dalam joule per kelvin.

Bagaimana informasi diukur? Pertimbangkan situasi yang mungkin memenuhi beberapa hasil. Sebagai contoh adalah melempar koin (2 hasil, kepala atau ekor) atau memilih sebuah kartu dari satu set kartu (52 hasil yang mungkin). Bagaimana seseorang (misalnya Dewi) menyatakan hasil ini kepada orang lain (Reza)?

Pertama pertimbangkan kasus dua hasil dalam melempar koin, dan kita andaikan keduanya memiliki peluang sama. Jika Dewi hendak menyatakan hasil dari melempar koin ini ke Reza, dia dapat menggunakan beberapa cara yang mungkin, dalam hal jumlah informasi yang disampaikan, untuk mengatakan “kepala” atau “ekor”, atau untuk hal tersebut menyatakan 0 atau 1. Kita menyatakan bahwa informasi yang disampaikan adalah satu bit.

Jika Dewi melemparkan 2 koin, dia dapat menyatakan 4 hasil yang mungkin bisa terjadi, dengan menyatakan 0 atau 1 sebanyak dua kali. Dengan cara yang sama, hasil percobaan dengan 8 hasil yang mungkin dapat dinyatakan dalam 3 bit, atau umumnya 2^n dalam n bit. Jadi jumlah informasi adalah logaritma basis 2 dari jumlah hasil yang mungkin sama.

Perlu dicatat bahwa menyampaikan informasi membutuhkan 2 tahap. Tahap pertama adalah “*setup*” di mana Dewi dan Reza setuju tentang hal yang akan mereka komunikasikan, tepatnya tentang pengertian tiap urutan bit. Pengertian umumnya disebut *code*. Hal ini dilakukan sebelum hasil diperoleh. Tahap kedua adalah “komunikasi” di mana urutan 0 dan 1 yang sebenarnya dikirimkan. Urutan ini merupakan data. Kemudian menyampaikan pilihan kartu yang tepat dari kumpulan kartu, mungkin 00 untuk keriting, 01 keling, 10 hati, dan 11 untuk wajah. Dengan menggunakan kode tersebut Dewi menggambarkan kartu dan menyatakan

ke Reza kartu yang tepat dengan mengirimkan 2 bit data. Dia dapat mengulanginya untuk beberapa percobaan dengan menggunakan kode yang sama.

Setelah Reza tahu kartu yang digambarkan tapi sebelum menerima pesan dari Dewi, dia masih ragu dengan kartu yang tepat. Keraguan atau kekurangan informasi dapat juga dinyatakan dalam bit. Dengan mendengar hasilnya, keraguan Reza berkurang dengan diterimanya informasi. Keraguan Reza bertambah selama tahap “*setup*” dan kemudian berkurang selama tahap “komunikasi”.

Catatan penting tentang informasi, diantaranya yang digambarkan dalam contoh berikut:

- Informasi dapat dipelajari dari pengamatan, percobaan dan pengukuran.
- Informasi bersifat subjektif atau tergantung pada pengamat. Hal yang diketahui Dewi berbeda dengan yang diketahui oleh Reza (jika informasi tidak subjektif, maka tidak dibutuhkan adanya komunikasi).
- Ketidakpastian seseorang dapat bertambah pada pemahaman saat pengamatan tentang informasi yang mungkin tersedia dan kemudian dapat berkurang saat diterimanya informasi.
- Informasi dapat hilang, baik hilangnya data itu sendiri atau hilangnya kode.
- Bentuk fisik dari informasi ditempatkan di dalam ruang dan waktu. Konsekuensinya :
 - Informasi dapat dikirimkan dari suatu tempat ke tempat lain.
 - Informasi dapat disimpan dan dipanggil kembali kemudian.

1.1 Bit Matematika

Seperti yang telah kita lihat bahwa informasi dapat dikomunikasikan dengan urutan nilai 0 dan 1. Kekuatan abstrak ini membuat kita mengabaikan beberapa detil yang berkaitan dengan sistem pengolahan dan pengiriman informasi yang spesifik.

Bit begitu sederhana, hanya memiliki dua nilai yang mungkin dan konsep matematika yang digunakan untuk memanipulasi bit tunggal adalah mudah. Hal ini dikenal sebagai *boolean algebra*, yang ditemukan oleh ahli matematika George Boole (1815-1864). *Boolean algebra* dalam beberapa hal hampir serupa dengan aljabar *integer* atau bilangan *real* yang dipelajari saat di SMU, tapi ada pula perbedaannya dalam beberapa hal.

Aljabar berkaitan dengan variabel yang memiliki nilai tertentu yang mungkin dan dengan fungsi di mana bila dinyatakan dengan satu atau lebih variabel memberikan hasil kembali beberapa nilai tertentu yang mungkin. Sedangkan dalam *boolean algebra* nilai yang mungkin hanya 0 dan 1.

Terdapat empat fungsi variabel tunggal dalam *boolean algebra*. Salah satunya disebut *the identity*, yang dengan sederhana mengembalikan argumennya. Sedangkan yang lain disebut “*not*” atau ingkaran atau inversi atau pelengkap, mengubah 0 menjadi 1 dan seterusnya. Dua fungsi lainnya adalah 0 atau 1 tanpa melihat argumennya. Untuk lebih jelasnya perhatikan tabel berikut :

Tabel 1.1 Fungsi Boolean untuk Variabel Tunggal

x Argumen	f(x)			
	IDENTITY	NOT	ZERO	ONE
0	0	1	0	1
1	1	0	0	1

Perlu dicatat bahwa *boolean algebra* lebih sederhana daripada aljabar integer ataupun bilangan real, yang masing-masing memiliki banyak fungsi tidak terbatas untuk variabel tunggal.

Berapa banyak fungsi boolean untuk dua variabel A dan B? Masing masing dari dua argumen dapat mengambil dua nilai, sehingga terdapat empat kombinasi input yang mungkin. Ada 16 cara yang berbeda untuk memberikan 2 nilai boolean menjadi 4 input. Dari 16 cara tersebut, dua cara mengabaikan *input*, empat cara memberikan *output* untuk menjadi A atau B atau pelengkapnya dan 10 cara lainnya tergantung pada argumen keduanya. Yang paling sering digunakan adalah AND, OR, XOR, NAND, dan NOR seperti yang dapat dilihat pada Tabel 1.2.

Hal yang menarik bahwa nilai boolean 0 dan 1 sama halnya pada integer 0 dan 1; AND untuk perkalian dan OR untuk penjumlahan. Akan tetapi kesimpulan biasa dari aljabar asli tidak tergantung pada *boolean algebra*, sehingga penganalogian begitu saja dapat berbahaya. Hal yang perlu dilakukan adalah membedakan integer dari nilai boolean; walau kadang hal tersebut sulit.

Fungsi AND digambarkan serupa halnya dengan perkalian, dengan menuliskan tanda titik diantara dua nilai boolean, yaitu A AND B dituliskan A.B. Fungsi OR dituliskan dengan tanda tambah : A+B artinya A OR B. Negasi atau fungsi NOT ditunjukkan dengan garis di atas symbol jadi NOT A adalah \bar{A} . Terakhir, fungsi XOR dilambangkan dengan tanda tambah di dalam lingkaran, $A \oplus B$.

Table 1.2 Fungsi Boolean untuk 2 Variabel

x Argument	f(x)				
	AND	NAND	OR	NOR	XOR
00	0	1	0	1	0
01	0	1	1	0	1
10	0	1	1	0	1
11	1	0	1	0	1

Table 1.3 Simbol Logika Boolean

NOT	\bar{A}
AND	A.B
NAND	$\overline{A.B}$
OR	A+B
XOR	$A \oplus B$

Ada beberapa sifat umum fungsi boolean yang berguna. Hal ini dapat dibuktikan dengan menyederhanakannya menjadi sejumlah terbatas kombinasi nilai yang memungkinkan.

Sebuah fungsi dikatakan reversibel jika diketahui outputnya maka input dapat ditemukan. Dua dari empat fungsi untuk variabel tunggal bersifat reversibel dalam keadaan ini (kenyataannya malah *self-inverse*). Jelasnya, tidak ada fungsi dari dua (atau lebih) input dapat bersifat reversibel dengan sendirinya. Hal ini disebabkan oleh lebih banyaknya variabel input daripada variabel output. Tetapi beberapa kombinasi dari dua atau lebih fungsi demikian dapat saja bersifat reversibel. Sebagai contoh adalah fungsi eksklusif-or, XOR, $A \oplus B$ bersifat reversibel ketika diperbesar oleh fungsi yang mengembalikan argumen pertama.

Untuk fungsi dari dua variabel, ada beberapa sifat yang perlu dipertimbangkan. Sebagai contoh jika A, B dan C adalah variabel boolean dapat menjadi 0 atau 1, lalu fungsi AND adalah komutatif karena $A \cdot B = B \cdot A$. Beberapa dari 16 fungsi tersebut adalah komutatif. Beberapa sifat dari *boolean algebra* dirangkum dalam table 1.4.

Ada beberapa notasi yang digunakan pada *boolean algebra*. Kadang AND, OR dan NOT ditunjukkan dalam bentuk AND (A,B), OR(A,B), dan NOT (A). Kadang notasi sisipan digunakan ketika $A \wedge B$ ditandai $A \cdot B$, $A \vee B$ ditandai $A + B$, dan $\sim A$ ditandai \bar{A} . *Boolean algebra* sangat berguna dalam matematika logika, ketika tanda $A \wedge B$ untuk $A \cdot B$, $A \vee B$ untuk $A + B$, dan $\sim A$ untuk \bar{A} biasanya digunakan.

Tabel 1.4 Sifat Boolean algebra

<i>Idempotent</i>	$A \cdot A = A$ $A + A = A$	<i>Absorption</i>	$A \cdot (A + B) = A$ $A + (A \cdot B) = A$
<i>Complement</i>	$A \cdot \bar{A} = 0$ $A + \bar{A} = 1$ $A \oplus A = 0$ $A \oplus \bar{A} = 1$	<i>Assosiative</i>	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$ $A + (B + C) = (A + B) + C$ $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
<i>Minimum</i>	$A \cdot 1 = A$ $A \cdot 0 = 0$	<i>Unnamed theorem</i>	$A \cdot (\bar{A} + B) = A \cdot B$ $A + (\bar{A} \cdot B) = A + B$
<i>Maximum</i>	$A + 0 = A$ $A + 1 = 1$	<i>De Morgan</i>	$\bar{A} \cdot \bar{B} = \overline{A + B}$ $\bar{A} + \bar{B} = \overline{A \cdot B}$
<i>Commutative</i>	$A \cdot B = B \cdot A$ $A + B = B + A$ $A \oplus B = B \oplus A$ $\overline{A \cdot B} = \overline{B \cdot A}$ $\overline{A + B} = \overline{B + A}$	<i>Distributive</i>	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ $A + (B \cdot C) = (A + B) \cdot (A + C)$

1.2 Bit Fisik

Jika informasi untuk disimpan atau ditransfer, maka informasi harus memiliki bentuk fisik. Perangkat yang bisa menyimpan bit harus memiliki dua keadaan yang berbeda, salah satunya diinterpretasikan sebagai 0, yang lain adalah 1. Bit disimpan dengan meletakkan perangkat dalam salah satu status tersebut. Dan ketika bit dibutuhkan, maka status perangkat tersebut diukur. Jika perangkat bergerak dari satu tempat ke tempat lain maka komunikasi telah terjadi. Jika perangkat bertahan untuk beberapa waktu lamanya maka perangkat tersebut berfungsi sebagai memori. Jika perangkat mengubah statusnya secara acak maka perangkat tersebut telah kehilangan nilai awalnya.

Secara natural kita tertarik pada bentuk fisik perangkat yang kecil. Batasan seberapa kecil objek yang dapat dan masih menyimpan bit informasi dapat diperoleh dari mekanika quantum. *Quantum bit* atau *qubit* adalah objek yang dapat menyimpan bit tunggal tetapi hal tersebut begitu kecilnya untuk menjadi subjek pembatasan pengukuran mekanika kuantum.

1.3 Bit Klasik

Dalam sistem elektronik saat ini, informasi dibawa oleh ribuan objek, yang semua itu dipersiapkan dengan cara yang sama (atau paling tidak dengan suatu cara yang sesuai). Jadi sebuah memori semikonduktor menyimpan sebuah bit tunggal dengan menggunakan keberadaan ataupun ketidakhadiran dari ribuan elektron yang mungkin. Dengan cara yang sama sejumlah besar foton digunakan pada komunikasi radio.

Karena banyak objek yang terlibat, maka pengukuran atas mereka tidak terbatas pada sebuah bentuk ya atau tidak, tetapi malahan dapat melewati suatu rentang kesatuan nilai. Jadi tegangan pada sebuah elemen logika semikonduktor dapat berupa nilai berapa saja di dalam rentang, katakanlah, dari 0V hingga 5V. Tegangan dapat ditafsirkan dengan memenuhi margin kesalahan, sehingga tegangan antara 0V hingga 1V akan merepresentasikan logika 0 dan tegangan antara 4V dan 5V adalah sebuah logika 1. Sebuah rangkaian tidak akan menjamin bahwa interpretasi tegangan antara 1V dan 4V adalah tepat.

Jika *noise* dalam sebuah rangkaian selalu lebih kecil dari 1V, dan *output* setiap gerbang rangkaian adalah 0V atau 5V, maka tegangan dapat selalu ditafsirkan sebagai bit tanpa kesalahan. Rangkaian jenis ini menunjukkan apa yang dikenal sebagai “logika yang diperbaiki” (*restoring logic*) yaitu sejak deviasi kecil yang dieleminasi sebagai informasi diolah. Kekuatan dari komputer modern tergantung pada penggunaan *restoring logic* tersebut.

Bit klasik merupakan sebuah abstraksi di mana suatu bit dapat diukur tanpa membingungkannya. Sebagai hasilnya, salinan sebuah bit klasik dapat dibuat (sedangkan suatu qubit tidak dapat disalin). Abstraksi ini berguna pada sirkuit yang menggunakan *restoring logic*.

Karena semua sistem fisika pada akhirnya mematuhi mekanika quantum, bit klasik selalu menjadi suatu cara penaksiran yang ideal untuk mewujudkannya.

Pertanyaan yang paling menarik adalah apakah abstraksi bit klasik akan berlanjut hingga bermanfaat seperti teknologi semikonduktor yang mengurangi ukuran komponen. Pada akhirnya, sebagaimana kita mencoba mengontrol bit dengan sejumlah kecil atom atau foton, untuk membatasi peran mekanika quantum akan menjadi suatu hal yang penting. Sulit untuk mengatakan dengan tepat kapan hal tersebut akan terjadi, tetapi beberapa orang percaya bahwa itu akan ada sebelum tahun 2015.

1.4 Bit Kuantum

Menurut mekanika kuantum, adalah mungkin bagi objek kecil untuk memiliki dua keadaan yang dapat diukur. Dugaan tersebut merupakan hal yang sempurna untuk menyimpan bit. Bagaimanapun, jika dua keadaan tersebut memiliki energi yang sama yang dihubungkan dengan mereka, maka merupakan hal yang mungkin untuk mempersiapkan objek sehingga memiliki kombinasi dari dua keadaan tersebut. Jadi, apa yang akan diukur?

Dalam konteks klasik, pengukuran dapat menentukan kombinasi tersebut dengan tepat. Lebih lanjut, untuk mencapai ketepatan yang lebih besar, pengukuran dapat diulangi dan hasil yang diperoleh dirata-ratakan. Tetapi bagaimanapun konteks kuantum adalah hal yang berbeda. Dalam pengukuran kuantum, yang menjadi pertanyaan adalah apakah objek berada di dalam beberapa keadaan khusus atau tidak. Sebagai jawabannya adalah selalu dengan “ya” atau “tidak”; tidak pernah dengan jawaban “mungkin”, atau tidak juga dengan misalnya 27% ya -73% tidak. Lebih lanjut, setelah pengukuran, sistem berakhir pada keadaan yang berkaitan dengan jawaban tersebut. Jadi pengukuran lebih lanjut tidak akan memberikan tambahan informasi. Hasil beberapa pengukuran khusus tidak dapat diperkirakan sebelumnya, tetapi kemungkinan dari dua jawaban bisa saja terjadi, yaitu dalam terminologi probabilitas. Sifat alami mekanika kuantum yang khas tersebut memberikan baik pembatasan tentang bagaimana informasi dapat dibawa oleh sebuah qubit tunggal maupun kesempatan untuk merancang sistem yang memanfaatkan keuntungan dari fitur tersebut.

Kita akan mengilustrasikan quantum bit dengan sebuah contoh. Kita ambil sebagai qubit adalah foton yang merupakan partikel dasar pada radiasi elektromagnetika seperti pada radio, televisi dan lampu. Foton adalah kandidat yang bagus untuk membawa informasi dari satu tempat ke tempat lain karena kecil dan dapat bergerak cepat.

Foton memiliki medan listrik dan magnet yang berorientasi secara simultan. Arah medan listrik disebut arah polarisasi (kita tidak akan mempertimbangkan foton yang berpolarisasi melingkar). Jadi jika foton dibawa dalam arah-z, maka medan listrik dapat dalam arah-x, arah-y atau kenyataannya dalam sejumlah arah lainnya dalam bidang x-y atau pada “bidang horisontal-vertikal”.

Polarisasi dapat digunakan untuk menyimpan bit informasi. Dengan demikian Dewi dapat menyiapkan foton dengan polarisasi horisontal jika bit adalah 0 dan polarisasi vertikal jika bit adalah 1. Kemudian ketika Reza menerima foton, dia dapat mengukur polarisasi vertikalnya

(diandaikan jika polarisasi vertikal). Jika jawabannya adalah ya, maka Reza mengambil kesimpulan bahwa bit adalah 1.

Dapat pula dipikirkan bahwa lebih dari sebuah bit tunggal informasi dapat ditransmisi oleh sebuah polarisasi foton tunggal. Mengapa Dewi tidak dapat mengirim dua bit dengan menggunakan sudut polarisasi yang berbeda dari horisontal dan vertikal? Mengapa tidak menggunakan horisontal, vertikal, setengah jalan diantaranya yang dimiringkan ke kanan, ataupun setengah jalan diantaranya yang dimiringkan ke kiri. Yang menjadi masalah adalah bahwa Reza harus menentukan sudut yang akan diukur. Dengan adanya pembatasan mekanika kuantum, Reza tidak dapat untuk menanyakan pertanyaan “apa sudut polarisasinya” tetapi hanya “apakah polarisasinya pada arah yang telah saya pilih untuk pengukuran”. Dan hasil pengukurannya hanya dalam jawaban “ya” atau “tidak”, dengan kata lain dalam bentuk bit tunggal. Setelah pengukuran tersebut, foton dapat berakhir pada bidang yang diukur (jika jawabannya adalah “ya”) atau pun tegak lurus bidang tersebut (jika jawabannya adalah “tidak”).

Jika Reza hendak mengukur sudut polarisasi dengan lebih akurat, kenapa dia tidak mengulang pengukuran beberapa kali dan mengambil nilai rata-ratanya. Hal tersebut tidak dapat dilakukan karena setiap kali akan mulai mengukur maka akan me-*reset* sudut polarisasi kepada sudut yang diukur atau sudut yang tegak lurus dengannya. Dengan demikian pengukuran berikutnya akan memberikan hasil yang tetap sama.

Atau Reza membuat beberapa duplikat foton dan mengukur tiap duplikat tersebut. Pendekatan ini pun tidak dapat digunakan karena ia hanya dapat membuat salinan foton dengan mengukur sifat-sifatnya dan membuat foton baru berdasarkan sifat-sifat tersebut.

Apa yang akan diukur Reza jika Dewi telah menyiapkan foton dengan sembarang sudut? Atau tepatnya sudut polarisasi foton telah berubah karena adanya interaksi acak sepanjang waktu? Atau foton telah diukur oleh sebuah *evil eavesdropper* (khususnya dinamakan Eve) pada beberapa sudut lainnya dan oleh karena itu (foton) di-*reset* pada sudut tersebut? Pada kasus seperti ini, Reza selalu mendapat jawaban “ya” atau “tidak” pada arah manapun polarisasi pengukuran yang dipilihnya. Dengan lebih dekat pada polarisasi aktualnya maka kemungkinan besar jawabannya adalah “ya”. Tegasnya untuk kemungkinan jawaban “ya” adalah kosinus kuadrat sudut antara sudut pengukuran Reza dengan sudut yang dipersiapkan Dewi. Hasil dari pengukuran-pengukuran yang dilakukan oleh Reza tidak mungkin diperkirakan sebelumnya. Sifat ketidakaturan tersebut merupakan aspek integral dari mekanika kuantum.

Qubits memiliki beberapa sifat lainnya yang menarik, yang tidak disebutkan lebih jauh, bahwa ketika ada dua atau lebih dari mereka dipersiapkan bersama-sama dengan cara yang khusus. Salah satu sifat qubit, yang tidak didiskusikan sekarang, adalah yang dikenal sebagai “entanglement” (belitan). Sifat tersebut memungkinkan bagi dua foton untuk bergerak ke tempat yang berbeda sebelum keadaan yang berkolerasi tunggal pada pengukuran sebuah foton mempengaruhi pengukuran foton selanjutnya.

Perlu dicatat bahwa tidak semua keanehan yang ditunjukkan oleh sistem kuantum dibahas disini. Model bit klasik cukup memadai untuk beberapa sistem kuantum. Sebagai contoh adalah jika sudut polarisasi dibatasi pada posisi horisontal dan vertikal dan tidak ada gangguan *noise*, sudut pengukuran yang sesuai diketahui, dan pengukuran dapat dilakukan tanpa gangguan foton, dengan demikian dalam kasus khusus ini penggandaan mungkin dilakukan. Jika ada sejumlah kecil *noise* yang begitu kecilnya sehingga gangguan pada sudut polarisasi tidak mempengaruhi probabilitas pengukuran, kemudian terdapat pula penyortiran *restoring logic* agar polarisasi selalu di-*reset* ke arah horisontal atau vertikal setelah pengukuran.

1. 5. Keunggulan Qubit

Terdapat beberapa hal yang dapat dilakukan berkaitan dengan kuantum non-klasik. Beberapa hal merupakan sesuatu yang menguntungkan. Ingat kembali saat Dewi mencoba mengirim informasi kepada Reza dengan menggunakan foton terpolarisasi. Dewi dapat menyiapkan foton pada beberapa sudut dan kemudian mengatakannya kepada Reza tentang sudut yang dipilih pada awal fase "*setup*". Lalu kita andaikan bahwa ada seorang penyabotase bernama Sam mengganggu komunikasi tersebut dengan mengolah foton pada beberapa titik di jalur antara Dewi dan Reza. Sam membangun sebuah mesin yang akan memantulkan polarisasi pada sudut yang ia pilih. Dengan demikian jika ia memilih sudut 45° maka tiap foton horisontal akan menjadi foton vertikal dan sebaliknya. Dengan mengetahui bahwa Dewi mengkode bit sebagai foton horisontal maupun vertikal, Sam menempatkan sudut 45° dan kemudian menjalankan mesinnya terhadap setengah foton dalam pengiriman pesan yang dipilihnya secara acak.

Oleh karena Dewi mengkode pesan Reza secara horisontal dan vertikal yang kemudian setengah dari bitnya diinversi oleh Sam, dan tidak ada informasi berguna yang mungkin ada, maka untuk setiap foton yang diukur Reza akan serupa dengan yang dikirim oleh Dewi, atau dengan nilai boolean lainnya.

Dewi kemudian mempelajari rencana busuk Sam dan berkeinginan untuk membangun komunikasi kembali dengan Reza. Untuk itu apa yang dapat dilakukannya?

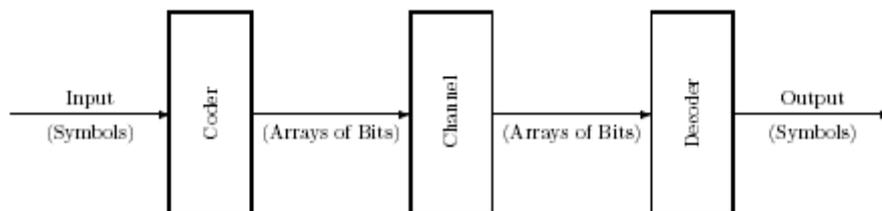
Dewi mengatakan kepada Reza (dengan menggunakan jalur yang tidak mungkin didengar oleh Sam) untuk mengukur foton pada sudut 45° dan 135° . Kemudian jika mesin Sam memantulkan sudut polarisasi sebesar 45° maka hal tersebut tidak akan mempengaruhi dua keadaan yang telah dipilih Dewi. Tentu saja jika Sam mengetahui yang dilakukan oleh Dewi, ia akan memutar mesinnya kembali ke arah vertikal. Atau pun melakukan pengukuran lain dan pengukuran tandingan.

Skenario tersebut bergantung pada sifat alami kuantum foton dan kenyataan bahwa foton tunggal tidak dapat diukur oleh Sam kecuali sepanjang sudut polarisasi tertentu. Dengan demikian cara yang digunakan Dewi untuk menghalangi Sam tidak mungkin dilakukan bila dengan bit klasik.

Di bab sebelumnya kita menguji satuan dasar informasi yaitu bit dan berbagai penyajian abstraknya yaitu bit matematika, bit klasik, dan bit kuantum.

Bit tunggal bermanfaat jika dua jawaban tepat bagi satu pertanyaan adalah mungkin. Sebagai contoh meliputi hasil lemparan koin (kepala atau ekor), jenis kelamin seseorang (pria atau wanita), putusan dewan juri (bersalah atau tidak bersalah), dan kebenaran dari suatu pernyataan (palsu atau benar). Kebanyakan situasi dalam hidup merupakan suatu hal yang rumit. Bab ini menitikberatkan pada bagaimana cara agar objek kompleks dapat direpresentasikan tidak oleh bit tunggal tetapi oleh susunan bit (*array of bit*).

Merupakan suatu hal yang menyenangkan untuk memusatkan perhatian pada model sistem yang sangat sederhana, di mana masukan merupakan salah satu dari susunan objek atau "simbol" yang telah ditentukan sebelumnya. Identitas dari simbol khusus yang terpilih disandikan dengan suatu susunan bit. Bit-bit tersebut ditransmisikan melalui ruang atau waktu, dan kemudian kode-kode tersebut diuraikan di lain waktu atau di suatu tempat berbeda untuk menentukan simbol asli yang terpilih. Di bab kemudian kita akan berdasar pada model ini dalam kaitannya dengan isu ketahanan dan efisiensi.



Gambar 2.1 Sistem Komunikasi Umum

Di bab ini kita akan melihat beberapa aspek perancangan kode, dan mendiskusikan beberapa contoh dalam aspek ini baik yang berfungsi dengan baik ataupun tidak terlalu baik. Catatan terpisah akan menguraikan kode yang menggambarkan poin-poin yang penting, yaitu:

- Huruf : BCD, EBCDIC, ASCII, *Unicode*, Kode Morse
- *Integer* : Biner, *Gray*, komplement 2

- Angka : *Floating-Point*
- Protein : Kode genetik
- Telepon : NANP, Kode internasional
- Host : *Ethernet, IP Address, Nama Daerah*
- Gambar : TIFF, GIF, dan JPEG
- Audio : MP3, AAC
- Video : MPEG, AVI

2.1 Ukuran Ruang Simbol

Pertimbangan yang berbeda merupakan hal yang penting tergantung pada jumlah lambang yang perlu untuk disandikan. Banyaknya simbol dalam suatu kode disebut **ukuran ruang simbol**. Kita akan mempertimbangkan ruang simbol dari ukuran yang berbeda, yaitu :

- 2
- Integral pangkat 2
- Terbatas
- Tanpa batas, Dapat dihitung
- Tanpa batas, Tidak dapat dihitung

Jika banyaknya lambang adalah 2, maka pemilihan dapat disandikan dalam sebuah bit tunggal. Jika banyaknya lambang yang mungkin adalah 4, 8, 16, 32, 64, atau integral pangkat 2 lainnya, maka pemilihan dapat disandikan dalam sejumlah bit yang sama besar dengan logaritma basis 2, dari ukuran ruang simbol. Dengan begitu, 2 bit dapat menunjuk gambar intan, hati, keriting atau pun sekop dalam suatu permainan kartu remi. Sedangkan 5 bit dapat menyandikan pemilihan seorang siswa di dalam suatu kelas yang terdiri dari 32 siswa.

Jika banyaknya lambang terbatas tetapi bukan suatu integral pangkat 2, maka banyaknya bit yang berfungsi untuk integral pangkat 2 yang lebih tinggi berikutnya dapat digunakan untuk menyandikan pemilihan, tetapi akan ada beberapa pola bit yang tak terpakai. Contoh lainnya yang meliputi yang 10 digit adalah yang enam sisi dadu kubus, 13 lembar kartu remi, dan 26 huruf dalam alfabet Inggris. Pada setiap kasus, ada kapasitas cadangan (6 pola tak terpakai dalam representasi digit 4 bit, 2 pola tak terpakai dalam representasi dadu 3 bit, dan lain-lain). Hal yang dapat dilakukan dengan kapasitas cadangan ini merupakan persoalan disain yang penting yang akan dibicarakan pada bagian berikutnya.

Jika jumlah simbol tak terbatas tetapi dapat dihitung (dapat diletakkan dalam hubungan satu lawan satu dengan integer) maka rangkaian bit dari ukuran panjang yang diberikan hanya dapat menunjukkan suatu jumlah materi yang terbatas dari susunan yang tak terbatas. Dengan demikian suatu kode 4-bit untuk integer positif dapat menunjuk integer dari 0 sampai 15, tetapi tidak akan dapat menangani integer di luar rentang tersebut. Sebagai hasil beberapa

perhitungan, jika diperlukan untuk menghadirkan angka-angka lebih besar, maka kondisi “luapan” tersebut harus dapat ditangani dengan suatu cara.

Jika jumlah lambang tak terbatas dan tak terhitung (seperti nilai suatu kuantitas fisik seperti voltase ataupun tekanan akustik), maka beberapa teknik harus digunakan untuk menggantikan nilai-nilai yang mungkin oleh suatu jumlah yang terbatas dari nilai-nilai terpilih yang kira-kira sama. Sebagai contoh adalah jika angka-angka antara 0 dan 1 adalah lambang dan jika 2 bit tersedia untuk representasi yang disandikan, maka satu pendekatan boleh jadi mendekati semua angka-angka antara 0 dan 0.25 sebesar 0.125, lalu semua angka-angka antara 0.25 dan 0.5 oleh 0.375, dan seterusnya. Apakah perkiraan seperti itu adalah cukup memadai tergantung pada penggunaan cara peletakkan data yang disandikan. Perkiraan bersifat tidak reversibel, yang dalam hal ini tidak ada dekoder yang akan menemukan kembali simbol asli yang diberikan; hanya kode untuk nilai perkiraan. Akan tetapi, jika jumlah bit yang tersedia cukup besar, maka untuk berbagai kegunaan dekoder bisa menyediakan suatu nomor yang cukup dekat. Representasi *Floating-Point* dari bilangan riil di komputer didasarkan pada filosofi tersebut.

2.2 Penggunaan Kapasitas Cadangan

Dalam banyak situasi ada beberapa pola kode yang tak terpakai, oleh karena jumlah simbol bukan suatu integral pangkat 2. Ada beberapa strategi yang mungkin untuk diterapkan berkaitan dengan hal tersebut, yaitu :

- Mengabaikan
- Pemetaan terhadap nilai-nilai lainnya
- Menyediakan untuk ekspansi di waktu mendatang
- *Control code*
- Singkatan umum

Pendekatan-pendekatan tersebut akan digambarkan pada sub-bab berikut disertai dengan contoh-contoh dari kode-kode yang umum.

II.2.1. Binary Coded Decimal (BCD)

Pertama, pola bit ekstra boleh jadi diabaikan begitu saja. Dekoder mungkin tidak mengembalikan apapun atau mungkin memberi sinyal suatu *error*. Kedua, pola ekstra mungkin dipetakan ke dalam nilai-nilai legal. Sebagai contoh adalah pola tak terpakai di BCD mungkin semuanya dikonversi ke 9, berdasarkan teori bahwa mereka merepresentasikan 10, 11, 12, 13, 14, dan 15, dan digit yang terdekat adalah 9. Atau mereka dapat didekodekan sebagai 2, 3, 4, 5, 6, atau 7, dengan men-*setting* bit awal ke 0, berdasarkan teori bit yang pertama bisa jadi salah.

Tabel 2.1 Sistem Desimal Coded Biner

Digit	Code
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

II.2.2. Kode Genetik

Contoh lain pemetaan pola tak terpakai ke dalam nilai-nilai legal diberikan oleh kode genetik, yang diuraikan secara mendalam pada sub-bab 2.7. Protein terdiri dari rangkaian panjang asam amino, yang terdiri dari 20 jenis asam amino yang berbeda yang masing-masingnya mengandung atom antara 10 dan 27 atom. Organisma hidup memiliki berjuta-juta protein yang berbeda, dan dipercaya bahwa semua aktivitas sel melibatkan protein. Protein harus dibuat sebagai bagian dari proses hidup, namun sukar untuk membayangkan berjuta-juta zat kimia dengan kegunaan khusus memproduksi unit-unit yaitu satu untuk tiap jenis protein. Malahan, suatu mekanisme dengan kegunaan umum memasang protein dengan dipandu oleh suatu deskripsi (bayangkan sebagai cetak birunya) yang terkandung dalam molekul DNA (*deoxyribonucleic acid*) dan RNA (*ribonucleic acid*). Baik DNA maupun RNA adalah rantai linear dari nucleotida-nukleotida kecil; sebuah molekul DNA mungkin terdiri lebih dari seratus juta nukleotida. Pada DNA terdapat empat jenis nukleotida, masing-masing terdiri dari beberapa struktur umum dan salah satu dari empat basa yang berbeda, yang dinamai *Adenine*, *Cytosine*, *Guanine*, dan *Thymine*. Pada struktur RNA pun serupa demikian kecuali bahwa *Thymine* digantikan oleh *Uracil*.

Kode genetika merupakan deskripsi tentang bagaimana suatu urutan nukleotida dapat menetapkan suatu asam amino. Sehubungan dengan hal tersebut, keseluruhan protein dapat ditetapkan oleh suatu urutan linear nukleotida. Perlu dicatat bahwa deskripsi kode suatu protein tidaklah dengan sendirinya lebih kecil atau pun lebih sederhana daripada protein itu sendiri; pada kenyataannya, jumlah atom yang diperlukan untuk menetapkan suatu protein adalah lebih besar daripada jumlah dalam protein itu sendiri. Bagaimanapun dalam representasi yang telah distandarisasi, memperbolehkan piranti perakitan yang sama untuk memfabrikasi berbagai protein berbeda pada waktu yang berbeda.

Dengan adanya empat nukleotida yang berbeda, maka salah satunya saja sudah dapat menetapkan paling banyak empat jenis asam amino yang berbeda. Dalam hal ini urutan dari dua nukleotida dapat menentukan 16 jenis asam amino yang berbeda. Tetapi ini belum cukup karena terdapat 20 asam amino yang berbeda yang digunakan dalam protein, maka untuk itu diperlukan suatu urutan tiga nukleotida. Urutan seperti itu disebut dengan kodon. Terdapat 64

kodon berbeda yang lebih dari cukup untuk dapat menetapkan 20 asam amino. Kapasitas cadangan digunakan untuk menyediakan lebih dari satu kombinasi untuk kebanyakan asam amino, dengan demikian menyediakan suatu derajat kekuatan. Sebagai contoh adalah asam amino Alanine yang memiliki 4 kode yang semuanya dimulai dengan GC; dengan begitu nukleotida ketiga dapat diabaikan, maka mutasi yang mengubahnya tidak akan mengganggu fungsi biologi manapun. Sesungguhnya, delapan dari 20 asam amino juga memiliki sifat yang sama seperti itu yaitu bahwa nukleotida ketiga merupakan suatu yang "jangan dipedulikan." (Hal tersebut dapat terjadi karena nukleotida yang ketiga lebih mungkin diubah selama proses transkripsi dibandingkan dengan dua nukleotida lainnya, menghasilkan suatu efek yang disebut "wobble" (goyangan)).

Suatu pengujian terhadap kode genetika mengungkapkan bahwa tiga kodon (UAA, UAG, dan UGA) tidak menetapkan asam amino manapun. Ketiga kodon tersebut menandakan ujung dari protein. Memang suatu "kode perhentian" (*stop code*) diperlukan karena protein yang berbeda memiliki panjang yang berbeda pula. Kodon AUG menetapkan asam amino Methionine dan juga penanda dari awal suatu protein. Dalam hal ini semua rantai protein diawali dengan Methionine. Hal tersebut juga berlaku dalam kode buatan manusia bahwa terdapat beberapa urutan bit sebagai penanda data dan beberapa lainnya sebagai penanda informasi kendali.

II.2.3. Kode Area Telepon

Hal ketiga yang dapat menggunakan kapasitas cadangan adalah memanfaatkannya ekspansi masa depan. Ketika AT&T mendirikan Kode Area telepon ditahun 1947 (yang tersedia untuk penggunaan publik di tahun 1951), kode berisi tiga digit dengan tiga pembatasan, yaitu :

- Digit yang pertama tidak dapat berupa 0 atau 1, untuk menghindari pertentangan dengan sambungan 0 ke operator, dan pada awalnya 1 menjadi efek yang tidak diharapkan dari putaran angka telepon yang sulit, saat ini sebagai suatu isyarat panggilan berbea (*toll call*).
- Digit tengah dapat berupa 0 atau 1 (0 untuk negara dan provinsi dengan hanya satu Kode Area, dan 1 untuk negara dan provinsi dengan lebih dari satu kode area)
- Dua digit terakhir dapat tidak sama (angka-angka dengan format *abb* dapat lebih mudah diingat dan oleh karena itu lebih berharga) dengan begitu panggilan x11 seperti 911 (keadaan darurat), 411 (bantuan direktori), dan 611 (layanan perbaikan) untuk jasa lokal telah dilindungi. Hal ini juga memungkinkan adopsi dari 500 (*follow-me*), 600 (*Canadian wireless*), 700 (*interconnect services*), 800 (panggilan bebas biaya), dan 900 (*added-value information services*).

Sebagai hasilnya, hanya terdapat 144 kode area yang mungkin. Pada awalnya, 86 kode area telah digunakan dan ditugaskan sedemikian sehingga akan lebih cepat dalam pemutaran nomor telepon pada ke daerah dengan lalu lintas sambungan telepon sangat besar (sebagai

contoh adalah 212 untuk wilayah Manhattan). Sisanya, yaitu 58 kode area telah disediakan untuk tugas di kemudian hari. Seperti yang terjadi bahwa kapasitas cadangan tersebut tidak cukup, dan pembatasan pada digit pertengahan yang hanya memberlakukan 0 atau 1 harus diperlonggar. Pada 15 Januari 1995, kode area pertama dengan digit tengah selain 0 atau 1 telah diberlakukan dalam layanan di Alabama. Pada saat ini pembatasan pada kode area adalah bahwa digit yang pertama tidak dapat berupa 0 atau 1, digit tengah tidak dapat berupa 9, dan dua digit terakhir tidak dapat sama.

Mulai tahun 2000, 108 kode area baru telah dimulai. Permintaan yang besar ini merupakan hasil dari perluasan penggunaan jaringan telepon untuk jasa lain yaitu seperti *fax* dan telepon selular, adanya tekanan politis yurisdiksi seperti di Kepulauan Karibia yang menginginkan kode area mereka sendiri, dan juga karena munculnya banyak perusahaan telepon baru yang menawarkan pelayanan dan oleh karena itu mereka memerlukan sedikitnya satu pertukaran menyeluruh di tiap-tiap wilayah penagihan tarip. Dapat diharapkan bahwa *North American Numbering Plan (NANP)* akan kehabisan kode area dalam 20 tahun, dan untuk itu terdapat berbagai proposal untuk mengatasi hal tersebut.

II.2.4. IP Address

Contoh lain dari kebutuhan akan kapasitas cadangan untuk penggunaan masa depan adalah *IP (Internet Protokol) Adresses (alamat IP)*, yang akan diuraikan secara detil pada Sub Bab 2.8. *IP Adresses (versi 4)* tersebut tersedia dalam format *x.x.x.x* dengan *x* adalah suatu angka antara 0 sampai dengan 255. Dengan begitu masing-masing alamat anترنت dapat disandikan dalam total 32 bit. *IP address* ditetapkan oleh *Internet Assigned Numbers Authority (IANA)* dengan situsnya adalah <http://www.iana.org>.

Ledakan minat akan internet menimbulkan adanya permintaan yang besar akan *IP address*. Dalam hal ini organisasi yang mengambil bagian dalam pengembangan Internet, yang telah menetapkan blok angka-angka dalam jumlah besar mulai merasakan bahwa seolah-olah mereka sedang menimbun suatu sumber daya berharga. Di antara organisasi-organisasi tersebut adalah AT&T, BBN, IBM, Xerox, HP, DEC, Apple, MIT, Ford, Stanford, BNR, Prudential, duPont, Merck, *U.S. Postal Service*, dan beberapa agen U.S. DoD (lihat Sub Bab 2.8). Industri tenaga listrik Amerika Serikat (AS), yaitu EPRI (Electric Power Research Institute), meminta banyak alamat internet untuk tiap pelanggan rumah tangga atau kantor, pada pengambilalihan penggunaan dengan peralatan membaca meteran dari jauh. *Internet Engineering Task Force (IETF)*, <http://www.ietf.org/>, menyadari bahwa alamat internet dibutuhkan dalam skala yang lebih baik dan memasuki berbagai bidang kehidupan, lebih daripada yang diimpikan. Sebagai contoh adalah akan ada suatu kebutuhan alamat untuk peralatan seperti lemari es, oven, telepon, dan tungku perapian yang telah *internet-enable*. Selain itu akan ada beberapa kebutuhan dalam tiap otomobil dan truk yang mungkin akan memiliki satu mikroprosesor dan sensor untuk tiap kendaraan. Sebagai hasil dari kondisi ini adalah adanya pengembangan versi

6, IPv6, dengan tiap alamat masih dalam format x.x.x.x, tetapi masing-masing x kini adalah angka 32-bit antara 0 sampai dengan 4,294,967,295. Dengan begitu alamat internet yang baru akan memerlukan 128 bit. Alamat yang telah ada tidak berubah, tetapi semua peralatan jaringan harus diubah untuk mengakomodasi ruang alamat yang lebih besar. Sejak Januari 2001, konversi ke IPv6 dimulai. Alokasi yang baru meliputi blok besar yang dicadangkan untuk ekspansi masa depan, dan dikatakan secara main-main bahwa ada penyimpanan blok alamat untuk penggunaan oleh planet lain. Ukuran ruang alamat adalah cukup besar untuk mengakomodasi suatu perangkat keras pengidentifikasi yang unik untuk masing-masing komputer pribadi, dan beberapa advokat pribadi menunjuk bahwa IPv6 tidak memungkinkan adanya *Web Surfing* tanpa nama.

II.2.5. ASCII

Penggunaan keempat dari kapasitas cadangan dalam kode adalah untuk *denoting formatting* atau operasi kendali. Banyak kode menyertakan pola-pola kode yang bukan merupakan data tetapi adalah kode kendali. Sebagai contoh adalah kode genetik yang memiliki tiga pola dari 64 pola sebagai *stop code* untuk mengakhiri produksi protein.

Kode yang paling umum digunakan untuk karakter teks adalah ASCII (American Standard Code for Information Interchange, yang diuraikan secara detil pada Sub Bab 2.5) yang mencadangkan 33 kode dari 128 kode dengan tegas untuk kendali, dan hanya 95 untuk karakter. Sembilan puluh lima kode tersebut meliputi 26 huruf besar dan 26 huruf kecil alfabet Inggris, 10 digit, spasi, dan 32 tanda baca.

2.3 Isu Strategis adalah Perancangan Kode

Banyak kode dirancang oleh manusia. Terkadang kode sangat kuat, sederhana, mudah dikerjakan, dan penggunaannya meluas. Tetapi terkadang juga bersifat rapuh, rahasia, kompleks, dan tak dapat diuraikan meski dengan generalisasi yang paling sederhana. Seringkali kode yang praktis dan sederhana dikembangkan untuk merepresentasikan sejumlah kecil materi, dan keberhasilannya menarik perhatian dan orang mulai menggunakannya di luar konteks aslinya, yaitu untuk merepresentasikan suatu kelas objek yang lebih besar yang semula bukan kegunaan yang diimpikan. Jika kode digunakan oleh organisasi berbeda dan dibangun dalam sistem perangkat keras atau perangkat lunak yang saling bekerja sama, maka kode-kode tersebut mencapai suatu derajat ketetapan dan seringkali kemudian distandarisasi.

Kode yang telah digeneralisasi dengan baik seringkali membawa bias yang tidak diharapkan dari konteks asli mereka. Kadang-Kadang hasilnya cukup menarik, tetapi dalam kasus lain bias seperti itu membuat kode sulit digunakan.

Salah satu contoh bias yang tak berbahaya adalah fakta bahwa ASCII memiliki dua karakter yang mula-mula dimaksudkan untuk diabaikan. ASCII diawali sebagai pola lubang 7-bit

pada pita kertas yang digunakan untuk merekam informasi dari mesin teleks. Aslinya, pita tidak memiliki lubang (kecuali satu rangkaian lubang kecil untuk meluruskan dan memasukkan pita), dan dimasukkan melalui sebuah lubang. Pita bisa dimasukkan baik dengan transmisi yang diterima atau oleh adanya ketikan manusia pada *keyboard*. Kepala pita (bagian pertama dari pita) yang tidak dimasukkan, oleh karena itu merepresentasi satu rangkaian karakter 0000000 yang panjangnya tak dapat ditentukan (0 berarti tidak ada lubang). Tentu saja ketika pita dibaca maka kepala pita harus diabaikan, yang dengan adanya konvensi karakter 0000000 disebut dengan NUL, dan hal tersebut telah diabaikan. Kemudian, jika ASCII digunakan dalam komputer, sistem berbeda memerlukan NUL dengan cara yang berbeda. Unix memberlakukan NUL sebagai ujung suatu kata dalam beberapa keadaan, dan penggunaan ini bertentangan dengan aplikasi di mana kode ASCII diberi suatu penafsiran kuantitatif. Kode ASCII lain yang pada mulanya dimaksudkan untuk diabaikan adalah DEL, 1111111. Konvensi ini sangat menolong bagi para juru ketik untuk dapat "menghapus" suatu kesalahan dengan membalikkan pita dan melubangi tiap lubang. Dalam konteks modern, DEL sering diberlakukan sebagai suatu *backspace* destruktif. Tetapi beberapa editor teks di masa lalu telah menggunakan DEL sebagai penghapus karakter *forward*, dan kadang-kadang dapat dengan mudah diabaikan.

Bias yang lebih serius lainnya yang dibawa oleh ASCII adalah dalam penggunaan dua karakter, yaitu CR (*carriage return*) dan LF (*line feed*), untuk berpindah ke baris cetakan yang baru. Mekanisme fisik dalam mesin teleks telah memisahkan perangkat keras untuk menggerakkan kertas ke atas (pada suatu *continuous roll*), dan memposisikan kembali elemen cetakan ke margin kiri. Para insinyur yang telah merancang kode yang merupakan pengembangan ASCII merasa pasti bahwa mereka sedang melakukan suatu hal yang baik dengan memungkinkan operasi ini untuk diadakan secara terpisah. Mereka tidak bisa membayangkan kesulitan yang mereka berikan kepada generasi insinyur mendatang dengan diadaptasinya ASCII pada situasi yang menggunakan perangkat keras yang berbeda dan tidak diperlukannya perpindahan titik cetakan seperti diadakan oleh CR dan LF secara terpisah. Sistem komputasi yang berbeda telah memilih penafsiran yang berbeda (Unix menggunakan LF untuk baris baru dan mengabaikan CR, Macintosh menggunakan CR dan mengabaikan LF, dan DOS/WINDOWS memerlukan keduanya) dan ketidakcocokan ini merupakan suatu sumber kesalahan dan frustrasi serius yang berkelanjutan.

Kode area telepon di Amerika Utara menawarkan contoh lain tentang bagaimana asumsi awal menyebabkan kesulitan di waktu kemudian, dan hal ini cukup mengejutkan mengingat standar mutu telepon di Amerika Utara yang tinggi, dan fakta bahwa industri dengan ketat dikoordinir. Hal tersebut memerlukan bahwa digit tengah untuk menjadi digit apapun, tidak hanya 0 atau 1. Banyak peralatan telepon yang yang digunakan telah dirancang di sekitar asumsi bahwa pertengahan digit bisa saja jadilah 0 atau 1, terutama PBX (Private Branch eXchanges) yang dirancang oleh penyuplai mandiri. Tidak semua peralatan di*upgrade* pada waktunya, sehingga beberapa pelanggan tidak dapat melakukan panggilan keluar ke kode area

yang baru untuk beberapa waktu. *Upgrade* memerlukan revisi perangkat lunak (software) ataupun dalam beberapa kasus memerlukan perangkat keras yang baru.

2.4 Kode dengan Panjang Tertentu dan Panjang Variabel

Sebuah keputusan yang harus dibuat pada awal perancangan suatu kode adalah apakah akan merepresentasikan semua lambang dengan kode dalam jumlah bit yang sama (panjang tetap/*fixed-length*) atau beberapa lambang menggunakan kode yang lebih pendek daripada beberapa yang lain (panjang variabel/*variable-length*). Dalam hal ini keduanya sama-sama memberi keuntungan.

Fixed-Length code pada umumnya lebih mudah digunakan karena baik pengkode dan dekoder dapat mengikuti jumlah bit yang terlibat, dan hal tersebut hanyalah masalah pengaturan atau pembacaan nilai. Dengan *variable-length code*, dekoder harus menggunakan nilai bit untuk menentukan kapan simbol diketahui dan oleh karena itu waktu awal interpretasi bit yang berikutnya adalah sebagai awal simbol kode yang berikutnya.

Fixed-Length code dapat didukung oleh interpretasi dan transmisi paralel, di mana bit dikomunikasikan dari koder kepada dekoder secara simultan misalnya dengan menggunakan kabel multipel sebagai pembawa voltase. Pendekatan ini menjadi kontras dengan transport informasi kode secara seri, yang menggunakan kawat tunggal untuk mengirimkan aliran bit dan dekoder harus memutuskan bagaimana bit digunakan untuk memecahkan kode simbol yang berbeda. Jika dekoder dikacaukan, atau berada dalam aliran bit setelah dimulai, maka mungkin tidak bisa diketahui jika satu simbol berhenti dan simbol lainnya mulai. Hal tersebut dikenal sebagai sebuah "*framing error*". Untuk menghilangkan *framing error* maka bit stop sering dikirim di antara simbol. Secara khusus ASCII mengirimkan *serial lines* yang memiliki 1 atau 2 bit stop yang secara normal memberi nilai 0. Dengan begitu jika suatu dekoder keluar dari prosedur, maka pada akhirnya akan ditemukan 1 pada apa yang seharusnya diasumsikan sebagai bit stop, dan dapat dicoba untuk mensinkronkan kembali. Walaupun framing error akan tetap berlaku untuk periode lama, dalam prakteknya penggunaan bit stop memberikan hasil yang baik.

II.4.1. Kode Morse

Variable-Length code sangat bermanfaat jika beberapa simbol lebih sering digunakan daripada simbol yang lainnya. Gagasan untuk menetapkan urutan singkat terhadap simbol yang umum dan urutan panjang pada simbol yang tidak umum/jarang digunakan, maka dengan demikian akan tercapai pesan kode yang secara rata-rata lebih pendek. Sebagai contoh adalah bahwa tidak semua huruf dalam alfabet Inggris sering digunakan, yaitu bahwa huruf X, J, Q, dan Z merupakan huruf yang jarang digunakan. Di bawah ini dicantumkan tabel frekuensi penggunaan huruf dalam tulisan berbahasa Inggris (angka yang tercantum merupakan frekuensi kemunculan huruf tertentu secara rata-rata dalam tiap 1000 huruf).

Tabel 2.2 Frekuensi Penggunaan Huruf dalam Tulisan Bahasa Inggris

132	E	61	S	24	U
104	T	53	H	20	G, P, Y
82	A	38	D	19	W
80	O	34	L	14	B
71	N	29	F	9	V
68	R	27	C	4	K
63	I	25	M	1	X, J, Q, Z

Variable-length code yang diperuntukkan dalam teks Bahasa Inggris dapat mencapai efisiensi lebih besar dengan memilih suatu pola bit pendek untuk huruf-huruf seperti E, T, A, O, dan N, dan membiarkan huruf-huruf yang lebih sedikit digunakan dengan pola bit yang lebih panjang. Tentu saja pola spesifik yang digunakan harus dipilih yang "bebas prefiks/imbuhan" dengan pengertian bahwa suatu kode yang pendek tidak bisa serupa seperti halnya bagian pertama dari kode lebih panjang lainnya, jika tidak dekoder tidak akan dapat membedakan mereka. Dalam praktek belum ada manfaat yang diperoleh, dalam kasus teks, cukup untuk memberikan kompleksitas yang lebih besar, dan teks secara normal dikode dalam *suatu fixed-length code* seperti ASCII. Bagaimanapun, satu pengecualian adalah dalam kode morse.

Kode morse dirancang untuk digunakan pada telegraf, dan kemudian digunakan pula dalam komunikasi radio sebelum radio AM dapat membawa suara. Sampai tahun 1999, hal tersebut masih diperlukan dalam komunikasi pada kapal samudra, walaupun itu jarang digunakan (tampaknya teori yang berlaku adalah bahwa beberapa perahu yang lebih tua belum mengkonversi cara komunikasinya ke cara yang lebih modern). Kemampuan mengirimkan dan menerima kode morse masih menjadi kebutuhan bagi warga negara Amerika Serikat yang ingin mendapatkan lisensi radio amatir.

Kode morse dinamai demikian setelah Samuel F. B. Morse, penemu telegraf, dihargai dengan format modernnya. Kode morse terdiri dari urutan panjang dan pendek dari suatu denyut atau nada (titik dan garis) yang dipisahkan oleh kesunyian beberapa saat. Seseorang dapat mengirimkan kode morse dengan cara membuat dan memutuskan sambungan listrik, dan orang yang berada di seberang jalur komunikasi tersebut dapat mendengarkan kode morse tersebut sebagai urutan titik dan garis yang kemudian mengkonversikannya ke dalam bentuk huruf, spasi dan tanda baca.

Sejak kode morse dirancang untuk dapat didengar, tidak untuk dilihat, maka tabel huruf dalam alfabet Inggris seperti yang tertera pada Tabel 2.3 hanya secara garis besar bermanfaat. Anda tidak dapat mempelajari kode morse dengan memandangi titik dan garis yang tertulis tetapi anda harus mendengar sendiri kode morse tersebut.

Tabel 2.3 Kode Kode Morse

A	..	K	---	U	---	0	-----	Period	-----
B	L	V	1	-----	Hyphen	-----
C	M	---	W	---	2	-----	Comma	-----
D	---	N	..	X	3	-----	Colon	-----
E	.	O	----	Y	----	4	-----	Question Mark	-----
F	P	Z	----	5	-----	Apostrophe	-----
G	---	Q	----			6	-----	Parenthesis	-----
H	R	---			7	-----	Quotation mark	-----
I	..	S	...			8	-----	Fraction bar	-----
J	----	T	-			9	-----	Delete prior word	-----
								End of Transmission	-----

Jika jangka waktu titik diambil sebagai satu unit maka untuk satu garis adalah tiga unit. Spasi antara titik dan garis di dalam satu karakter adalah satu unit, dan yang di antara karakter adalah tiga unit, dan spasi antar kata adalah tujuh unit. Spasi tidak dianggap sebagai satu karakter, seperti halnya dalam ASCII.

Perbandingan dua tabel di bawah ini menunjukkan bahwa kode morse cukup baik dalam menerapkan urutan pendek pada huruf-huruf umum. Pernah dilaporkan bahwa Morse tidak menciptakan kode morse tersebut dengan membaca buku dan surat kabar ataupun menghitung huruf, tetapi dengan mengunjungi suatu percetakan. Percetakan ketika itu menggunakan mesin *movable type*, di mana huruf yang berbeda disusun oleh pencetak ke dalam baris. Masing-masing huruf tersedia dalam berbagai salinan untuk masing-masing jenis huruf dan ukuran dalam wujud potongan timah. Morse pun dengan mudah menghitung potongan yang tersedia untuk masing-masing huruf yang terdapat dalam alfabet, dengan mengasumsikan bahwa percetakan tahu akan bisnisnya dan abjad, mengira bahwa pencetak itu mengenal bisnis mereka dan menyediakan cukup stok untuk tiap huruf agar dapat memenuhi permintaan cetakan. Dalam kasus tipe kaku, huruf-huruf diatur dalam dua baris, yaitu huruf besar pada baris atas dan huruf kecil pada baris di bawahnya. Mesin cetak mengacu pada hal tersebut dengan mencetak kode pada baris bagian atas sebagai huruf besar.

2.5 Detail: ASCII

ASCII, yang merupakan *The American Code for Information Interchange*, diperkenalkan oleh *American National Standards Institute* (ANSI) pada tahun 1963. ASCII merupakan kode karakter yang paling umum digunakan. Dalam hal 8-bit, bit awal diset pada 0; dengan begitu ASCII dapat dikatakan sebagai "*bottom half*" dari kode karakter 8-bit. Kode-kodenya adalah sebagai berikut:

Tabel 2.4 Paket Karakter ASCII

Control Characters				Digits			Uppercase			Lowercase		
HEX	DEC	CHR	Ctrl	HEX	DEC	CHR	HEX	DEC	CHR	HEX	DEC	CHR
00	0	NUL	^@	20	32	SP	40	64	␣	60	96	·
01	1	SOH	^A	21	33	!	41	65	A	61	97	a
02	2	STX	^B	22	34	"	42	66	B	62	98	b
03	3	ETX	^C	23	35	#	43	67	C	63	99	c
04	4	EOT	^D	24	36	\$	44	68	D	64	100	d
05	5	ENQ	^E	25	37	%	45	69	E	65	101	e
06	6	ACK	^F	26	38	&	46	70	F	66	102	f
07	7	BEL	^G	27	39	'	47	71	G	67	103	g
08	8	BS	^H	28	40	(48	72	H	68	104	h
09	9	HT	^I	29	41)	49	73	I	69	105	i
0A	10	LF	^J	2A	42	*	4A	74	J	6A	106	j
0B	11	VT	^K	2B	43	+	4B	75	K	6B	107	k
0C	12	FF	^L	2C	44	,	4C	76	L	6C	108	l
0D	13	CR	^M	2D	45	-	4D	77	M	6D	109	m
0E	14	SO	^N	2E	46	.	4E	78	N	6E	110	n
0F	15	SI	^O	2F	47	/	4F	79	O	6F	111	o
10	16	DLE	^P	30	48	0	50	80	P	70	112	p
11	17	DC1	^Q	31	49	1	51	81	Q	71	113	q
12	18	DC2	^R	32	50	2	52	82	R	72	114	r
13	19	DC3	^S	33	51	3	53	83	S	73	115	s
14	20	DC4	^T	34	52	4	54	84	T	74	116	t
15	21	NAK	^U	35	53	5	55	85	U	75	117	u
16	22	SYN	^V	36	54	6	56	86	V	76	118	v
17	23	ETB	^W	37	55	7	57	87	W	77	119	w
18	24	CAN	^X	38	56	8	58	88	X	78	120	x
19	25	EM	^Y	39	57	9	59	89	Y	79	121	y
1A	26	SUB	^Z	3A	58	:	5A	90	Z	7A	122	z
1B	27	ESC	^[3B	59	:	5B	91	{	7B	123	{
1C	28	FS	^\	3C	60	i	5C	92	\	7C	124	—
1D	29	GS]`	3D	61	=	5D	93		7D	125	}
1E	30	RS	^^	3E	62	>	5E	94	~	7E	126	~
1F	31	US	^_	3F	63	?	5F	95	·	7F	127	DEL

Karakter kendali digunakan untuk isyarat akan kondisi-kondisi khusus, seperti yang dicakup dalam Tabel 2.5.

Seratus dua puluh delapan (128) karakter yang diwakili oleh kode antara HEX 80 dan HEX FF (kadang-kadang disebut sebagai "*high ASCII*") digambarkan secara berbeda dalam konteks yang berbeda. Pada banyak sistem operasi, mereka meliputi huruf aksen Eropa Barat dan berbagai tanda baca tambahan. Pada IBM PCS mereka meliputi karakter *line-drawing*. Macs menggunakan suatu pengkodean yang berbeda.

Kebetulan, orang-orang sekarang menghargai kebutuhan akan interoperabilitas *platform* komputer, maka standard yang lebih universal akan memenuhi hal tersebut. Kode yang paling banyak digunakan untuk halaman web adalah ISO-8859-1 (ISO-Latin) yang menggunakan 96 kode antara HEX A0 Dan HEX FF untuk berbagai huruf dan tanda baca aksen dalam bahasa Eropa Barat dan sedikit simbol lainnya. Sedangkan 32 karakter antara HEX 80 dan HEX 9F dicadangkan sebagai karakter kendali di dalam ISO-8859-1.

Tabel 2.5 Karakter Pengendalian ASCII

HEX	DEC	CHR	Ctrl	Meaning
00	0	NUL	^@	NULL blank leader on paper tape; generally ignored
01	1	SOH	^A	Start Of Heading
02	2	STX	^B	Start of TeXt
03	3	ETX	^C	End of TeXt; matches STX
04	4	EOT	^D	End Of Transmission
05	5	ENQ	^E	ENQuiry
06	6	ACK	^F	ACKnowledge; affirmative response to ENQ
07	7	BEL	^G	BELL; audible signal, a bell on early machines
08	8	BS	^H	BackSpace; nondestructive, ignored at left margin
09	9	HT	^I	Horizontal Tab
0A	10	LF	^J	Line Feed; paper up or print head down; new line on Unix
0B	11	VT	^K	Vertical Tab
0C	12	FF	^L	Form Feed; start new page
0D	13	CR	^M	Carriage Return; print head to left margin; new line on Macs
0E	14	SO	^N	Shift Out; start use of alternate character set
0F	15	SI	^O	Shift In; resume use of default character set
10	16	DLE	^P	Data Link Escape; changes meaning of next character
11	17	DC1	^Q	Device Control 1; if flow control used, XON, OK to send
12	18	DC2	^R	Device Control 2
13	19	DC3	^S	Device Control 3; if flow control used, XOFF, stop sending
14	20	DC4	^T	Device Control 4
15	21	NAK	^U	Negative Acknowledge; response to ENQ
16	22	SYN	^V	SYNchronous idle
17	23	ETB	^W	End of Transmission Block
18	24	CAN	^X	CANcel; disregard previous block
19	25	EM	^Y	End of Medium
1A	26	SUB	^Z	SUBstitute
1B	27	ESC	^[ESCape; changes meaning of next character
1C	28	FS	^\	File Separator; coarsest scale
1D	29	GS	^]	Group Separator; coarse scale
1E	30	RS	^^	Record Separator; fine scale
1F	31	US	^_	Unit Separator; finest scale
20	32	SP		SPace; not always considered a control character
7F	127	DEL		DELeTe; originally ignored; often destructive backspace

Hal yang tidak disukai yaitu bahwa: Kebanyakan orang-orang tidak menginginkan 32 karakter kendali tambahan (tentu saja 33 karakter kendali dalam ASCII 7-bit, hanya selusin atau kira-kira sedemikian yang secara teratur digunakan dalam teks). Konsekuensinya adalah tidak ada hentinya gagasan untuk menggunakan HEX 80 sampai dengan HEX 9F. Konvensi yang paling luas digunakan adalah halaman kode Microsoft 1252 (Latin I) yang serupa dengan ISO-8859-1 (ISO-LATIN) kecuali 27 dari 32 kode kendali ditetapkan untuk karakter cetak. Hal yang paling menarik dari karakter ini adalah HEX 80 Euro. Tidak semua *platform* dan sistem operasi mengenal CP-1252, maka dokumen dan khususnya halaman web memerlukan perhatian khusus.

Di luar 8 Bit

Untuk menghadirkan bahasa-bahasa di Asia maka dibutuhkan karakter lebih banyak lagi. Saat ini sedang dikembangkan secara aktif akan standard yang sesuai, dan umumnya dirasakan bahwa total jumlah karakter adalah lebih sedikit dari 65.536. Hal ini menguntungkan sebab banyak karakter berbeda dapat direpresentasikan dalam 16 bit atau 2 *byte*. Dalam rangka mempertahankan jumlah tersebut, maka versi tulisan berbagai dialek Cina harus berbagi simbol untuk yang terlihat mirip. Kandidat utama untuk karakter standar kode 2-byte saat ini adalah Unicode.

Referensi

Terdapat banyak halaman web yang memberikan tabel ASCII, dengan ekstensi semua bahasa di dunia. Di antaranya yang paling bermanfaat adalah:

- Jim Price, dengan Tabel 8-Bit PC dan Windows, dan beberapa sambungan lebih lanjut
<http://www.jimprice.com/jim-asc.htm>
- Mac OS Characters yang mencakup Tabel Mac 8-Bit
<http://developer.apple.com/techpubs/mac/text/text-30.html>
- A Brief History of Character Codes, dengan suatu diskusi ekstensi ke Bahasa Asia
<http://tronweb.super-nova.co.jp/characodehist.html>
- Unicode home page
<http://www.unicode.org/>
- CP-1252 standard, definitif
<http://www.microsoft.com/globaldev/reference/sbcs/1252.htm>
- CP-1252 membandingkan untuk:
 - Unicode
<ftp://ftp.unicode.org/Public/MAPPINGS/VENDORS/MICSFT/WINDOWS/CP1252.TXT>
 - Unicode/HTML
<http://www.hclrss.demon.co.uk/demos/ansi.html>
 - ISO-8859-1/MacOS
<http://www.jwz.org/doc/charsets.html>

2.6 Detail: Kode Integer

Ada banyak cara untuk menghadirkan integer sebagai pola bit. Semua cara tersebut memiliki ketidakmampuan untuk menghadirkan integer jumlah besar yang berubah-ubah dalam jumlah bit yang tetap. Perhitungan yang memberikan hasil yang keluar dari rentang dikatakan “*overflow*”.

Representasi yang paling umum digunakan adalah kode biner untuk integer tak bertanda (misal: alamat memori), komplemen 2 untuk integer bertanda (misal : aritmatika biasa), dan kode *gray biner* untuk instrumen pengukuran jumlah yang berubah.

Tabel berikut ini memberikan lima contoh kode integer 4 bit. **MSB** (*most significant bit*) pada sisi kiri dan **LSB** (*least significant bit*) pada sisi kanan.

Tabel 2.6 Kode Integer 4 Bit

Range	Unsigned Integers		Signed Integers		
	Binary Code [0, 15]	Binary Gray Code [0, 15]	2's Complement [-8, 7]	Sign/Magnitude [-7,7]	1's Complement [-7,7]
-8			1000		
-7			1001	1111	1000
-6			1010	1110	1001
-5			1011	1101	1010
-4			1100	1100	1011
-3			1101	1011	1100
-2			1110	1010	1101
-1			1111	1001	1110
0	0000	0000	0000	0000	0000
1	0001	0001	0001	0001	0001
2	0010	0011	0010	0010	0010
3	0011	0010	0011	0011	0011
4	0100	0110	0100	0100	0100
5	0101	0111	0101	0101	0101
6	0110	0101	0110	0110	0110
7	0111	0100	0111	0111	0111
8	1000	1100			
9	1001	1101			
10	1010	1111			
11	1011	1110			
12	1100	1010			
13	1101	1011			
14	1110	1001			
15	1111	1000			

Kode Biner

Kode biner diperuntukkan bagi integer non-negatif. Untuk kode dengan panjang n, maka pola 2^n merepresentasikan integer 0 sampai dengan $2^n - 1$. LSB adalah 0 untuk integer genap dan 1 untuk integer ganjil.

Kode Gray Binary

Kode gray binary adalah untuk integer nonnegatif. Untuk kode yang panjangnya n, pola 2^n merepresentasikan integer 0 sampai dengan $2^n - 1$. Pola dua bit dari integer yang berdekatan berbeda persis satu bit. Sifat tersebut menjadikan kode berguna sensor di mana integer yang disandikan berubah selama pengukuran berlangsung.

Komplemen 2

Kode ini adalah untuk integer baik positif maupun negatif. Untuk kode dengan panjang n, maka pola 2^n dapat merepresentasikan integer -2^{n-1} sampai dengan $2^{n-1} - 1$. LSB adalah 0 untuk integer genap dan 1 untuk integer ganjil. Jika mereka tumpang-tindih maka kode ini adalah sama seperti kode biner. Kode ini digunakan secara luas.

Sign/Magnitude

Kode ini adalah untuk integer baik positif maupun negatif. Untuk kode dengan panjang n, maka pola 2^n merepresentasikan integer $-(2^{n-1} - 1)$ sampai dengan $2^{n-1} - 1$. MSBnya adalah 0 untuk integer positif dan 1 untuk integer negatif; bit lainnya membawa *magnitude*. Jika mereka tumpang-tindih, maka kode ini adalah sama seperti halnya kode biner. Sedangkan bila secara

konseptual sederhana, maka kode ini tampak aneh dalam prakteknya. Representasi terpisah untuk +0 dan -0 tidaklah bermanfaat secara umum.

Komplemen 1's

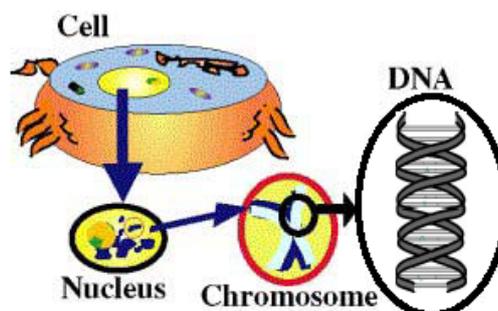
Kode ini adalah untuk integer positif maupun negatif. Untuk kode dengan panjang n maka pola 2^n menghadirkan integer $-(2^{n-1} - 1)$ sampai dengan $2^{n-1} - 1$. MSB adalah 0 untuk integer positif; sedangkan untuk integer negatif dibentuk dengan melengkapi masing-masing bit dengan integer positif yang bersesuaian. Jika mereka tumpang-tindih maka kode ini sama seperti halnya kode biner. Kode ini merupakan kode yang aneh dan jarang digunakan pada saat ini. Representasi terpisah untuk +0 dan -0 tidak bermanfaat secara umum.

2.7 Detail: Kode Genetik

Blok dasar yang membangun badanmu adalah sel. Dua atau lebih kelompok sel membentuk jaringan, seperti tulang atau otot; kelompok jaringan-jaringan membentuk organ badan, seperti jantung atau otak; organ membentuk sistem organ, seperti sistem peredaran darah atau sistem syaraf; dan sistem organ bersama-sama membentuk organisme, yaitu anda. Sel dapat digolongkan menjadi sel *eukaryote* (ada inti sel) maupun sel *prokaryote* (tanpa inti sel). Sel yang menyusun badan manusia dan juga semua hewan, tumbuhan, dan jamur adalah sel eukaryotic. Sedangkan sel prokaryote terdapat pada bakteri dan *cyanobacteria*.

Inti sel (nukleus) membentuk suatu kompartemen terpisah di dalam badan sel; kompartemen ini bertindak sebagai pusat penyimpanan untuk semua informasi hereditas dari sel *eukaryote*. Semua informasi genetik yang membentuk catatan hidup disimpan dalam kromosom yang ditemukan di dalam inti. Pada manusia sehat terdapat 23 pasang kromosom (total terdapat 46 kromosom). Masing-masing kromosom berisi satu molekul DNA yang berbentuk seperti benang. Gen merupakan bagian fungsional yang berada di sepanjang untai DNA, dan merupakan unit fisik dasar yang membawa informasi hereditas dari satu generasi kepada generasi berikutnya.

Di dalam sel *prokaryote*, kromosom bebas mengapung di dalam badan sel karena tidak memiliki inti sel.



Gambar 2.2 Penempatan DNA di dalam Suatu Sel

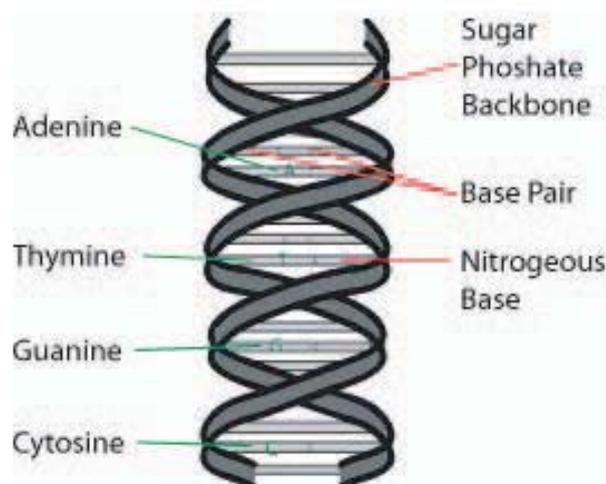
Molekul DNA terdiri atas dua rantai nukleotida yang saling berhubungan untuk membentuk satu untai DNA. Masing-masing nukleotida terdiri atas gula, fosfat, dan salah satu dari empat basa. Basa-basa tersebut adalah *adenine*, *guanine*, *cytosine*, dan *thymine*. Agar lebih mudah dalam penyebutannya, tiap nukleotida mengacu pada basanya, sebagai contoh untuk *deoxyguanosine monophosphate* maka hanya dengan mengatakan *guanine* (atau G) ketika mengacu pada nukleotida individual. Dengan demikian kita bisa tulis CCACCA untuk menandai rantai nukleotida *cytosine-cytosine-adenine-cytosine-cytosine-adenine* yang saling berhubungan.

Rantai nukleotida individu saling berhubungan dengan memasang basa nukleotida mereka ke dalam struktur *double helix* tunggal. Aturan pasangan yang berlaku adalah bahwa *cytosine* selalu berpasangan dengan *guanine* sedangkan *thymine* selalu berpasangan dengan *adenine*. Rantai DNA tersebut direplikasi selama pembelahan sel somatik berlangsung (yaitu semua pembelahan sel kecuali yang diperuntukkan menjadi sel kelamin) dan semua informasi genetik lengkap diterapkan pada sel yang dihasilkan.

Gen merupakan bagian dari kromosom dan kode bagi untai DNA. Bagian fungsional individu yang seperti benang DNA disebut gen. Kode informasi dalam gen mengatur pemeliharaan dan pengembangan sel dan organisma. Informasi tersebut bergerak dalam suatu alur dari *input* ke *output*, yaitu: DNA (gen) => mRNA => ribosom/tRNA => Protein. Hal yang penting adalah bahwa protein merupakan output akhir yang dibentuk dari gen; dengan demikian dapat dikatakan bahwa gen adalah cetak biru dari protein individu. Protein itu sendiri dapat menjadi komponen struktural dari tubuh manusia (seperti serabut otot) ataupun sebagai komponen fungsional (yaitu enzim yang membantu pengaturan beribu-ribu proses biokimia dalam tubuh manusia).

Bagian ini didasarkan pada catatan yang ditulis oleh Tim Wagner

Bagian 2.7 adalah kehormatan Tim Wagner.



Gambar 2.3 Suatu Bagan DNA yang Mempertunjukkan Struktur Seperti Bentuk Sekrup

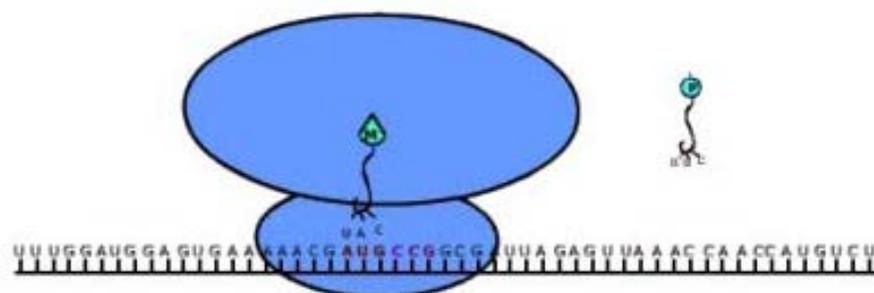
Protein dibangun dari rantai polipeptida yang merupakan rangkaian asam amino (sebuah rantai polipeptida tunggal membuat satu protein, tetapi seringkali bahwa protein fungsional adalah terdiri dari banyak rantai polipeptida).

Pesan genetik dikomunikasikan dari inti sel DNA ke ribosom yang berada di luar inti sel melalui mRNA (messenger RNA). Ribosom adalah komponen sel yang membantu dalam pembentukan protein akhir). Transkripsi adalah proses di mana mRNA dibentuk dari DNA. mRNA adalah salinan dari bagian rantai nukleotida tunggal. Ini merupakan suatu rantai tunggal, persis seperti DNA kecuali perbedaan dalam jenis gula nukleotida dan basa *thymine* digantikan oleh *uracil*. mRNA dibentuk dengan aturan pasangan basa yang sama seperti pada DNA kecuali bahwa T digantikan oleh U (C dengan G, U dengan A).

mRNA kemudian ditranslasi dalam badan sel, dengan bantuan ribosom dan tRNA, menjadi rantai asam-asam amino (yaitu suatu protein). Ribosom menjaga mRNA pada tempatnya dan tRNA (transportRNA) menempatkan asam amino yang sesuai untuk membentuk protein. Proses tersebut digambarkan secara skematik dalam diagram di bawah ini.

mRNA ditranslasi menjadi protein dengan lebih dahulu mengaitkan diri pada ribosom. Pemrakarsa tRNA berikatan dengan ribosom pada suatu titik yang berhubungan dengan kodon awal pada rantai mRNA; pada manusia ini sesuai dengan kodon AUG. Molekul tRNA ini membawa asam amino yang sesuai oleh kodon dan memasangkannya dengan rantai mRNA pada tempat lain di sepanjang rantai nukleotida yang disebut sebagai antikodon. Ikatan yang terbentuk menggunakan aturan pasangan basa yang sama untuk mRNA dan DNA (ada beberapa pengecualian yang diabaikan untuk penyederhanaan). Kemudian molekul tRNA kedua akan berikatan dengan ribosom pada tempat disebelah tRNA pertama yang ditandai oleh kodon berikutnya. Hal itu juga akan membawa asam amino yang bersesuaian yang diminta kodon. Sekali kedua molekul tRNA terkait pada ribosom maka asam amino yang mereka bawa terikat pula. Molekul tRNA awal akan lepas meninggalkan asam aminonya pada rantai asam amino yang sedang tumbuh. Kemudian ribosom akan bergeser ke satu lokasi pada crantai mRNA untuk memberi ruang bagi molekul tRNA lain untuk berkaitan dengan asam amino lain. Proses ini akan terus berlangsung sampai terdapat kodon stop yang dibaca pada mRNA. Pada manusia faktor penghentian adalah UAG, UAA, dan UGA. Ketika kodon stop tersebut dibaca maka rantai asam-asam amino (protein) tersebut akan dilepaskan dari struktur ribosom.

Apakah asam amino itu? Asam amino adalah senyawa organik dengan pusat atom karbon yang terikat dalam ikatan kovalen.



Gambar 2.4 RNA ke Rekaman Protein yang diadaptasikan Oleh Tim Wagner

- atom hidrogen tunggal : H
- gugus amino : NH₂
- gugus karboksil : COOH
- rantai samping, yang berbeda untuk tiap asam amino

Rantai samping mencakup kompleksitas dari atom hidrogen tunggal (asam amino glisin) sampai dengan struktur yang memiliki 18 atom (arginin). Jadi tiap asam amino terdiri dari antara 10 sampai 27 atom. Tepatnya terdapat dua puluh asam amino berbeda (terkadang disebut sebagai asam amino yang umum) yang digunakan pada produksi protein seperti yang diuraikan di atas. Sepuluh di antara asam amino tersebut disebut asam amino esensial karena asam-asam amino tersebut tidak dapat dihasilkan oleh tubuh sendiri dan oleh karena itu harus diperoleh dari sumber pangan (arginin merupakan asam amino yang penting bagi bayi dan pertumbuhan anak-anak). Sembilan asam amino bersifat hidrofil (dapat larut air) dan delapan asam amino bersifat hidrofobik (tiga yang lain disebut "spesial"). Pada jenis asam amino hidrofil, dua asam amino memiliki muatan jaring negatif pada rantai sampingnya sehingga bersifat asam (*acidic*), tiga asam amino memiliki muatan jaring positif sehingga bersifat basa; dan empat asam amino memiliki rantai samping tak bermuatan. Pada umumnya rantai samping memiliki kesemua dari atom berikut, yaitu atom hidrogen, nitrogen, karbon, dan oksigen. Meskipun dua asam amino (yaitu sistein dan metionin) juga memiliki atom belerang.

Terdapat dua puluh asam amino umum yang berbeda yang diperlukan untuk mengkodekan hanya empat basa yang berbeda. Bagaimana ini dilaksanakan? Sebagai kesatuan tunggal nukleotida (A,C,T, atau G) hanya dapat mengkodekan untuk empat asam amino, tentu saja jumlah ini tidak cukup. Sebagai pasangan nukleotida dapat mengkodekan untuk 16 (4²) asam amino, jumlah ini masih belum cukup juga. Secara triplet dapat dikodekan 64 (4³) asam amino yang mungkin. Cara ketiga ini yang digunakan oleh tubuh manusia, dan untaian dari tiga nukleotida itu disebut kodon. Mengapa ini dilaksanakan? Bagaimana mungkin evolusi mengembangkan kode yang tidak efisien seperti itu dengan banyak pemborosan? Terdapat banyak kodon untuk satu asam amino tunggal dengan dua alasan biologi utama : terdapat berbagai jenis tRNA dengan antikodon berbeda untuk membawa asam amino tertentu ke ribosom, dan kesalahan pasangan dapat terjadi selama proses translasi (hal ini disebut *wobble/* goyangan).

Kodon, yaitu untaian tiga nukleotida, menjadi kode untuk asam amino. Pada tabel di bawah ini terdapat kode genetik dari kodon mRNA menjadi protein, dan berbagai sifat asam amino¹. Pada tabel di bawah ini, tanda * mewakili (U, C, A, atau G); CU* dapat berarti CUU, CUC, CUA, atau CUG.

¹*shown adalah one-letter singkatan untuk masing-masing, Rezaot molekular nya, dan sebagian dari kekayaan nya, mengambil dari H. Lodish, D. Baltimore, A. Berk, S. L. Zipursky, P. Matsudaira, dan J. Darnell, " Biologi Sel Molekular," edisi ketiga, W. H. Freeman dan Company, New York, NY; 1995.*

Tabel 2.7 Tabel Singkat Asam Amino

		Second Nucleotide Base of mRNA Codon			
		U	C	A	G
First Nucleotide Base of mRNA Codon	U	UUU = Phe UUC = Phe UUA = Leu UUG = Leu	UC* = Ser	UAU = Tyr UAC = Tyr UAA = stop UAG = stop	UGU = Cys UGC = Cys UGA = stop UGG = Trp
	C	CU* = Leu	CC* = Pro	CAU = His CAC = His CAA = Gln CAG = Gln	CG* = Arg
	A	AUU = Ile AUC = Ile AUA = Ile AUG = Met (start)	AC* = Thr	AAU = Asn AAC = Asn AAA = Lys AAG = Lys	AGU = Ser AGC = Ser AGA = Arg AGG = Arg
	G	GU* = Val	GC* = Ala	GAU = Asp GAC = Asp GAA = Glu GAG = Glu	GG* = Gly

Tabel 2.8 Tabel Asam Amino dan Beberapa Sifatnya

Symbols	Amino Acid	M Wt	Properties		Codon(s)	
Ala	A	Alanine	89.09	Non-essential	Hydrophobic	GC*
Arg	R	Arginine	174.20	Essential	Hydrophilic, basic	CG* AGA AGG
Asn	N	Asparagine	132.12	Non-essential	Hydrophilic, uncharged	AAU AAC
Asp	D	Aspartic Acid	133.10	Non-essential	Hydrophilic, acidic	GAU GAC
Cys	C	Cysteine	121.15	Non-essential	Special	UGU UGC
Gln	Q	Glutamine	146.15	Non-essential	Hydrophilic, uncharged	CAA CAG
Glu	E	Glutamic Acid	147.13	Non-essential	Hydrophilic, acidic	GAA GAG
Gly	G	Glycine	75.07	Non-essential	Special	GG*
His	H	Histidine	155.16	Essential	Hydrophilic, basic	CAU CAC
Ile	I	Isoleucine	131.17	Essential	Hydrophobic	AUU AUC AUA
Leu	L	Leucine	131.17	Essential	Hydrophobic	UUA UUG CU*
Lys	K	Lysine	146.19	Essential	Hydrophilic, basic	AAA AAG
Met	M	Methionine	149.21	Essential	Hydrophobic	AUG
Phe	F	Phenylalanine	165.19	Essential	Hydrophobic	UUU UUC
Pro	P	Proline	115.13	Non-essential	Special	CC*
Ser	S	Serine	105.09	Non-essential	Hydrophilic, uncharged	UC* AGU AGC
Thr	T	Threonine	119.12	Essential	Hydrophilic, uncharged	AC*
Trp	W	Tryptophan	204.23	Essential	Hydrophobic	UGG
Tyr	Y	Tyrosine	181.19	Non-essential	Hydrophobic	UAU UAC
Val	V	Valine	117.15	Essential	Hydrophobic	GU*
start		Methionine				AUG
stop						UAA UAG UGA

2.8 Detail: IP Address

Tabel ini merupakan dari IPv4, <http://www.iana.org/assignments/ipv4-address-space> (versi 4, yang sedang dalam proses bertahap menjadi versi 6). IP address ditetapkan oleh *Internet Assigned Numbers Authority* (IANA) yang dapat dikunjungi di alamat <http://www.iana.org/>.

Berdasarkan pada University of Southern California's Information Sciences Institute, <http://www.isi.edu/>, IANA adalah yang bertanggung jawab atas semua "parameter unik" pada internet, yang mencakup *IP (Internet Protokol) address*. Setiap nama domain diasosiasikan dengan suatu *IP address* yang unik, yang merupakan suatu nama numerik yang terdiri dari empat blok yang masing-masingnya dapat terdiri sampai tiga digit, misalnya adalah 204.146.46.8, yang merupakan sistem yang digunakan untuk mengarahkan informasi dalam jaringan.

Ruang *Internet Protokol Adress*

Alokasi ruang IPv4 (InternetProtokol versi 4) menempati berbagai registrasi seperti yang terdaftar di bawah ini. Aslinya, semua *space* IPv4 diatur secara langsung oleh IANA. Di kemudian hari, sebagian dari *space* alamat dialokasikan ke berbagai kantor pendaftaran lain untuk pengaturan bagi kegunaan atau daerah regional tertentu di dunia. Dokumen RFC 1466 merupakan tempat kebanyakan alokasi.

Tabel 2.9 Daftar Parsial Penempatan IP Address

Address Block	Registry - Purpose	Date
000/8	IANA - Reserved	Sep 81
001/8	IANA - Reserved	Sep 81
002/8	IANA - Reserved	Sep 81
003/8	General Electric Company	May 94
004/8	Bolt Beranek and Newman Inc.	Dec 92
005/8	IANA - Reserved	Jul 95
006/8	Army Information Systems Center	Feb 94
007/8	IANA - Reserved	Apr 95
008/8	Bolt Beranek and Newman Inc.	Dec 92
009/8	IBM	Aug 92
010/8	IANA - Private Use	Jun 95
011/8	DoD Intel Information Systems	May 93
012/8	AT & T Bell Laboratories	Jun 95
013/8	Xerox Corporation	Sep 91
014/8	IANA - Public Data Network	Jun 91
015/8	Hewlett-Packard Company	Jul 94
016/8	Digital Equipment Corporation	Nov 94
017/8	Apple Computer Inc.	Jul 92
018/8	MIT	Jan 94
019/8	Ford Motor Company	May 95
020/8	Computer Sciences Corporation	Oct 94
021/8	DDN-RVN	Jul 91
022/8	Defense Information Systems Agency	May 93
023/8	IANA - Reserved	Jul 95
024/8	IANA - Cable Block	Jul 95
025/8	Royal Signals and Radar Establishment	Jan 95
	⋮	