

# Tugas Mandiri 2

Deadline: **24** Maret 2023, 17.00 Waktu SCELE

Pengantar Organisasi Komputer  
Semester Genap 2022/2023

Revisi 0

---

Petunjuk pengerjaan:

1. Jawaban tugas **HARUS** ditulis tangan atau menggunakan pen tablet. Untuk penulisan tangan dapat ditulis di kertas polos A4 atau kertas folio bergaris dan discan. **Tidak boleh diketik!** Kumpulkan semua halaman jawaban ke satu file yang berformat PDF.
2. Tuliskan **nama, NPM, kelas, dan kode asisten** di **setiap lembar** berkas jawaban Anda. Kelalaian menuliskan keempat informasi ini (lupa atau salah tulis) akan **diberikan penalti -5 poin/kesalahan**.
3. Pastikan tulisan tangan rapi dapat dibaca!
4. Keterlambatan **10 menit < x < 2 jam** dari deadline akan dikenakan penalti **sebesar 30 poin** dari nilai tugas. Keterlambatan **2-6 jam** dari deadline dikenakan penalti **sebesar 60 poin** dari nilai tugas. Pengumpulan yang melewati batasan waktu tersebut **tidak akan dinilai**.
5. Silahkan tambahkan asumsi sendiri bila diperlukan.
6. **Plagiarisme adalah pelanggaran serius dengan sanksi nilai 0.**
7. **Warna biru** menunjukan revisi, silahkan tulis pada revisi mana anda mengerjakan, jika tidak menulis maka akan dianggap revisi terbaru.
8. Format penamaan:

**Tugas2\_KodeAsdos\_NPM\_Nama.pdf**

Contoh: **Tugas2\_RT\_2206110311\_Suzumelwato.pdf**

Revisi 0

---

1. **[15]** Peokra, yang sedang mempelajari MIPS Instruction Set Architecture, merasa kesulitan dalam memahami format-format instruksi yang terdapat di MIPS ISA, yakni R-Format, I-Format, dan J-Format. Sebagai seorang teman yang baik, Peokra meminta bantuan kepadamu untuk menjelaskan beberapa hal mengenai perbedaan dari ketiga format instruksi tersebut. Apakah perbedaan dari ketiga format instruksi tersebut? Bagaimana ciri umum mereka terhadap alur Datapath (termasuk control unit)? (misalkan, format tertentu akan menyebabkan nilai 0 atau 1 pada bagian tertentu)

2. **[4+4+4+4+4]** Peokra sedang membuat suatu program menggunakan MIPS assembly. Namun, terdapat beberapa instruksi dalam programnya yang diragukan kebenarannya. Periksa apakah instruksi-instruksi di bawah ini dapat dijalankan atau tidak! Jika iya, sebutkan fungsi/kegunaan dari instruksi tersebut beserta register yang terdampak (isinya berubah)! Jika tidak, cukup berikan alasan mengapa instruksi tersebut tidak dapat dijalankan!

- a. `addi $9, $8, 0xAAAAA`
- b. `jal myLabel`
- c. `sw $8, $10($a0)`
- d. `sub $v2, $a0, $a1`
- e. `lw $a0, -4($10)`

3. **[4+4+4+4+4]** Peokra menemukan beberapa bilangan yang merepresentasikan instruksi MIPS. Tentukanlah tipe format serta instruksi MIPS apa yang direpresentasikan oleh bilangan-bilangan dibawah ini! Tuliskan cara mendapatkan instruksi tersebut! Sebutkan ciri khusus yang dapat digunakan untuk mengidentifikasi/membedakan masing-masing instruksi tersebut!

- a.  $(0001\ 0101\ 0100\ 1001\ 1111\ 1111\ 1111\ 1011)_2$
- b.  $(02082020)_{16}$
- c.  $(1515776412)_8$
- d.  $(600256)_{10}$
- e.  $(ACABAAA9)_{16}$

**Reminder:** Jika bilangan di atas tidak berukuran 32 bit, extend terlebih dahulu MSB-nya dengan bit 0 hingga memenuhi 32 bit.

4. **[20]** Setelah mempelajari MIPS ISA, Peokra ingin mencoba untuk membuat suatu program untuk membantunya dalam pekerjaannya. Peokra ditugaskan untuk membuat suatu program yang dapat menghitung suku-suku yang terdapat dalam barisan  $(3, 9, 27, 81, 243, \dots, 3^{n-1}, 3^n)$  hingga suku ke- $n$ . Ia pun meminta bantuan kepadamu untuk membuat suatu program yang dapat melakukan hal itu, dengan spesifikasi sebagai berikut:

- Nilai  $n$  diketahui sudah tersimpan di dalam memori.
- Dijamin nilai  $n$  merupakan integer dalam interval  $1 \leq n \leq 18$
- Suku-suku dari barisan tersebut perlu disimpan pada memori pada alamat yang berurutan (seperti misalkan `0x10011200`, `0x10011204`, `0x10011208`, dst).
- Pada awalnya, register `$a0` berisi address memori yang menyimpan nilai  $n$ , dan register `$a1` berisi address memori yang akan digunakan untuk menyimpan suku pertama barisan tersebut.

Buatlah program yang memenuhi permintaan Peokra di atas menggunakan MIPS assembly!

5. **[5+5+5]** Setelah memahami MIPS ISA (berkat bantuan kamu!), sekarang saatnya Peokra mempelajari tentang Datapath dan Control. Namun, Peokra kembali mengalami kebingungan untuk menentukan alur Datapath dan Control saat suatu instruksi MIPS dijalankan. Oleh karena itu, Peokra meminta bantuan kalian untuk menggambarkan jalur yang dilewati datapath beserta control saat menjalankan instruksi-instruksi berikut:

- a. add \$16, \$10, \$12
- b. sw \$9, 8(\$19)
- c. beq \$18, \$14, someLoop (asumsikan nilai \$18 == nilai \$14 sehingga branching akan terjadi)

**Ketentuan menjawab:**

- Tandai alur datapath dengan warna merah.
- Tandai alur control signal yang pasti memiliki nilai 1 saat menjalankan suatu instruksi dengan warna biru. Khusus untuk control signal ALUOp, tandai control signal tersebut apabila memiliki nilai selain 0.

**Note:**

Untuk mengerjakan soal ini, gunakanlah gambar datapath yang disediakan pada link [MIPS Datapath \(r0\).png](#). Gambar dapat diperbanyak secara digital atau dicetak fisik (print). Untuk soal ini, diperbolehkan menjawab dengan mengedit gambar menggunakan aplikasi digital (seperti misalnya *Paint*, *GIMP*, *Photoshop*, dll.). Terdapat 3 buah gambar datapath yang harus dikumpulkan (satu untuk masing-masing instruksi).

6. **[10]** Isilah tabel berikut sesuai dengan kode assembly nomor 5a berdasarkan nilai yang dilewati pada datapath.

<b>RegDst</b>	
<b>RegWrite</b>	
<b>WriteReg (binary, 5 bit)</b>	
<b>ReadReg1 (binary, 5 bit)</b>	
<b>ReadReg2 (binary, 5 bit)</b>	
<b>ReadData1 (hex, 32 bit)</b>	
<b>ReadData2 (hex, 32 bit)</b>	
<b>ALUSrc</b>	
<b>ALU Input 2 (hex, 32 bit)</b>	
<b>ALU Control (binary, 4 bit)</b>	
<b>ALU Result (hex, 32 bit)</b>	
<b>Zero</b>	
<b>MemWrite Data (hex, 32 bit)</b>	
<b>MemWrite</b>	
<b>MemRead</b>	
<b>MemToReg</b>	

**Note:** dalam mengisi tabel di atas, gunakan nilai-nilai dari tabel ini sebagai referensi:

<b>Registers in MIPS</b>	<b>Register initial values</b>	<b>ALU Control (4 bit binary)</b>
R2-R3 = \$v0-\$v1 R4-R7 = \$a0-\$a3 R8-R15 = \$t0-\$t7	R10 = 0x 0304 0CA0 R11 = 0x 0000 0B16 R12 = 0x 0020 1D20	AND = 0000 OR = 0001 ADD = 0010

R16-R23 = \$s0-\$s7	R16 = 0x 0043 C27A R17 = 0x 0002 0A11	SUB = 0110 SLT = 0111 NOR = 1100
---------------------	--	--