# MODUL MIPS: Lab 4

## Advanced Array Manipulation

Lab minggu ini berfokus kepada pemanipulasian array yang lebih canggih menggunakan **branching** dan **jumping** untuk melakukan **looping**. Lab minggu ini dapat diselesaikan murni dengan **branching, jumping, read/write** dan **aritmetika.** Silahkan membaca ulang modul lab 1 & 2 karena masih sangat berguna untuk lab minggu ini.

### Looping Example

Program ini akan memprint '12345678910'.

```
1   .text
2   .globl main
3
4           li $t0, 0                       # initialize $t0
5
6   loop:
7
8           beq $t0, 10, exit               # if t0 == 10, goto exit
9
10          addi $t0, $t0, 1                # increment $t0
11
12          li $v0, 1                       # print $t0
13          la $a0, ($t0)
14          syscall
15
16          j loop                          # goto loop
17
18  exit:
19          li $v0, 10
20          syscall
21
22
```

### Branch Instructions

Branch instructions are used for, well, branching. Do note though that branching instruction's jump range is smaller than jump instructions.

| Instruction | Syntax | Operation |
|---|---|---|
| beq | $s, $t, label | Go to label if $s == $t |

| | | |
|---|---|---|
| bgtz | $s, label | Go to label if $s > 0 |
| blez | $s, label | Go to label if $s <= 0 |
| bne | $s, $t, label | Go to label if $s != $t |

**Jump Instructions**

Jump instructions are used for an immediate jump to part of the program without a condition.

| Instruction | Syntax | Operation |
|---|---|---|
| j | label | Jump to label |
| jal | label | Jump to label, $31 (return address) = pc |
| jr | $ra | Jump to address ra stored (pc = ra) |

Source: **MIPS_Green_Sheet.pdf**