

Lab 5 Clement Samuel Marly 2206082114

1. Fungsi dari stack pointer dalam program:
Mendapatkan data dari tabel dan mengumpulkannya dalam sebuah *stack* yang kemudian bisa digunakan dengan menge-*pop* atau mengeluarkan satu nilai dari *stack*.
2. Fungsi dari pointer X, Y dan Z yang digunakan pada program.
Pointer X berfungsi sebagai tempat awal menaruh data di memory dalam tabel B dengan angka awal 8
Pointer Y berfungsi sebagai tempat awal menaruh data di memory dalam tabel A dengan angka awal 0
Pointer Z berfungsi untuk membaca dan memasukkan data dari tabel ke dalam stack pointer.
3. Alur program:

```
.include "m8515def.inc"
```

```
.equ    BLOCK1      = $72          ;start address of SRAM array #1
.equ    BLOCK2      = $DC          ;start address of SRAM array #2
.def    temp        = r16          ;temporary storage variable
.def    temp2       = r17          ;temporary storage variable (only if needed)
.def    size        = r18          ;size of block to be copied
```

START:

```
ldi     temp,low(RAMEND)
out     SPL,temp                ;init Stack Pointer
ldi     temp,high(RAMEND)
out     SPH,temp
```

INIT_BLOCK1:

```
ldi     ZH,high(TABLE*2)
ldi     ZL,low(TABLE*2)        ;init Z-pointer
ldi     YH,high(BLOCK1)
ldi     YL,low(BLOCK1)        ;init Y-pointer
ldi     size,8                 ;init total loop at FUNGSI 1
rcall   FUNGSIA
```

INIT_BLOCK2:

```
ldi     ZH,high(TABLE*2)
ldi     ZL,low(TABLE*2)        ;init Z-pointer
ldi     XH,high(BLOCK2)
ldi     XL,low(BLOCK2)        ;init X-pointer
ldi     size,8
rcall   FUNGSIB ;
```

FOREVER:

rjmp FOREVER

TABLE:

.db	2,3	;start of table (6 bytes)
.db	5,4	
.db	6,7	
.db	9,8	

FUNGSI:

lpm temp,Z+	;Ambil nilai dari tabel
push temp	;Input nilai ke Stack Pointer
dec size	;Kurangi size / total loop ke 0
brne FUNGSI	;Loop sampai semua nilai selesai dipindah
rjmp TABEL_A	

TABEL_A:

pop temp	;Ambil nilai dari Stack Pointer
rcall SPLIT_A	;Pisah bagian tabel menjadi 2
rcall SETA	;Set kondisi untuk add value ke dalam tabel
inc size	;Tambah jumlah iterasi
cpi size,8	;Cek iterasi
brne TABEL_A	;Loop iterasi sampai iterasi bernilai 8
breq INIT_BLOCK2	;Saat iterasi bernilai 8, lanjut ke tabel B

SPLIT_A:

cpi size,4	;Cek iterasi
breq SETINPUT	;iterasi 4
brlo LEFTA	;iterasi < 4 (pisah secara kolom)
brpl RIGHTA	;iterasi > 4

SETINPUT:

ldi temp2,1	;insert 1 dalam temp2 (0 dan 1 menjadi penanda isi value atau isi
dengan 0)	
ret	;return ke TABEL_A

LEFTA:

cpi size,0	;Cek iterasi
breq SETINPUT	;iterasi 0
cpi size,3	;Cek iterasi
breq SETINPUT	;iterasi 3
ldi temp2,0	;insert 0 dalam temp2 selain iterasi 1 dan 3
ret	;return ke TABEL_A

RIGHTA:

cpi size,7	;Cek iterasi
breq SETINPUT	;iterasi 7
ldi temp2,0	;insert 0 dalam temp2 selain iterasi 7
ret	;return ke TABEL_A

SETA:

cpi temp2,1	;Cek temp2
-------------	------------

breq SETVALUEA	;temp 2 = 1
brne SET0A	;temp 2 = 0
SET0A:	
ldi temp2,0	;set temp2 = 0
st Y+,temp2	;simpan data sesuai pointer Y dan increment Y
ret	;return ke TABEL_A
SETVALUEA:	
st Y+,temp	;simpan data sesuai pointer Y dan increment Y
ret	;return ke TABEL_A
FUNGSIB:	
lpm temp,Z+	;Ambil nilai dari tabel
push temp	;Input nilai ke Stack Pointer
dec size	;Kurangi size / total loop ke 0
brne FUNGSIB	;Loop sampai semua nilai selesai dipindah
adiw X,8	;Set pointer X ke 8
rjmp TABEL_B	
TABEL_B:	
pop temp	;Ambil nilai dari Stack Pointer
pop temp2	;Ambil nilai dari Stack Pointer
add temp,temp2	;Tambah kedua nilai
rcall SPLIT_B	;Pisah bagian B menjadi 4
inc size	;Tambah jumlah iterasi
cpi size,4	;Cek iterasi
brne TABEL_B	;Loop iterasi sampai iterasi bernilai 4
breq FOREVER	;Saat iterasi bernilai 4, end program
SPLIT_B:	
cpi size,0	;Cek iterasi
breq RIGHTB	;iterasi = 0 (right berarti variabel di kanan dalam 1 baris tabel)
cpi size,1	;Cek iterasi
breq LEFTB	;iterasi = 1 (left berarti variabel di kiri dalam 1 baris tabel)
cpi size,2	;Cek iterasi
breq RIGHTB	;iterasi = 2 (right berarti variabel di kanan dalam 1 baris tabel)
cpi size,3	;Cek iterasi
breq LEFTB	;iterasi = 3 (left berarti variabel di kiri dalam 1 baris tabel)
RIGHTB:	
rcall SET0B	;Set 0 di kiri
rcall SETVALUEB	;Set value di kanan
ret	;return ke TABEL_B
LEFTB:	
rcall SETVALUEB	;Set value di kiri
rcall SET0B	;Set 0 di kanan
ret	;return ke TABEL_B
SET0B:	

```
ldi temp2,0  
st -X,temp2  
ret
```

```
;set temp 2 = 0  
;simpan data sesuai pointer X dan decrement X  
;return ke TABEL_B
```

SETVALUEB:

```
inc temp  
st -X,temp  
ret
```

```
;tambah nilai temp  
;simpan data sesuai pointer X dan decrement X  
;return ke TABEL_B
```