

Bab 9

Studi Kasus Konversi Bilangan

Pokok Bahasan

1. Konversi bilangan biner ke desimal
2. Konversi bilangan desimal ke biner

Tujuan

1. Memahami logika melakukan konversi bilangan
2. Memahami algoritma konversi bilangan
3. Menjelaskan pemakaian looping untuk konversi bilangan.
4. Menjelaskan pemakaian operasi stack untuk konversi bilangan.

9.1 . Studi Kasus 1: Konversi bilangan biner ke desimal

9.1.1 Permasalahan

Buatlah flowchart untuk konversi bilangan biner ke desimal (maksimum bilangan=11111111).

Contoh:

Masukkan bilangan biner : 00110101

Bilangan desimal : 53

9.1.2 Cara Penyelesaian Masalah

Konversi bilangan biner ke desimal dilakukan dengan cara mengalikan setiap elemen bilangan biner (bi) dengan deretan bilangan 2 yang berpangkat sebagai berikut:

$$b_1 * 2^{n-1} + b_2 * 2^{n-2} + \dots + b_{n-2} * 2^2 + b_{n-1} * 2^1 + b_n * 2^0$$

$$\text{atau} : b_1 * 2^{n-1} + b_2 * 2^{n-2} + \dots + b_{n-2} * 4 + b_{n-1} * 2 + b_n * 1$$

Dalam kasus permasalahan diatas, input bilangan biner yang dimasukkan sepanjang 8 elemen bilangan dapat diselesaikan sebagai berikut:

$$\begin{array}{c}
 00110101 \\
 \downarrow \\
 0*2^7 + 0*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 \\
 = 0*128 + 0*64 + 1*32 + 1*16 + 0*8 + 1*4 + 0*2 + 1*1 \\
 = 0 + 0 + 32 + 16 + 0 + 4 + 0 + 1 \\
 = 53
 \end{array}$$

9.1.3 Struktur Data Yang Dibutuhkan

- n sebagai variabel banyaknya elemen bilangan (dalam hal ini n=8)
- biner sebagai array bilangan biner dengan panjang n elemen
- hasil sebagai variabel penampung hasil konversi biner ke desimal
- i sebagai variabel penunjuk elemen array

9.1.4 Deklarasi dan Inisialisasi

Variabel terdiri dari n, biner, hasil dan i. Untuk n, pada permasalahan n=8. Sedangkan hasil, karena dipakai sebagai penampung hasil konversi, saat awal hasil harus diset menjadi 0. Dari sini, deklarasi dan inisialisasi variabel pada flowchart dapat dibuat sebagai berikut:

n=8, biner=[n],
hasil=0, i

9.1.5 Input

Bilangan biner yang terdiri dari 8 elemen, sehingga input pada flowchart dapat dibuat sebagai berikut:

Input biner

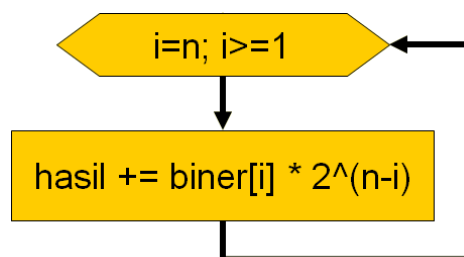
9.1.6 Output

Bilangan desimal hasil konversi dari bilangan biner yang disimpan pada variabel *hasil*, sehingga output pada flowchart dapat dibuat sebagai berikut:

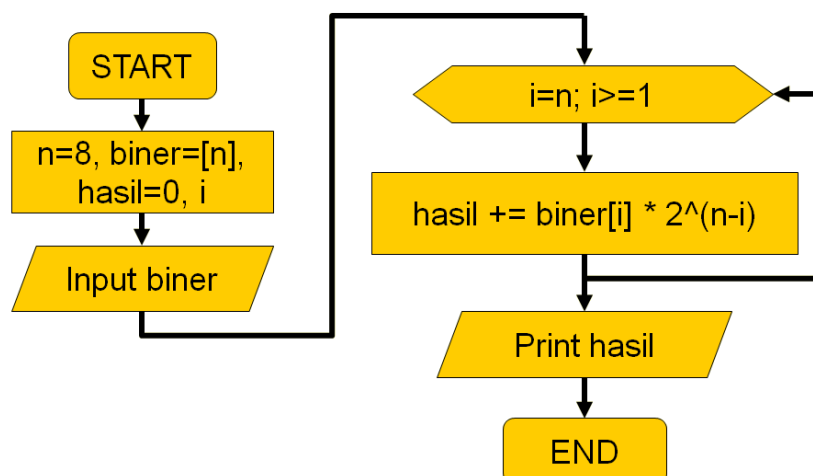


9.1.7 Proses Penyelesaian

Karena proses perhitungan konversi dilakukan dari kanan ke kiri, maka looping juga dilakukan menurun dari $i=n$ sampai $i=1$. Saat awal $i=n$, dilakukan perhitungan pada $b_n * 2^0$ dan hasilnya diakumulasikan pada *hasil*. Pada iterasi berikut nilai i menurun menjadi $i=n-1$, dilakukan perhitungan pada $b_{n-1} * 2^1$ dan hasilnya diakumulasikan pada *hasil*. Dari sini, dapat diambil kesimpulan bahwa pada setiap iterasi i , *hasil* dapat dihitung dengan rumusan $hasil = hasil + b_i * 2^{n-i}$. Dengan demikian, untuk proses dapat direpresentasikan dengan flowchart sebagai berikut:



9.1.8 Flowchart Keseluruhan



9.2 Studi Kasus 2: Konversi bilangan desimal ke biner

9.2.1 Permasalahan

Buatlah flowchart untuk konversi bilangan desimal ke biner (maksimum bilangan=255)

Contoh:

Masukkan bilangan desimal : 53

Bilangan biner : 110101

9.2.2 Cara Penyelesaian Masalah

Konversi bilangan desimal ke biner dilakukan dengan cara melakukan pembagian dengan nilai 2 dan mengambil sisa baginya. Sisa bagi tersebut kemudian disimpan pada suatu tempat kumpulan sisa bagi. Bilangan hasil bagi tersebut dibagi lagi dan diambil sisa baginya untuk kemudian disimpan pada tempat kumpulan sisa bagi. Demikian seterusnya sampai didapatkan hasil baginya sama dengan nol. Setelah itu, baru diambil sisa bagi dalam keadaan terbalik (yang masuk terakhir, diambil pertama kali) pada kumpulan sisa bagi. Proses pembacaan terbalik ini membutuhkan operasi stack.

Dalam kasus permasalahan diatas, dapat diselesaikan sebagai berikut:

Bilangan	Hasil bagi dengan 2	Sisa bagi dengan 2	Kumpulan sisa bagi
53	26	1	1
26	13	0	10
13	6	1	101
6	3	0	1010
3	1	1	10101
1	0	1	101011

Bilangan biner yang didapatkan adalah dari pembacaan kumpulan sisa bagi yang dibaca terbalik, sehingga hasilnya adalah 110101.

9.2.3 Struktur Data Yang Dibutuhkan

- bil sebagai variabel nilai bilangan
- sisa sebagai variabel nilai sisa bagi dengan 2
- stack sebagai penampung sisa bagi sehingga dapat dibaca terbalik

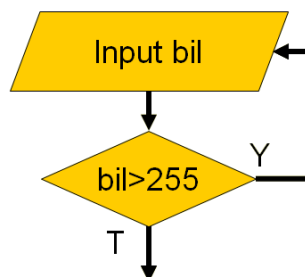
9.2.4 Deklarasi dan Inisialisasi

Variabel terdiri dari bil, sisa, dan stack. Dari sini, deklarasi dan inisialisasi variabel pada flowchart dapat dibuat sebagai berikut:

bil, sisa, stack

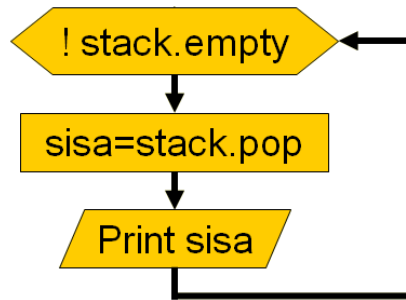
9.2.5 Input

Saat pertama kali, user diminta untuk memasukkan nilai bilangan bil yang akan dikonversi ke biner. Karena pada permasalahan terdapat syarat bahwa nilai bilangan tidak boleh lebih dari 255, maka perlu diberikan pengecekan apakah bil sesuai dengan syarat. Dari sini, flowchart Input dapat dibuat sebagai berikut:



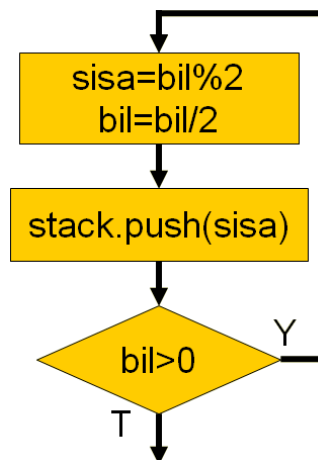
9.2.6 Output

Bilangan desimal hasil konversi dari bilangan desimal yang dikeluarkan dari *stack* dengan operasi pop sehingga dapat dibaca dalam keadaan terbalik. Setiap hasil pembacaan kemudian disimpan pada variabel *sisa*, sehingga output pada flowchart dapat dibuat sebagai berikut:

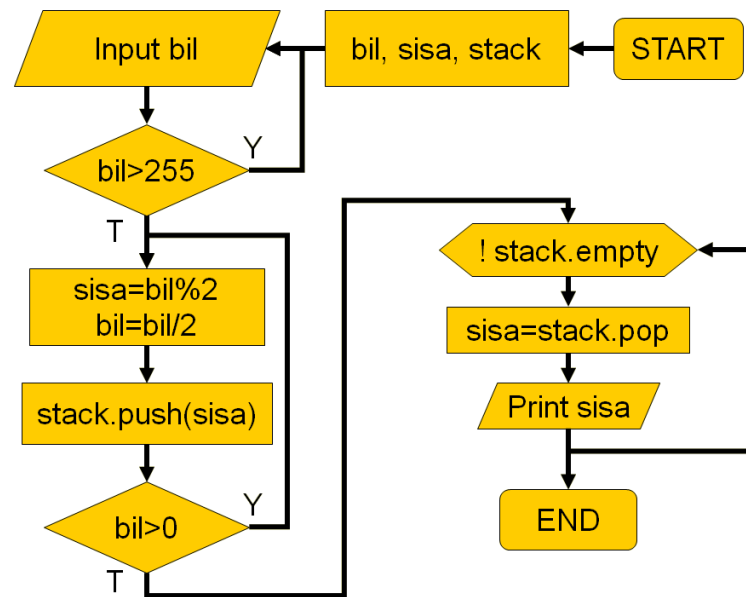


9.2.7 ProsesPenyelesaian

Sebagai dijelaskan pada sebelumnya, konversi bilangan desimal ke biner dilakukan dengan cara melakukan pembagian dengan nilai 2 dan mengambil sisa baginya, dimana proses ini dilakukan secara iteratif dan berhenti saat bilangan hasil bagi sama dengan 0. Setiap kali didapatkan sisa bagi, nilai tersebut dimasukkan ke dalam stack melalui operasi push. Adapun tujuan dipergunakannya stack adalah agar data yang masuk ke dalam stack, dapat dibaca dalam keadaan terbalik, sehingga ini berguna untuk membaca terbalik nilai sisa bagi yang dimasukkan ke dalam stack. Dari sini, flowchart Proses dapat direpresentasikan sebagai berikut:



9.2.8 Flowchart Keseluruhan



9.3 Latihan

9.3.1 Permasalahan

Buatlah flowchart untuk konversi bilangan desimal negatif ke biner (maksimum bilangan=-127).

Contoh:

Masukkan bilangan desimal : -1

Bilangan biner : 11111111

Masukkan bilangan desimal : -13

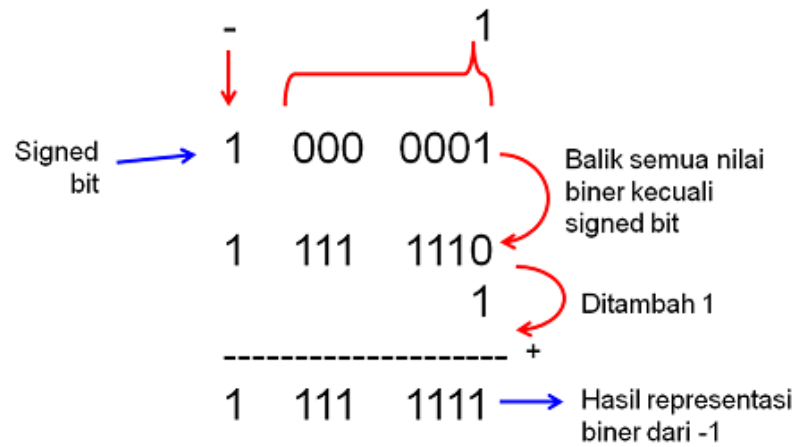
Bilangan biner : 11110011

9.3.2 Cara Penyelesaian Masalah

Karena batasan permasalahan bahwa maksimum bilangan adalah -127, maka total bit yang dipakai adalah sebanyak 8 bit. Pertama kali, signed bit (bit pada depan) diset dengan nilai 1 sebagai tanda negatif. Sisa 7 bit yang lainnya adalah nilai bit hasil konversi dari input bilangan desimal yang positif. Selanjutnya, 7 bit sisa tersebut (tidak termasuk signed bit) dibalik nilainya (0 menjadi 1, 1 menjadi 0). Kemudian 7 bit sisa ditambahkan dengan nilai

bit1. Susunan bit keseluruhannya adalah hasil dari konversi desimal negatif. Untuk lebih mudahnya, perhatikan ilustrasi berikut:

Masukkan bilangan desimal : -1



Masukkan bilangan desimal : -13

