



**FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

LAB SHEET PRAKTIK SISTEM MIKROPROSESOR

SEM. 3

INSTRUKSI JUMP

JOB 4

4x50"

EKA5219

Revisi:01

30 Agu 2017

Hal 1 / 10

A. TUJUAN

1. Mahasiswa tahu jenis-jenis JUMP
2. Mahasiswa dapat menggunakan instruksi JUMP

B. ALAT

Trainer MPF – 1

C. DASAR TEORI

Dalam mikroprosesor Zilog Z – 80 CPU instruksi-instruksi pencabangan menggunakan instruksi JUMP. Instruksi JUMP membuat mikroprosesor menjadi perangkat yang sangat ampuh. Instruksi JUMP dapat dikategorikan menjadi empat kategori yaitu:

→ JUMP Bersyarat

JUMP bersyarat adalah jenis instruksi JUMP yang bekerja melakukan lompatan atau kontinyu/tidak melompat berdasarkan syarat yang diberikan. Mnemonic untuk lompatan bersyarat ada tiga yaitu:

➤ JP CC

Lompatan absolute bersyarat adalah lompatan yang langsung menuju alamat sasaran dengan data alamat 16 bit.

➤ JR CC

Lompatan relatif bersyarat adalah lompatan yang penunjukan alamatnya bernilai relative terhadap alamat posisi saat melompat.

➤ DJNZ

Lompatan relatif khusus terhadap register B adalah lompatan yang penunjukan alamatnya bernilai relative terhadap alamat posisi saat melompat.

Bentuk perintah JUMP bersyarat adalah sebagai berikut :

➤ Lompatan absolute bersyarat :JP cc, nn :

Jika kondisi syarat cc terpenuhi maka PC ← nn

Jika syarat cc tidak terpenuhi maka kontinyu.

➤ Lompatan relative bersyarat: JR cc, nn :

Jika kondisi syarat cc terpenuhi maka PC ← PC + e

Jika syarat cc tidak terpenuhi maka kontinyu.

➤ Lompatan Relatif bersyarat khusus: DJNZ :

B ← B – 1

Jika B = kontinyu dan jika B ≠ 0 maka PC ← PC + e



**FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

LAB SHEET PRAKTIK SISTEM MIKROPROSESOR

SEM. 3

INSTRUKSI JUMP

JOB 4

4x50"

EKA5219

Revisi:01

30 Agu 2017

Hal 2 / 10

Syarat yang dimaksud untuk setiap perintah JUMP terkait dengan kondisi bit status flag dari satu step perintah sebelumnya. Ada 8 kemungkinan syarat yang dapat diberikan terkait dengan bit status flag. Kedelapan syarat itu dalam mikroprosesor dicatat dalam sebuah register yang disebut register flag. Untuk mikroprosesor Z – 80 CPU susunan status dari register flag sebagai berikut:

| | | | | | | | |
|---|---|---|---|---|-----|---|---|
| S | Z | X | H | X | P/V | N | C |
|---|---|---|---|---|-----|---|---|

Makna masing-masing bit dari tanda status flag pada register F adalah sebagai berikut :

| | | |
|----------------------|-----|--|
| S (Sign) | = 1 | Menunjukkan hasil operasi aritmetika/logika sebelumnya bertanda negative (b7=1) |
| | = 0 | Menunjukkan hasil operasi aritmetika/logika sebelumnya bertanda positif (b7=0) |
| Z (Zero) | = 1 | Menunjukkan hasil operasi aritmetika/logika sebelumnya bernilai nol. |
| | = 0 | Menunjukkan hasil operasi aritmetika/logika sebelumnya bernilai tidak nol. |
| H (Half-Carry) | = 1 | Jika ada Carry dari bit B3 ke bit B4 |
| | = 0 | Jika tidak ada Carry dari bit B3 ke bit B4 |
| P/V(Parity Overflow) | = 1 | Jika hasil operasi aritmetika/logika sebelumnya menunjukkan paritas genap atau terjadi Overflow |
| | = 0 | Jika hasil operasi aritmetika/logika sebelumnya menunjukkan paritas ganjil atau tidak terjadi Overflow |
| N (Non Carry) | = 1 | Operasi sebelumnya adalah operasi SUBTRACK/Pengurangan |
| | = 0 | Operasi sebelumnya adalah operasi bukan SUBTRACK/Pengurangan |
| C (Carry) | = 1 | Jika operasi sebelumnya menghasilkan Carry atau Borrow |
| | = 0 | Jika operasi sebelumnya tidak menghasilkan Carry atau Borrow |
| X | = | tidak digunakan |



FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA

LAB SHEET PRAKTIK SISTEM MIKROPROSESOR

SEM. 3

INSTRUKSI JUMP

JOB 4

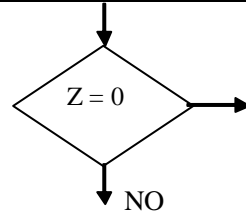
4x50"

EKA5219

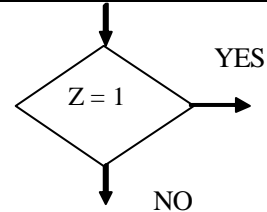
Revisi:01

30 Agu 2017

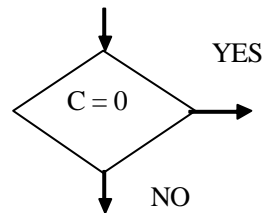
Hal 3 / 10



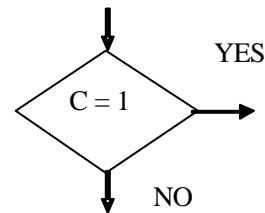
JPNZ / JRNZ / DJNZ



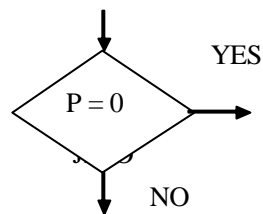
JPZ / JRZ



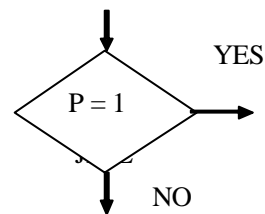
JPNC / JRNC



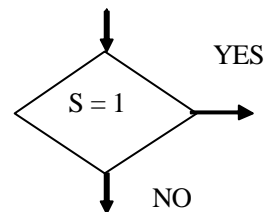
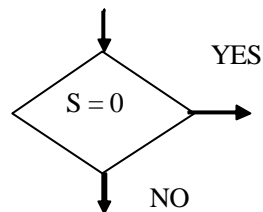
JPC / JRC



JPP



JPM



→ JUMP Tanpa Syarat

Jump tanpa syarat adalah jenis instruksi jump yang bekerja melakukan lompatan atau kontinyu tanpa adanya syarat yang diberikan. Mnemonik untuk lompatan tanpa syarat ada dua yaitu :



**FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

LAB SHEET PRAKTIK SISTEM MIKROPROSESOR

SEM. 3

INSTRUKSI JUMP

JOB 4

4x50"

EKA5219

Revisi:01

30 Agu 2017

Hal 4 / 10

1. JP : Lompatan absolut tanpa syarat adalah lompatan yang langsung menunjuk alamat sasaran dengan data alamat 16 bit.
2. JR : Lompatan Relatif tanpa syarat adalah lompatan yang penunjukan alamatnya bernilai relatif terhadap alamat posisi saat melompat.
 - Lompatan Absolut tanpa syarat : JP nn : PC ← nn
Instruksi ini memasukkan alamat memori nn ke register PC (Program Counter), sehingga mikroprocessor akan menjalankan instruksi yang ada pada lokasi alamat nn.
 - Lompatan Relatif tanpa syarat : JR e : PC ← PC + e
e adalah bilangan bertanda yang bernilai positif jika melompat maju ke belakang ke alamat sebelumnya.

Nilai relatif lompatan dapat dihitung dengan rumus :

1. Jika lompatannya maju ke alamat atasnya :

$$e = d - (S + 02)$$

2. Jika lompatannya mundur ke alamat sebelumnya :

$$e = (S + 02) - d \text{ lalu dikomplemen duakan}$$

dimana d = alamatnya tujuan dan S = alamat asal perintah Jump.

Disamping Jump bersyarat masih ada tiga jenis Jump lainnya yaitu Jump Absolut berbasis register HL, IX, dan IY dengan mnemonic:

JP (HL) : PC ← HL

JP (IX) : PC ← IX

JP (IY) : PC ← IY

D. LANGKAH KERJA

- 1.a. Buatlah Algoritma Program, Flow Chart dan Program untuk mengosongkan 100 lokasi memori mulai alamat 1900.
- 1.b. Buatlah Algoritma Program, Flow Chart dan Program untuk membangkitkan 100 data bilangan desimal mulai dari 0 di memori mulai alamat 1900.
- 1.c. Buatlah Algoritma Program, Flow Chart dan Program untuk membangkitkan 50 data bilangan desimal ganjil mulai dari 1 di memori mulai alamat 1900.
- 1.d. Buatlah Algoritma Program, Flow Chart dan Program untuk membangkitkan 50 data bilangan desimal genap mulai dari 2 di memori mulai alamat 1900.

E. DATA

1. Kasus I

Program mengosongkan 100 lokasi memori dari alamat 1900 sampai dengan 1963.

2. Algoritma

1. Buat cacahan jumlah data sama dengan 100 (64h) di Register C
2. Set alamat awal tempat simpan data di 1900h di Register IX
3. Set data awal sama dengan 0 di Register A



**FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

LAB SHEET PRAKTIK SISTEM MIKROPROSESOR

SEM. 3

INSTRUKSI JUMP

JOB 4

4x50"

EKA5219

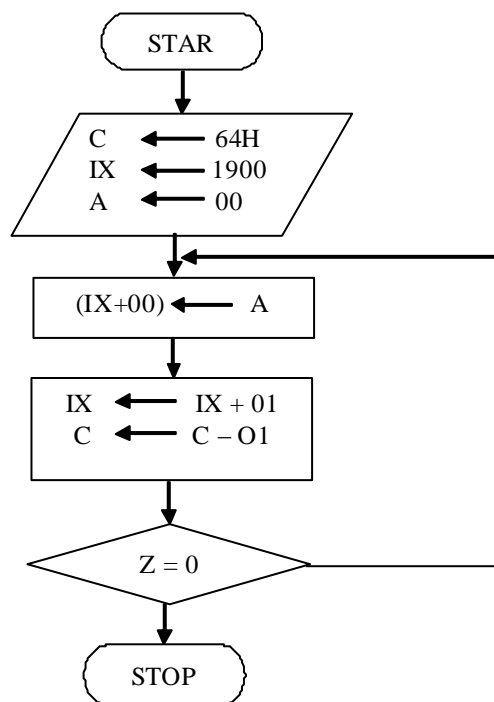
Revisi:01

30 Agu 2017

Hal 5 / 10

4. Simpan data di Reg. A ke memori tempat simpan data Reg. IX
5. Naikkan nilai pencatat alamat Reg. IX satu alamat
6. Kurangi data cacahan Reg. C dengan 1
7. Lihat apakah nilai cacahan Reg C sama dg 0, jika tidak kembali ke langkah 4
8. Berhenti.

3. Flow Chart



4. Program

| Address | Kode Operasi | No. | Label | Assembly | Keterangan |
|---------|--------------|-----|-------|---------------|------------|
| 1800 | 0E 64 | 1 | | Ld C, 64H | |
| 1802 | DD 21 00 19 | 2 | | LD IX, 1900 | |
| 1806 | 3E 00 | 3 | | LD A, 00 | |
| 1808 | DD 77 00 | 4 | Catat | LD (IX+00), A | |
| 180B | DD 23 | 5 | | INC IX | |
| 180D | 0D | 6 | | DEC C | |
| 180E | C2 08 18 | 7 | | JPNZ, Catat | |



**FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

LAB SHEET PRAKTIK SISTEM MIKROPROSESOR

SEM. 3

INSTRUKSI JUMP

JOB 4

4x50"

EKA5219

Revisi:01

30 Agu 2017

Hal 6 / 10

1811

FF

8

RST 38

Setelah program diatas dimasukkan dan dieksekusi, memori alamat 1900 sampai alamat 1963 berisi 00 (kosong).

1. Kasus II

Membangkitkan 100 data desimal dari 00 sampai dengan 99 di memori alamat 1900 – 1963.

2. Algoritma

1. Buat cacahan jumlah data sama dengan 100 (64H) di register C.
2. Set alamat awal tempat simpan data di 1900h di register IX.
3. Set data awal sama dengan 0 di register A.
4. Simpan data di register A ke memori tempat simpan data register IX.
5. Jumlahkan data yang ada pada register A dengan 1.
6. Desimalkan
7. Naikkan nilai pencacah alamat register IX satu alamat.
8. Kurangi data cacahan register C dengan 1.
9. Lihatlah apakah nilai cacahan register C sama dengan 0, jika tidak kembali ke langkah 4.
10. Berhenti.

3. Program

| Address | Kode Operasi | No. | Label | Assembly | Keterangan |
|---------|--------------|-----|-------|---------------|------------|
| 1800 | 0E 64 | 1 | | LD C, 64H | |
| 1802 | DD 21 00 19 | 2 | | LD IX, 1900H | |
| 1806 | 3E 00 | 3 | | LD A, 00H | |
| 1808 | DD 77 00 | 4 | Catat | LD (IX+00), A | |
| 180B | C6 01 | 5 | | ADD A, 01 | |
| 180D | 27 | 6 | | DAA | |
| 180E | DD 23 | 7 | | INC IX | |
| 1810 | 0D | 8 | | DEC C | |
| 1811 | C2 08 18 | 9 | | JPNZ, Catat | |
| 1814 | FF | 10 | | RST 38 | |

Setelah program dimasukkan dan dieksekusi, memori alamat 1900 sampai dengan 1963 berisi 00 sampai dengan 99.



**FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

LAB SHEET PRAKTIK SISTEM MIKROPROSESOR

SEM. 3

INSTRUKSI JUMP

JOB 4

4x50"

EKA5219

Revisi:01

30 Agu 2017

Hal 7 / 10

1. Kasus III

Program untuk membangkitkan 50 data bilangan desimal ganjil mulai dari 1 di memori mulai di alamat 1990.

2. Algoritma

1. Buat cacahan jumlah data sama dengan 50 (32h) di register C
2. Set alamat awal tempat simpan data di 1990h di register IX
3. Set data awal sama dengan 1 di register A
4. Simpan data di register A ke memori tempat simpan data register IX
5. Jumlahkan data yang ada di register A dengan 2
6. Naikkan nilai pencatat alamat register IX satu alamat
7. Kurangi data cacahan register C dengan 1
8. Lihat apakah nilai cacahan register C sama dengan 0, jika tidak kembali ke langkah 4
9. Berhenti

3. Flow Chart



**FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

LAB SHEET PRAKTIK SISTEM MIKROPROSESOR

SEM. 3

INSTRUKSI JUMP

JOB 4

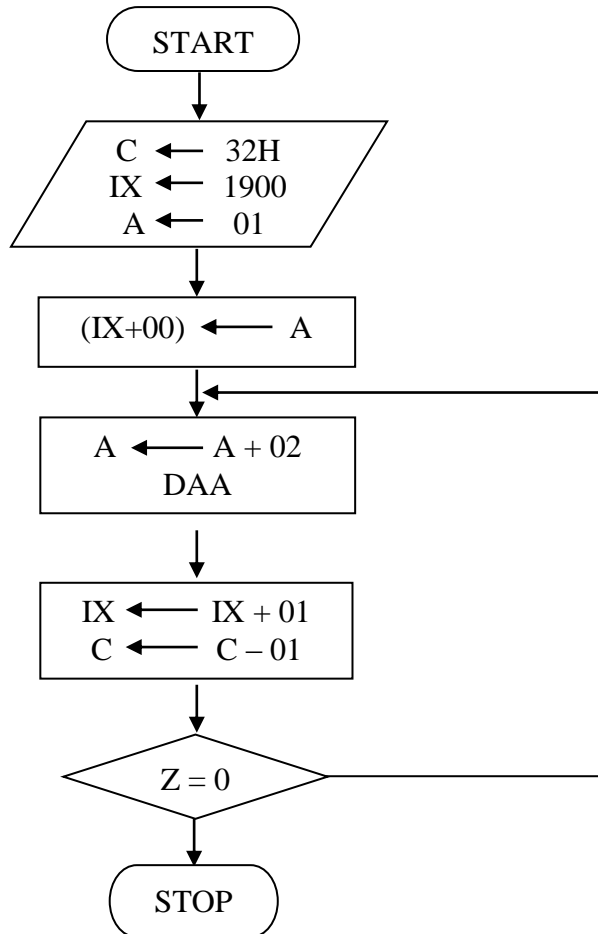
4x50"

EKA5219

Revisi:01

30 Agu 2017

Hal 8 / 10



4. Program

| Address | Kode Operasi | No. | Label | Assembly | Keterangan |
|---------|--------------|-----|-------|---------------|------------|
| 1800 | 0E 32 | 1 | | LD C, 32H | |
| 1802 | DD 21 00 19 | 2 | | LD IX, 1900 | |
| 1806 | 3E 01 | 3 | | LD A, 01 | |
| 1808 | DD 77 00 | 4 | Catat | LD (IX+00), A | |
| 180b | C6 02 | 5 | | ADD A, 02 | |
| 180d | 27 | 6 | | DAA | |
| 180e | DD 23 | 7 | | INC IX | |
| 1810 | 0D | 8 | | DEC C | |



**FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

LAB SHEET PRAKTIK SISTEM MIKROPROSESOR

SEM. 3

INSTRUKSI JUMP

JOB 4

4x50"

EKA5219

Revisi:01

30 Agu 2017

Hal 9 / 10

| | | | | | |
|------|----------|----|--|-------------|--|
| 1811 | C2 08 18 | 9 | | JPNZ, Catat | |
| 1814 | FF | 10 | | RST 38 | |

Setelah program dimasukkan dan dieksekusi memori alamat 1900 sampai dengan 1931 akan berisi bilangan ganjil mulai dari 01 sampai dengan 99.

1. Kasus IV

Program untuk membangkitkan 50 data bilangan desimal genap mulai dari 2 di memori mulai dari alamat 1900.

2. Algoritma

1. Buat cacahan jumlah data sama dengan 50 (32h) di register C
2. Set alamat awal tempat simpan data di 1900 di register IX
3. Set data awal sama dengan 2 di register A
4. Simpan data di register A ke memori tempat simpan data register IX
5. Jumlahkan data yang ada pada register A dengan 2
6. Naikkan nilai pencatat alamat register IX satu alamat
7. Kurangi data cacahan register C dengan satu
8. Lihat apakah nilai cacahan register C sama dengan 0, jika tidak kembali ke langkah 4
9. Berhenti

3. Flow Chart



**FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

LAB SHEET PRAKTIK SISTEM MIKROPROSESOR

SEM. 3

INSTRUKSI JUMP

JOB 4

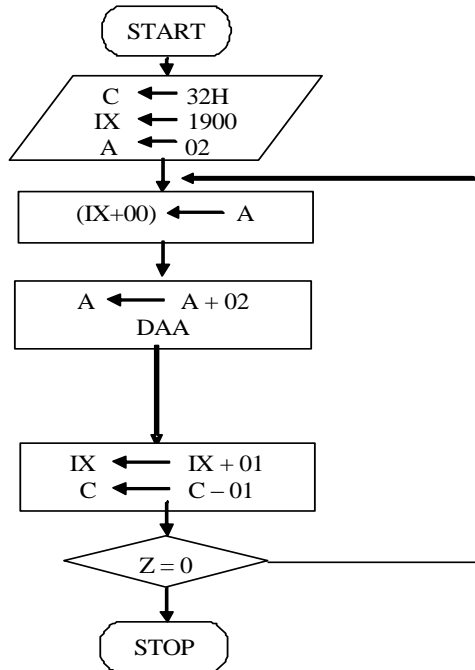
4x50"

EKA5219

Revisi:01

30 Agu 2017

Hal 10 / 10



4. Program

| Address | Kode Operasi | No. | Label | Assembly | Keterangan |
|---------|--------------|-----|-------|---------------|------------|
| 1800 | 0E 32 | 1 | | LD C, 32H | |
| 1802 | DD 21 00 19 | 2 | | LD IX, 1900 | |
| 1806 | 3E 02 | 3 | | LD A, 02 | |
| 1808 | DD 77 00 | 4 | Catat | LD (IX+00), A | |
| 180B | C6 02 | 5 | | ADD A, 02 | |
| 180D | 27 | 6 | | DAA | |
| 180E | DD 23 | 7 | | INC IX | |
| 1810 | 0D | 8 | | DEC C | |
| 1811 | C2 08 18 | 9 | | JPNZ, Catat | |
| 1814 | FF | 10 | | RST 38 | |

Setelah program dimasukkan dan dieksekusi, maka memori alamat 1900 sampai dengan 1931 akan berisi bilangan ganjil mulai dari 02 sampai dengan 98.