

# APLIKASI TEORI BILANGAN BULAT PADA SISTEM BARCODE

Arga Dinata – 13505066

*Program Studi Informatika  
Sekolah Teknik Elektro dan Informatika  
Insitut Teknologi Bandung*

E-Mail: [if15066@students.if.itb.ac.id](mailto:if15066@students.if.itb.ac.id)

## Abstrak

Di dalam makalah ini akan dibahas mengenai aplikasi teori bilangan bulat pada metode pengkodean *barcode*. *Barcode* yang dalam Bahasa Indonesia dapat disebut dengan kode baris (yang selanjutnya akan kita gunakan istilah ini), merupakan sebuah kode sederhana yang digunakan untuk menyimpan data-data spesifik, seperti: kode produksi, tanggal kadaluwarsa, maupun nomor identitas. Kode baris pada dasarnya terdiri dari susunan garis vertikal hitam (bar) dan putih (spasi) dengan ketebalan yang berbeda-beda yang merepresentasikan karakter-karakter *alpha-numerik* (*alphabet* dan numerik). Pembahasan mengenai aplikasi teori bilangan bulat pada sistem kode baris dalam makalah ini akan mencakup: berbagai aplikasi teori bilangan bulat yang digunakan pada bentuk dan jenis pengkodean kode baris yang berbeda-beda. Selain itu, makalah ini juga memberikan gambaran tentang bentuk-bentuk *kode baris* beserta jenisnya, sistem *kode baris*, serta penjelasan singkat mengenai cara kerja pemindai kode baris (*barcode scanner*).

**Kata kunci:** kode baris, *barcode scanner*.

## 1. Pendahuluan

Mungkin tanpa kita sadari, sebenarnya kode baris banyak kita jumpai pada kehidupan sehari-hari. Misalnya pada produk-produk makanan, obat-obatan, tiket pesawat, kartu tanda mahasiswa, bahkan pada sampul surat yang kita terima (biasanya yang berasal dari luar negeri) pun dapat kita temui kode baris.

Pada awal perkembangannya, penggunaan kode baris dilakukan untuk membantu proses pemeriksaan barang-barang secara otomatis pada pasar-pasar swalayan. Namun, pada saat ini kode baris sudah banyak digunakan pada kartu identitas, kartu kredit, maupun untuk pemeriksaan secara otomatis pada perpustakaan.

Pada dasarnya kode baris terdiri atas susunan garis-garis vertikal hitam (bar) dan putih (spasi) dengan ketebalan yang berbeda-beda. Selain itu, kode baris juga dapat digambarkan dengan angka 1 untuk melambangkan garis hitam dan 0 untuk garis putih. Misalnya

0011001 merepresentasikan spasi-spasi-garis-garis-spasi-spasi-garis. Garis-garis ini digambarkan berderet secara horisontal dan merupakan representasi karakter-karakter *alpha-numerik* (*alphabet* dan numerik). Untuk membantu pembacaan kode baris secara manual, biasanya dicantumkan juga angka-angka atau huruf-huruf di bawah kode baris tersebut. Angka-angka tersebut tidak mendasari pola kode baris yang tercantum. Ukuran dari kode baris tersebut dapat diperbesar atau diperkecil dari ukuran nominalnya tanpa tergantung dari mesin yang membacanya (Mardiana, 1996).

Kode baris ini merupakan sebuah sistem pengkodean yang sangat sederhana namun sangat berguna. Fungsi dari kode baris ini adalah menyimpan data-data spesifik seperti: kode produksi, tanggal kadaluwarsa, maupun nomor identitas. Tujuan pokok dari kode baris adalah mengidentifikasi sesuatu dengan memberi label yang berisi kode baris. Kode baris pada umumnya digunakan pada aplikasi basis data dimana data pada kode baris hanya

memuat indeks basis data, yang menghubungkan pada basis data yang memuat informasi yang lebih lengkap. Sebagai contoh, pegawai toko memindai kode baris pada suatu produk, data pada kode baris terhubung ke komputer kemudian mencari informasi di basis data yang memuat informasi yang lebih detail, termasuk deskripsi barang beserta harganya. Dengan kode baris, pihak toko tidak perlu mencantumkan harga dari tiap-tiap produk. Dan jikalau ada perubahan harga, pihak toko hanya perlu mengubah harga yang tercantum pada basis data.

Walaupun teknologi pengkodean semacam ini terus mengalami perkembangan dan kemajuan, seperti ditemukannya *media magnetic*, *rfd*, *electronics tags*, maupun *serial eeprom* (pada *smart card*), kode baris terus bertahan dan masih memiliki kelebihan-kelebihan tertentu. Kelebihan utama dari kode baris adalah aplikasinya yang mudah dan biayanya yang relatif murah. Hal ini dikarenakan kode baris hanya membutuhkan media kertas dan tinta dalam pembuatannya. Sedangkan media untuk membacanya ada begitu banyak pilihan di pasaran dengan harga yang relatif murah. Mulai dari yang berbentuk seperti pena (*wand*), *slot*, pemindai (*scanner*), sampai *CCD*, atau bahkan kita dapat membuatnya sendiri. *Barcode scanner* dapat membaca informasi atau data dengan kecepatan yang jauh lebih tinggi daripada jika kita mengetikkan data. Selain itu, *barcode scanner* memiliki tingkat ketelitian yang lebih tinggi (Galbiati, Jr., 1990).

Terdapat banyak metode untuk membuat label kode baris. Salah satunya adalah Metode EAN-13, metode ini sering dijumpai dalam bisnis atau jual-beli di Indonesia. Metode ini hanya mengkodekan 13 digit data numerik yang merupakan gabungan dari kode nomor sistem, kode perusahaan, kode produk, dan satu digit untuk pengecekan. Lain halnya dengan metode Code 39, yang dapat mengkodekan huruf kapital, angka, dan beberapa karakter tertentu. Panjang data yang dikodekan bebas, selama tidak melebihi kemampuan alat pembaca kode baris.

## 2. Bentuk Barcode

### 2.1. Barcode Satu Dimensi

Kode baris satu dimensi biasa disebut kode baris linier (*linier barcode*). Beberapa contoh dari kode baris satu dimensi adalah sebagai berikut:

#### 2.1.1. Code 39

Salah satu bentuk kode baris adalah Code 39 atau dapat juga disebut Code 3 of 9. Code 39 dapat mengkodekan karakter *alpha-numerik*, yaitu angka desimal, huruf besar, serta karakter tambahan: *-.\*/+-%\$*.

Satu karakter dalam Code 39 terdiri dari 9 elemen, yaitu 5 bar dan 4 spasi yang disusun bergantian antara bar dengan spasi. Tiga dari sembilan elemen tersebut memiliki ketebalan yang lebih dari yang lainnya. Oleh karena itu, metode ini juga dapat disebut Code 3 of 9. Elemen yang lebih tebal tersebut terdiri dari 2 bar dan 1 spasi. Elemen yang lebar mewakili digit biner 1, sedangkan elemen yang sempit mewakili digit biner 0. Tabel karakter Code 39 beserta nilainya dapat dilihat pada tabel 2 halaman 3.

QZ	Gambar 1 adalah gambar Struktur Kode Baris dimana:
SC	X ketebalan elemen yang sempit (minimum 0.19 mm)
ICG	QZ <i>Quiet Zone</i> atau <i>Start-Pop Margin</i> dengan ketebalan min. 6 mm atau 10 kali X
CI	SC <i>Start Character</i>
ICG	ICG <i>Inter-Character Gap</i> dengan ketebalan sama dengan X
C2	CC <i>Check Character</i>
ICG	PC <i>Stop Character</i>
...	
CN	C1..CN Karakter ke-1 s/d ke-N
ICG	
CC	
ICG	
PC	
QZ	

Gambar 1

Untuk dapat membedakan garis vertikal lebar dan sempit maka perbandingan ketebalan antara garis vertikal yang lebar dengan yang sempit minimum 2:1, tapi perbandingan 3:1 akan lebih baik. Lebar keseluruhan kode baris dapat dirumuskan sebagai berikut:

$$L = N(3RX + 7X) + (6RX + 13X) + (3RX + 7X) + (M1 + M2)$$

I
II
III
IV

dimana:

- L lebar keseluruhan kode baris
- N jumlah karakter

Karakter ASCII	Nilai Karakter
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15
G	16
H	17
I	18
J	19
K	20
L	21
M	22
N	23
O	24
P	25
Q	26
R	27
S	28
T	29
U	30
V	31
W	32
X	33
Y	34
Z	35
-	36
.	37
space	38
*	-
%	39
/	40
+	41
%	42

Tabel 1

- R perbandingan garis vertikal lebar dengan sempit
- X ketebalan garis vertikal sempit
- I lebar N karakter ditambah *Inter-Character Gap*
- II lebar *Start* dan *Stop Character* ditambah *Inter-Character Gap* antara *Start Character* dengan karakter pertama
- III lebar *Check Character* ditambah *Inter-Character Gap*
- IV lebar dua kali *Quiet Zone* ( $M1(\text{Start Margin}) + M2(\text{Stop Margin})$ )

Cek karakter adalah sisa dari jumlah seluruh nilai karakter dibagi dengan 43, sebagai contoh:

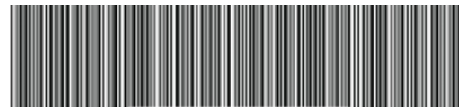
*Message* : CODE 39  
 Karakter : C O D E space 3 9  
 Nilai Karakter : 12 24 13 14 38 3 9  
 Jumlah : 12+24+13+14+38+3+9  
 =139

Nilai karakter uji :  $139 \text{ mod } 43 = 27$

Jadi nilai karakter ujinya adalah 27, yang jika kita lihat dalam tabel 1 merupakan nilai dari karakter R, maka:

*Message+Check Character* : CODE 39R

Gambar 2 adalah contoh kode baris Code 39.



Gambar 2

### 2.1.2. Interleaved 2 of 5 (ITF)

Kode baris ITF hanya dapat mengkodekan angka saja, dan sering digunakan pada produk-produk yang memiliki kemasan yang tidak rata. Hal ini dikarenakan cara pengkodean ITF yang tergolong unik.

Setiap karakter pada kode baris ITF dikodekan dengan 5 elemen, yaitu 2 elemen tebal dan 3 elemen sempit. Elemen tebal mewakili digit biner 1, sedangkan yang sempit mewakili digit biner 0. Perbandingan ketebalan antara yang tebal dengan yang sempit adalah 2:1 s/d 3:1.

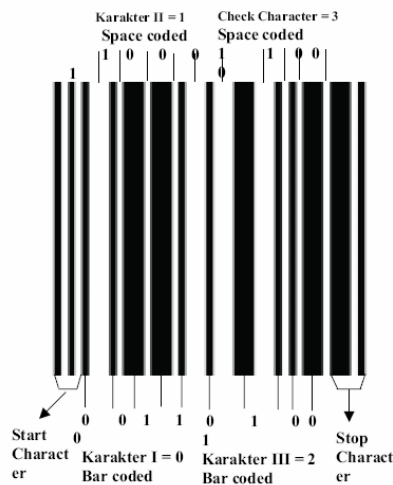
Keunikan dari kode baris ITF adalah cara pengkodeannya. Penggunaan bar atau spasi ditentukan dari posisinya, sesuai dengan namanya, yaitu *interleaved*. Atau lebih jelasnya sebagai berikut: karakter pertama dikodekan menggunakan bar setelah *start character*, sedangkan karakter kedua dikodekan menggunakan spasi secara *interleaved* pada karakter pertama, karena sifat berpasang-pasangan itulah panjang *message* yang termasuk *check character* haruslah genap. Jika jumlahnya ganjil, maka harus ditambahkan karakter 0 pada awal *message*. Berikut adalah tabel nilai karakter pada kode baris ITF.

Karakter ASCII	Nilai Karakter
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
start	-
stop	-

Tabel 2

\*Start dan Stop Character tidak dikodekan secara *interleaved*, tetapi bergantian bar dengan spasi.

Berikut adalah contoh kode baris ITF:



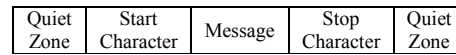
Gambar 3

Contoh penghitungan *check character* pada kode baris ITF adalah sebagai berikut:

Message	2632534
Karakter	2 6 3 2 5 3 4
Posisi	e o e o e o e
	(karakter paling kanan dimulai sebagai e:even)
Jmlh nilai odd	6+2+3=11
Jmlh nilai even	2+3+5+4=14
3 kali even	3X14=42
Jmlh o+3e	11+42=53
Check-Character	10-(x mod 10)= 10-(53 mod 10)=7
Tag to be Encoded	263225347

Jumlah keseluruhan karakter yang termasuk *check character* sudah berjumlah genap, sehingga tidak diperlukan *leading zero*.

Struktur kode baris ITF dapat dilihat pada gambar di bawah ini:



Gambar 4

Lebar dan tinggi kode baris ITF mengikuti aturan di bawah ini:

$$L=(C(2R+3)+6+R)X$$

Dimana:

- L lebar kode baris, tidak termasuk *Quiet Zone*
- C jumlah karakter
- X lebar elemen tersempit (minimum 0.19 mm)
- R perbandingan elemen lebar dengan elemen sempit (2:1 s/d 3:1)

*Quiet Zone* minimum 10 kali X.

Tinggi kode baris minimum 0.15 kali lebar kode baris.

Untuk menghindari terjadinya pemindaian parsial (*partial scanning*) atau pembacaan kode baris yang terpotong (dapat mengakibatkan kesalahan informasi), maka seringkali kode baris ITF diberi tambahan *bearer bar*, yaitu bar sepanjang kode baris yang ditempatkan di atas dan di bawah kode baris. Seperti yang diperlihatkan di bawah ini:



Gambar 4

### 2.1.3. Code 128

Code 128 adalah kode baris dengan kerapatan yang tinggi, dapat mengkodekan seluruh karakter ASCII (128 karakter) dalam lingkup yang paling minimum diantara jenis kode baris yang lain. Hal ini dikarenakan Code 128 menggunakan 4 ketebalan elemen

(bar atau spasi) yang berbeda-beda (jenis yang lain biasanya hanya menggunakan 2 macam ketebalan elemen). Setiap karakter pada Code 128 dikodekan dengan 3 bar dan 3 spasi (6 bar keseluruhan) dengan ketebalan masing-masing elemen 1 sampai 4 kali ketebalan minimum (*module*). Jika dihitung dengan satuan *module*, maka tiap karakter Code 128 terdiri dari 11 *module*, kecuali untuk *stop character* yang terdiri dari 4 bar dan 3 spasi (13 *module*). Jumlah total *module* untuk bar selalu genap, sedangkan untuk spasi selalu ganjil. Selain itu, Code 128 memiliki 3 *start character* yang berbeda, sehingga Code 128 memiliki 3 *sub-set character*-nya. Seperti yang diperlihatkan pada tabel di bawah ini:

Value	Set A	Set B	Set C
0	SP	SP	00
1	!	!	01
2	“	“	02
3	#	#	03
4	\$	\$	04
5	%	%	05
6	&	&	06
7	‘	‘	07
8	(	(	08
9	)	)	09
10	*	*	10
11	+	+	11
12	,	,	12
13	-	-	13
14	.	.	14
15	/	/	15
16	0	0	16
17	1	1	17
18	2	2	18
19	3	3	19
20	4	4	20
21	5	5	21
22	6	6	22
23	7	7	23
24	8	8	24
25	9	9	25
26	:	:	26
27	;	;	27
28	<	<	28
29	=	=	29
30	>	>	30
31	?	?	31
32	@	@	32
33	A	A	33
34	B	B	34
35	C	C	35
36	D	D	36
37	E	E	37
38	F	F	38
39	G	G	39
40	H	H	40

Value	Set A	Set B	Set C
41	I	I	41
42	J	J	42
43	K	K	43
44	L	L	44
45	M	M	45
46	N	N	46
47	O	O	47
48	P	P	48
49	Q	Q	49
50	R	R	50
51	S	S	51
52	T	T	52
53	U	U	53
54	V	V	54
55	W	W	55
56	X	X	56
57	Y	Y	57
58	Z	Z	58
59	[	[	59
60	\	\	60
61	]	]	61
62	^	^	62
63			63
64	NUL	‘	64
65	SOH	a	65
66	STX	b	66
67	ETX	c	67
68	EOT	d	68
69	ENQ	e	69
70	ACK	f	70
71	BEL	g	71
72	BS	h	72
73	HT	i	73
74	LF	j	74
75	VT	k	75
76	FF	l	76
77	CR	m	77
78	SO	n	78
79	SI	o	79
80	DLE	p	80
81	DC1	q	81
82	DC2	r	82
83	DC3	s	83
84	DC4	t	84
85	NAK	u	85
86	SYN	v	86
87	ETB	w	87
88	CAN	x	88
89	EM	y	89
90	SUB	z	90
91	ESC	{	91
92	FS		92
93	GS	}	93
94	RS	~	94
95	US	DEL	95
96	FNC3	FNC3	96
97	FNC2	FNC2	97
98	SHIFT	SHIFT	98
99	CODE C	CODE C	99
100	CODE B	CODE B	CODE B
101	FNC 4	FNC 4	CODE A
102	FNC 1	FNC 1	FNC 1

Value	Set A	Set B	Set C
103	START A	START A	START A
104	START B	START B	START B
105	START C	START C	START C
106	STOP	STOP	STOP

Tabel 3

Code 128 memiliki fitur untuk dapat beralih dari satu *sub-set* ke *sub-set* yang lain dengan menggunakan karakter CODE dan SHIFT. CODE X menyebabkan seluruh *message* beralih menjadi *sub-set* X (misalnya, CODE A pada *sub-set* B membuat *message* beralih ke *sub-set* B). Sedangkan SHIFT menyebabkan satu karakter di depannya beralih *sub-set* (ini hanya berlaku untuk *sub-set* A ke *sub-set* B atau sebaliknya).

Struktur dari kode baris Code 128 seperti yang terlihat di bawah ini:



Gambar 5

Tinggi kode baris minimum adalah 0.15 kali lebar baris kode. Lebar kode baris dinyatakan dengan aturan di bawah ini:

$L=X(11C+35)$  untuk alpha-numerik (CODE A & B)  
 $L=X(5.5C+35)$  untuk double density numeric (C)

Dimana:

L lebar kode baris termasuk *Quiet Zone*  
 C jumlah karakter  
 X lebar *module*

Code C disebut sebagai *double density numeric*. Dalam perhitungan, lebar per karakternya hanya 5.5 kali X, sebab satu karakter CODE C mewakili 2 digit numerik (tabel 3).

Penghitungan *check character* pada kode baris 128 adalah sebagai berikut:

Message CODE 128

Karakter START A C O D E 1 2 8  
 Nilai 103 35 47 36 37 0 17 18 24  
 Posisi - 1 2 3 4 5 6 7 8  
 Jmlh  $103+(1X35)+(2X47)+(3X36)+$   
 $(4X37)+(0X5)+(6X17)+(7X18)+$   
 $(8X24)=908$

Check  $908 \text{ mod } 103=84$

Pada tabel 3, 84 adalah nilai dari karakter DC4 pada sub-set A. Jadi message akhir-nya adalah (START A)CODE 128(DC4)(STOP).

### 2.1.4. Universal Product Code (UPC)

UPC digunakan secara luas pada industri grosir, khususnya di Amerika Serikat dan Kanada. Pada standar kode baris-nya, yaitu UPC-A, terdiri atas: 1 digit nomor sistem pada awal kode baris, 5 digit nomor manufaktur, 5 digit nomor produk, serta 1 digit cek. Nomor sistem menunjukkan penggunaan satu diantara sepuluh nomor sistem yang telah ditetapkan oleh UPC, yaitu:

- 0, 6, dan 7 untuk kode UPC reguler
- 2 untuk barang-barang di toko
- 3 untuk obat-obatan dan alat-alat kesehatan
- 4 untuk barang selain makanan
- 5 untuk penggunaan karcis dan kupon
- 1, 8, 9 masih belum digunakan

#### 2.1.4.1. Penghitungan Digit Cek UPC-A

Digit cek dihitung dari hasil modulus 10. Berikut adalah langkah-langkah untuk menghitung digit cek kode baris UPC-A:

1. Nilai digit 1, 3, 5, 7, 9, dan 11 dijumlahkan.
2. Nilai digit 2, 4, 6, 8, dan 10 juga dijumlahkan
3. 3 kali hasil langkah 1.
4. Menjumlahkan hasil langkah 2 dan 3.
5. Digit cek= $10 - ((\text{hasil } 4) \text{ mod } 10)$ .

Sebuah contoh sederhana, diberikan kode baris UPC-A: 01234567890.

1.  $0+2+4+6+8+0=20$
2.  $1+3+5+7+9=25$
3.  $3X20=60$ .
4.  $60+25=85$ .
5.  $X=10 - (85 \text{ mod } 10)$ .  
 $X=5$  (digit cek).



Gambar 6

### 2.1.4.2. Pengkodean Simbol UPC-A

Pengkodean simbol pada UPC-A adalah sebagai berikut: angka 1 mewakili baris, sedangkan angka 0 mewakili spasi. Oleh karena itu, kode 1101 mewakili satu baris lebar (dua buah baris), diikuti dengan spasi pendek dan kemudian baris pendek. Jika digambarkan akan seperti pada gambar 7.



Gambar 7

- UPC-A mempunyai struktur sebagai berikut:
1. Batas kiri, dikodekan dengan 101.
  2. Karakter pertama dari kode nomor sistem dikodekan seperti pada tabel 4 di bawah.
  3. 5 kode manufaktur dikodekan seperti pada tabel 4.
  4. Batas tengah, dikodekan dengan 01010.
  5. 5 karakter kode produk, dikodekan sebagai karakter *right-hand*.
  6. Batas kanan, dikodekan dengan 101.

Karakter yang dikodekan pada bagian kiri batas tengah disebut *left-hand side*, sedangkan yang di sebelah kanan disebut *right-hand side*. Tabel 4 menggambarkan bagaimana mengkodekan setiap digit UPC-A bergantung pada bagian kiri atau kanan kode baris.

Digit	Left-Hand	Right-Hand
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

Tabel 4

### 2.1.4.3. Pengkodean Simbol UPC-E

UPC-E digunakan pada barang yang dikemas dengan pak atau bungkus yang ukurannya sangat kecil. UPC-E menggambarkan keseluruhan simbol UPC-A. Namun, simbol UPC-A yang terdiri dari bilangan nol tertentu yang dapat diringkas menjadi simbol UPC-E. Jumlah nomor produk ditentukan oleh bentuk dan jumlah angka nol pada nomor manufakturnya.

Pengkodean simbol pada UPC-E adalah sebagai berikut: angka 1 mewakili baris, sedangkan angka 0 mewakili spasi. Digit cek pada UPC-E tidak secara langsung dikodekan sebagai baris atau spasi pada kode baris, melainkan dikodekan dengan metode pengkodean ganjil-genap kode baris.

## 2.2 Barcode Dua Dimensi

Kode baris dua dimensi telah dikembangkan lebih dari sepuluh tahun yang lalu, namun baru sekarang mulai populer. Kode baris dua dimensi ini memiliki beberapa keunggulan dibandingkan kode baris satu dimensi (*linier barcode*). Yang paling utama adalah: kode baris dua dimensi ini dapat menyimpan data atau informasi dalam suatu ruang yang lebih kecil. Contoh kode baris dua dimensi adalah *Symbology PDF417* yang dapat menyimpan lebih dari 2000 karakter dalam sebuah ruang yang berukuran 4 inch persegi.

## 3. Metode Pengkodean Barcode

Metode pengkodean kode baris ada 2 macam, yaitu:

### 3.1 Binary Coding

*Binary coding* disebut juga pengkodean biner. Dalam metode ini digunakan dua bar dan spasi untuk meng-*encode*-kan data. Bar dan spasi ini akan dikonversi ke dalam bentuk bilangan biner. Bilangan biner ini kemudian akan dikonversi lagi ke dalam karakter-karakter ASCII.

### 3.2 Proportional Coding

Bar dan spasi memiliki beberapa ukuran yang berbeda. Ukuran serta urutan dari bar dan spasi merupakan representasi dari karakter-karakter kode baris. Kode tersebut lebih sukar diterjemahkan ke dalam bilangan biner. Ketelitian sangat diperlukan dalam mencetak dan memindai kode baris yang menggunakan metode pengkodean ini. Pada umumnya ada 4 ukuran yang berbeda pada bar dan spasi yang digunakan untuk meng-encode data. Contoh jenis kode baris yang menggunakan teknik *encoding* ini adalah Code 128.

Pengkodean data dalam sebuah kode baris adalah sebagai berikut: sebuah *fixed number* pada bar digunakan per karakter. Hal ini berarti bahwa jika sebuah bar tidak terbaca, maka kode baris tersebut tidak akan dapat dibaca. Jumlah karakter yang mungkin yang dapat di-encode dalam beberapa jenis kode baris lebih tinggi dibandingkan dengan jumlah karakter yang valid. Hal ini berarti bahwa jika ukuran dari salah satu bar atau spasi salah terbaca, maka karakter ilegal akan terbaca. Oleh karena itu, kecepatan substitusi karakter menjadi sangat rendah. Jadi, kesimpulannya kode baris sangat aman. Kesalahan dalam pembacaan kode kecil kemungkinan dapat terjadi.

bilangan 1 untuk menyatakan bar dan bilangan 0 untuk menyatakan spasi. Misalnya, kode baris dengan representasi 0011001 dapat dinyatakan sebagai spasi-spasi-bar-bar-spasi-spasi-bar.

Sebuah kode baris UPC, bilangan di sisi bagian kiri kode baris (kode perusahaan atau manufaktur) dikodekan berbeda dengan bilangan di sisi bagian kanan (kode produk). Bilangan yang berada di sebelah kiri merupakan kebalikan dari bilangan yang ada di sebelah kanan. Misalkan jika bar di sebelah kanan, berarti sebuah spasi di sebelah kiri. Pengkodean di sebelah kanan dinamakan kode *even parity*, sebab unit bar-nya berjumlah genap. Sedangkan pengkodean di sebelah kiri dinamakan kode *odd parity* sebab unit bar-nya berjumlah ganjil. Bilangan-bilangan yang dikodekan mempunyai perbedaan untuk tiap-tiap sisi kode baris, sehingga kode baris dapat dibaca (*scanned*) dari sebelah kiri maupun dari sebelah kanan.

Gambar 8 memperlihatkan pengkodean sisi kiri dan sisi kanan yang dipisahkan ke dalam tujuh unit.

LEFT SIDE (ODD PARITY) CODES									
1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567
0	1	2	3	4	5	6	7	8	9
0001101	0011001	0101001	0111101	0100011	0110001	0101111	0111011	0110111	0001011

RIGHT SIDE (EVEN PARITY) CODES									
1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567	1234567
0	1	2	3	4	5	6	7	8	9
1110010	1100110	1101100	1000010	1011100	1001110	1010000	1000100	1001000	1110100

Gambar 8

#### 4. Penerjemahan Barcode oleh Komputer

Suatu bilangan kode baris tunggal sebenarnya terdiri dari tujuh unit baris dan spasi. Satu unit terdiri dari salah satu warna hitam atau putih. Sebuah unit yang berwarna hitam ditunjukkan dengan sebuah bar, sedangkan yang berwarna putih ditunjukkan dengan sebuah spasi. Cara lain penulisan kode baris adalah dengan menggunakan

Penjelasan tabel pengkodean di atas adalah sebagai berikut: seperti yang sebelumnya disebutkan bahwa bilangan-bilangan di sebelah kiri merupakan kebalikan dari bilangan-bilangan di sebelah kanan. Setiap kode baris memiliki empat buah *mark* (marka) yang berbeda. Sebuah marka dapat terdiri dari salah satu bar atau spasi. Marka-marka tersebut lebarnya bermacam-macam, tetapi jumlahnya selalu empat. Contohnya,



bilangan pengkodean yang berada di sebelah kiri pada bagian angka 0 yaitu 0001101 berarti terdiri dari 3 spasi (marka 1), 2 bar (marka 2), 1 spasi (marka 3), dan 1 bar (marka 4). Pengkodean di sisi kiri selalu dimulai dengan sebuah spasi atau 0 dan berakhir dengan sebuah bar atau 1, sedangkan untuk sisi sebelah kanan selalu dimulai dengan sebuah bar atau 1 dan berakhir dengan sebuah spasi atau 0.

#### 4.1. Number System Character

Angka ini merupakan sebuah sistem bilangan kode baris UPC yang meng-karakterisasi-kan jenis-jenis khusus pada kode baris. Di dalam kode baris UPC, *Number System Character* ini biasanya terletak di sebelah kiri kode baris. Kode-kode pada *Number System Character* adalah sebagai berikut:

- 0 *Standard UPC number.*
- 1 *Reserved.*
- 2 *Random weight items like fruits, vegetables, and meats, etc.*
- 3 *Pharmaceuticals*
- 4 *In-store code for retailers.*
- 5 *Coupons*
- 6 *Standard UPC number.*
- 7 *Standard UPC number.*
- 8 *Reserved.*
- 9 *Reserved.*

#### 4.2. Manufacturer Code

Kode perusahaan ini ada lima digit bilangan yang secara khusus menentukan manufaktur suatu produk. Kode perusahaan atau manufaktur ini dilindungi dan ditetapkan oleh *Uniform Code Council (UCC)*.

#### 4.3. Three Guard Bars

Ada tiga *guard bars* yang ditempatkan di awal, tengah, dan akhir pada kode baris. *Guard bars* bagian awal dan akhir di-*encode*-kan sebagai bar-spasi-bar atau 101. *Guard bars* bagian tengah di-*encode*-kan sebagai spasi-bar-spasi-bar-spasi atau 01010.

#### 4.4. Product Code

Kode produk ini terdiri dari lima digit bilangan yang ditetapkan oleh perusahaan atau manufaktur untuk setiap produk yang dihasilkannya. Untuk setiap produk dan ukuran yang berbeda, akan memiliki kode produk yang berbeda pula.

#### 4.5. Check digit

*Check digit* disebut sebagai digit *self-check*. *Check digit* ini terletak di bagian luar sebelah kanan kode baris. *Check digit* ini merupakan suatu *old-programmer's trick* untuk mem-validasi-kan digit-digit lainnya (*number system character, manufacturer code, product code*) yang dibaca secara teliti.

Sedangkan cara komputer meng-kalkulasikan *check digit* adalah sebagai berikut: menjumlahkan semua digit-digit yang ganjil. Kemudian mengalikan hasil penjumlahan tersebut dengan bilangan 3. Menjumlahkan semua bilangan-bilangan genap. Kemudian hasil penjumlahan tersebut ditambahkan dengan hasil kali di atas. *Check digit* adalah angka yang harus ditambahkan pada digit terakhir penulisan kode baris. *Check digit* dapat dihitung dari hasil penjumlahan keseluruhan dengan modulus sepuluh. Angka ini kemudian digunakan untuk mengurangi bilangan sepuluh. Angka terakhir inilah yang disebut *check digit*.

Berikut merupakan contoh penghitungan *check digit* pada sebuah kode baris: misalkan ada sebuah kode baris 01234567890X dengan X sebagai *check digit*. Pertama-tama kita jumlahkan semua angka dengan digit ganjil, yaitu  $0+2+4+6+8+0=20$ . Hasil ini kita kalikan dengan 3,  $3 \times 20=60$ . Kemudian angka dengan digit genap juga kita jumlahkan,  $1+3+5+7+9=25$ . Angka ini kita jumlahkan dengan hasil perkalian di atas,  $60+25=85$ . Akhirnya, dengan hasil ini kita dapat menghitung *check digit* X dengan cara  $10-(85 \bmod 10)=5$ . Jadi *check digit* X yang valid adalah 5.

### 5. Kesimpulan

1. Kode baris adalah sebuah kode yang terdiri dari sejumlah bar (garis vertikal hitam) serta spasi (garis vertikal putih).
2. Kode baris digunakan untuk menyimpan data-data spesifik seperti: kode produksi, tanggal kadaluwarsa, maupun nomor identitas.
3. Kode baris memiliki berbagai macam bentuk, serta memiliki cara pengkodean yang beragam.
4. Teori bilangan bulat memiliki peran pada sistem pengkodean kode baris, khususnya pada penghitungan digit cek.

#### **Daftar Pustaka**

- [1] Planet Church, Barcodes, <[www.666soon.com/barcode.htm](http://www.666soon.com/barcode.htm)>, tanggal akses 27 Desember 2006.
- [2] Mardiana, 1996, *Pengembangan Perangkat Lunak untuk Pengolah Data dari Mesin Pembaca kode Baris ABX-10*, Skripsi, Jurusan Teknik Elektro, ITENAS Bandung.
- [3] Louis J. Galbiati, Jr., 1990, *Machine Vision and Digital Image Processing Fundamentals*, Englewood Cliffs, New Jersey, Prentice-Hall.
- [4] Indrawaty, Youllia, 2001, *Sebuah Gagasan Penggunaan Sistem Pengkodean Baris (Barcode) Sebagai Kunci Pendeteksian Uang Secara Otomatis*.