

## MODUL MIPS: Lab 2

Pada minggu ini, kita akan berfokus mempelajari bagaimana mengolah serta memanipulasi sebuah array menggunakan bahasa MIPS assembly, seperti :

- Declare Array

Kita dapat men-declare suatu array pada .data

Contoh :

```
1  .data
2  array: .word 1,2,3,4,5
3  array_2: .word 0,0,0,0,0
```

- Memasukkan input ke dalam Array

Untuk memasukan input ke dalam suatu array kita perlu men-declare array terlebih dahulu di .data . Setelah itu, kita perlu mendapatkan address dari array tersebut dan menyimpannya ke dalam sebuah variabel.

```
.data
array: .word 0,0,0,0,0

.text
.globl main

main:
la $t0, array
```

Setelah menyiapkannya, kita siap untuk menerima input lalu memasukkannya ke dalam setiap indeks di array tersebut. Perlu diingat bahwa banyaknya bit pada 1 indeks adalah sebanyak 4. Sehingga untuk pindah ke indeks selanjutnya kita perlu menambahkan 4 pada addressnya.

Contoh :

Menginput 5 digit angka ke dalam array t0 diatas

```
li $v0,5                # meminta input integer
syscall                 # execute

add $t1, $v0, $zero     # memindahkan input dari $v0 ke $t1
addi $t2, $zero,0       # membuat variabel sebagai counter

input_loop:
div $t1,$t1,10          #membagi t1 dengan 10
mflo $t1                #hasil simpan ke t1
mfhi $t4                #sisia simpan t4

sw $t4,0 ($t0)          #store sisa pembagian ke $t0
addi $t0,$t0,4          #memindahkan pointer ke indeks selanjutnya
addi $t2,$t2,1          #mengurangi counter loop
bne $t2,5,input_loop   #looping
```

Alur Loop:

Loop	Input	Array
ke-0	12345	[0,0,0,0,0]
ke-1	1234	[5,0,0,0,0]
ke-2	123	[5,4,0,0,0]
ke-3	12	[5,4,3,0,0]
ke-4	1	[5,4,3,2,0]
ke-5	0	[5,4,3,2,1]

- Mencetak isi Array

Untuk mencetak isi dari array, kita perlu melakukan looping kembali di setiap indeks pada array.

```

output:
subi $t0,$t0,4
li $v0,1                                #print integer
lw $a0,0($t0)
syscall

subi $t2,$t2,1                          #menambahkan counter
bne $t2,0,output                       #looping

exit:
li $v0,10                              # exit command
syscall                                # execute

```

Perlu diperhatikan bahwa pointer t0 sekarang ada di indeks terakhir yang diakibatkan dari looping input sebelumnya.

Alur Cetak:

Loop	Pointer Array	Hasil Cetak
ke-0	[5,4,3,2, <b>1</b> ]	1
ke-1	[5,4,3, <b>2</b> ,1]	12
ke-2	[5,4, <b>3</b> ,2,1]	123
ke-3	[5, <b>4</b> ,3,2,1]	1234
ke-4	[ <b>5</b> ,4,3,2,1]	12345

- Hasil input-output:

```

Input :56789
56789
-- program is finished running --

```

Masih banyak sekali fungsi yang bisa digunakan untuk memanipulasi array pada MIPS. Hal ini dapat dilihat lebih lengkap di MIPS\_Green\_Sheet.pdf.