

# **LAPORAN TUGAS**

## **Implementasi Sistem Sederhana Berbasis 8051**

### **Menggunakan Bahasa Rakitan**

Disusun untuk Memenuhi Tugas Mata Kuliah Teknik Mikroprosesor dan Antarmuka



**Disusun oleh:**

Juliant Raffa

(21120120130127)

**Departemen Teknik Komputer**

**Fakultas Teknik**

**Universitas Diponegoro**

**Semarang**

**2022**

## A. Spesifikasi Kebutuhan Sistem

Dirancang sebuah sistem untuk melakukan monitor terhadap sistem pengisian tandon air. Tandon air yang terletak di atas menara terhubung dengan bak PAM dan sumur yang terletak di bawah. Penampung air di bawah masing-masing memiliki pompa untuk memompa air (pompa bak PAM dan sumur). Karena posisinya yang lebih tinggi dari sumur, bak PAM memiliki prioritas lebih tinggi sebagai sumber pengisian air untuk tandon. Terdapat 5 sensor saklar yang berfungsi sebagai masukan sistem. 3 sensor saklar digunakan untuk mendeteksi keadaan tandon air, bak PAM, dan sumur. 2 sensor saklar lain digunakan sebagai masukan sinyal *interrupt* untuk kerusakan pompa bak PAM atau pompa sumur. Sistem akan dirancang menggunakan *microcontroller* 8051 yang disimulasikan menggunakan MCU 8051 IDE.

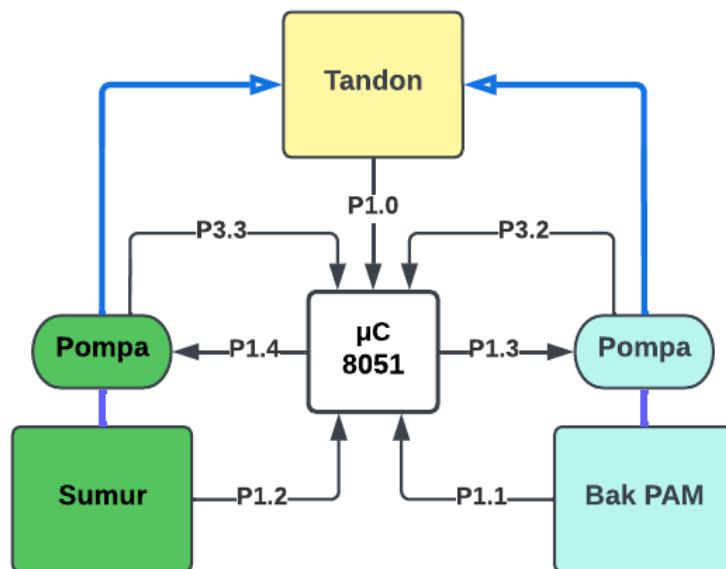
Jika sistem mendeteksi tandon kosong, sistem akan memeriksa jika bak PAM terisi dan pompanya tidak rusak. Jika kedua kondisi terpenuhi, maka pompa bak PAM akan dinyalakan untuk mengisi tandon. Jika pompa bak PAM tidak dapat dinyalakan, maka akan diperiksa kondisi isi sumur dan keadaan pompa sumur. Jika sumur terisi dan pompa sumur tidak rusak, maka pompa sumur akan dinyalakan untuk mengisi tandon. Jika bak PAM dan sumur kosong dan tidak ada pompa yang rusak, maka kedua pompa akan dimatikan.

Karena keterbatasan pada simulator MCU 8051 IDE, sensor saklar akan direpresentasikan dengan *switch* yang bersifat *active low* dan pompa akan direpresentasikan dengan LED yang juga bersifat *active low*. Berikut adalah tabel kebenaran sistem.

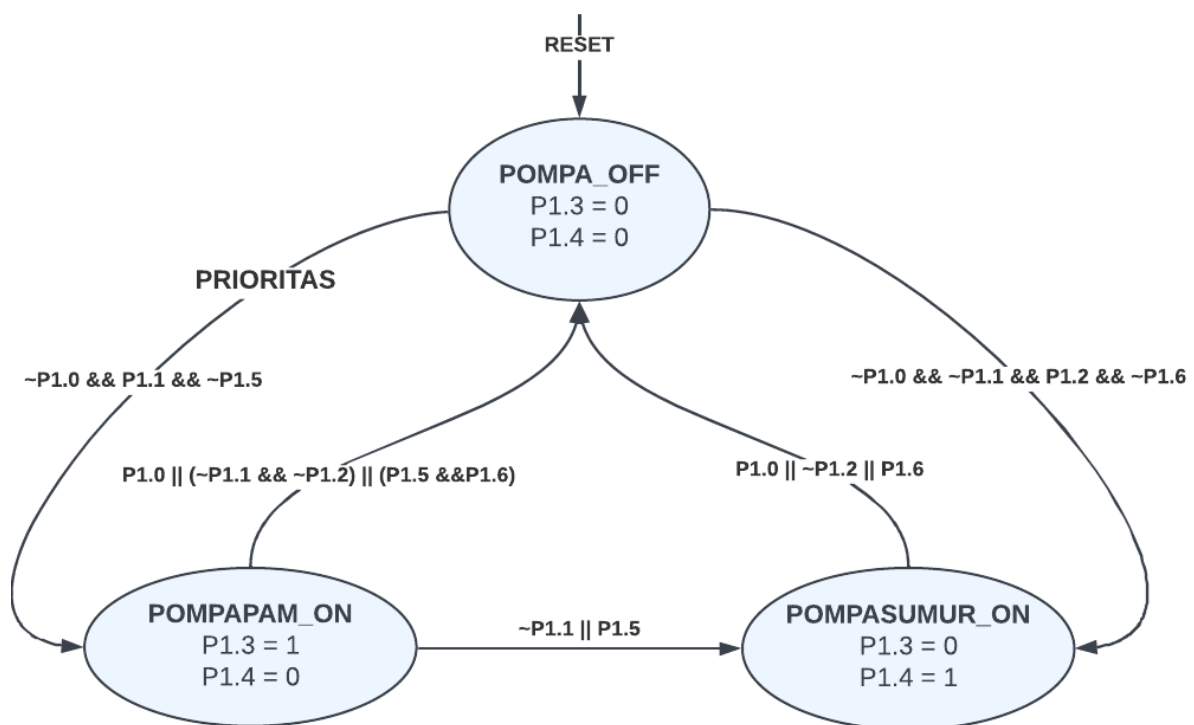
Isi Tandon (P1.0)	Isi Bak PAM (P1.1)	P. Bak PAM Rusak (P1.5)	Isi Sumur (P1.2)	P. Sumur Rusak (P1.6)	Pompa Bak PAM (P1.3)	Pompa Sumur (P1.4)
0	0	X	0	X	0	0
0	0	X	1	0	0	1
0	0	X	X	1	0	0
0	1	0	X	X	1	0
0	X	1	0	X	0	0
0	1	1	1	1	0	0
1	X	X	X	X	0	0

## B. Perancangan Sistem

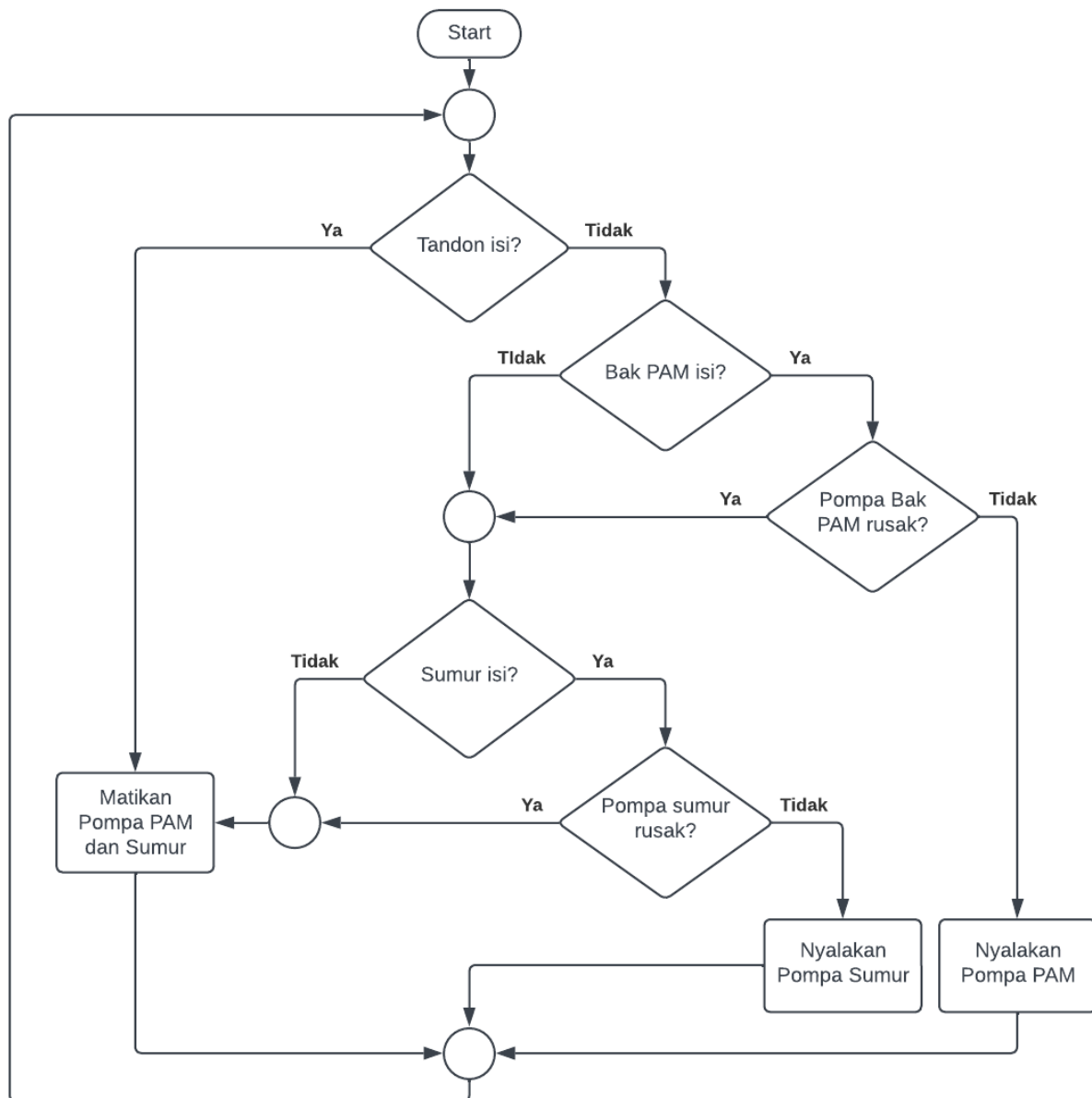
Berikut adalah diagram perancangan sistem yang meliputi diagram blok (HW), diagram FSM (SW) dan diagram alir (*flowchart*). Dilakukan beberapa modifikasi untuk mengimplementasikan pendekatan *interrupt*. **P3.2** dan **P3.3** merupakan pin untuk memasukkan sinyal *interrupt* ketika pompa mengalami kerusakan. ISR yang dijalankan akan melakukan *set* terhadap pin **P1.5** dan **P1.6** yang berfungsi sebagai masukan kondisi pompa.



Gambar 1. Diagram blok



Gambar 2. Diagram FSM



Gambar 1. Diagram alir

### C. Implementasi Kode Rakitan

Sistem diprogram menggunakan bahasa rakitan khusus *microcontroller* 8051 yang disimulasikan menggunakan MCU 8051 IDE. Berikut adalah *source code* serta komentar yang menjelaskan fungsi setiap baris kode.

```

1  $MOD51
2  ; Melakukan penamaan pin dengan instruksi equ
3  ; Pin input
4  TANDON      equ P1.0
5  PAM         equ P1.1
6  SUMUR      equ P1.2
7
8  ; Pin output
9  POMPA_PAM   equ P1.3
10 POMPA_SUMUR equ P1.4
11
12 ; Pin output (HASIL interrupt)
13 PPAM_RUSAK  equ P1.5 ; (P3.2)
14 PSUMUR_RUSAK equ P1.6 ; (P3.3)

```

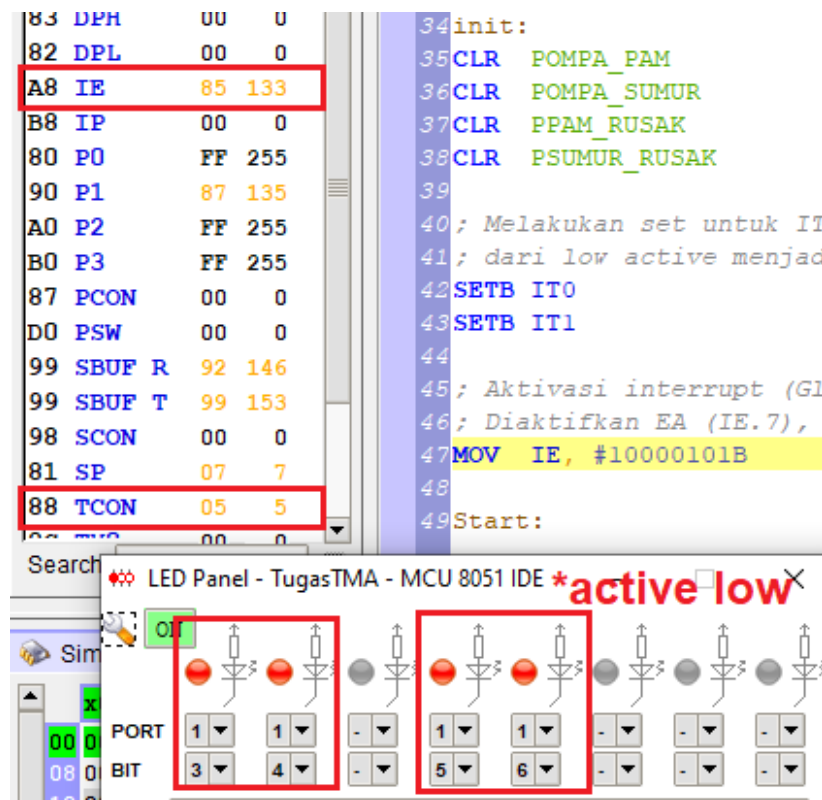
```

15
16
17 ORG 0000h          ; vector untuk reset
18 sjmp init          ; melompat ke init
19
20 ; P3.2
21 ORG 0003h          ; vector INT0 => Interrupt untuk pompa PAM rusak
22 CLR  POMPA_PAM      ; Matikan pompa PAM
23 SETB PPAM_RUSAK     ; Set bit untuk PPAM_RUSAK (1)
24 reti               ; Kembali ke program
25
26 ; P3.3
27 ORG 0013h          ; vector INT1 => Interrupt untuk pompa sumur rusak
28 CLR  POMPA_SUMUR    ; Matikan pompa sumur
29 SETB PSUMUR_RUSAK   ; Set bit untuk PSUMUR_RUSAK (1)
30 reti               ; Kembali ke program
31
32
33 ORG 0030h          ; vector program utama
34 init:              ; Inisialisasi
35 CLR  POMPA_PAM      ; Semua output dinolkan
36 CLR  POMPA_SUMUR
37 CLR  PPAM_RUSAK
38 CLR  PSUMUR_RUSAK
39
40 ; Melakukan set untuk IT0 dan IT1 untuk mengubah mode interrupt
41 ; dari low active menjadi falling-edge active.
42 SETB IT0
43 SETB IT1
44
45 ; Aktivasi interrupt (Global, INT1, INT0)
46 ; Diaktifkan EA (IE.7), EX1 (IE.2), dan EX0 (IE.0)
47 MOV  IE, #10000101B
48
49 Start:              ; Program utama
50 JB   TANDON,POMPA_OFF ; Matikan pompa jika tandon penuh
51 JB   PAM,POMPAPAM_ON  ; Nyalakan pompa PAM jika bak PAM penuh
52 JB   SUMUR,POMPASUMUR_ON; Nyalakan pompa sumur jika sumur penuh
53 AJMP POMPA_OFF        ; Matikan semua pompa
54
55 POMPAPAM_ON:         ; Untuk menyalakan pompa PAM
56 JB   PPAM_RUSAK,POMPASUMUR_ON ; Pengecekkan status pompa rusak
57 CLR  POMPA_SUMUR     ; Mematikan pompa sumur
58 SETB POMPA_PAM       ; Menyalakan pompa PAM
59 AJMP Start           ; Kembali ke Start
60
61 POMPASUMUR_ON:       ; Untuk menyalakan pompa sumur
62 JB   PSUMUR_RUSAK,POMPA_OFF ; Pengecekkan status pompa rusak
63 CLR  POMPA_PAM       ; Mematikan pompa PAM
64 SETB POMPA_SUMUR     ; Menyalakan pompa sumur
65 AJMP Start           ; Kembali ke Start
66
67 POMPA_OFF:           ; Untuk mematikan semua pompa
68 CLR  POMPA_SUMUR     ; Mematikan pompa sumur
69 CLR  POMPA_PAM       ; Mematikan pompa pam
70 AJMP Start           ; Kembali ke Start
71 End

```

## D. Hasil Percobaan dan Analisis

Instruksi pertama yang dijalankan pada *vector* reset dengan alamat 0000h (baris 17) adalah melompat ke *init* (baris 34, alamat 0030h) yang berisi instruksi-instruksi untuk inisialisasi program. Instruksi berupa perintah *clear* untuk pin-pin *output* seperti P1.3 dan P1.4 (pompa), serta P1.5 dan P1.6 (kondisi pompa). Perintah tersebut merubah bit pada pin menjadi 0 yang menyalakan LED (**ACTIVE LOW**). Selanjutnya dilakukan *set* untuk IT0 dan IT1 yang merupakan nilai dari register TCON (*Timer Control*). Setelah di-*set*, nilai **TCON** pada SFR menjadi 05h (00000101b). Dimasukkan juga nilai 85h (10000101b) pada **IE** yang berupa register *interrupt* setelah EA, EX0, dan EX1 diaktifkan.



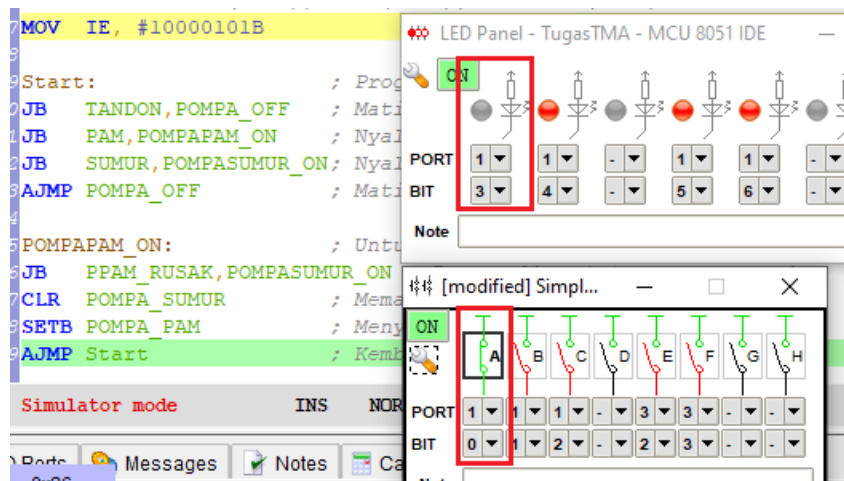
Gambar 4. Kondisi LED dan SFR setelah inisialisasi

Setelah inisialisasi program akan berlanjut ke *Start* yang berisi program utama. Berikut adalah hubungan antara pin dan representasinya.

Input (Switch)		Output (LED)	
P1.0	Isi Tandon	P1.3	Pompa PAM
P1.1	Isi Bak PAM	P1.4	Pompa Sumur
P1.2	Isi Sumur	P1.5	Pompa PAM Rusak
P3.2	<i>Interrupt</i> Pompa PAM Rusak	P1.6	Pompa Sumur Rusak
P3.3	<i>Interrupt</i> Pompa Sumur Rusak		

## 1. Kondisi Pertama: Tandon (0), Bak PAM (1), (Tanpa *interrupt*)

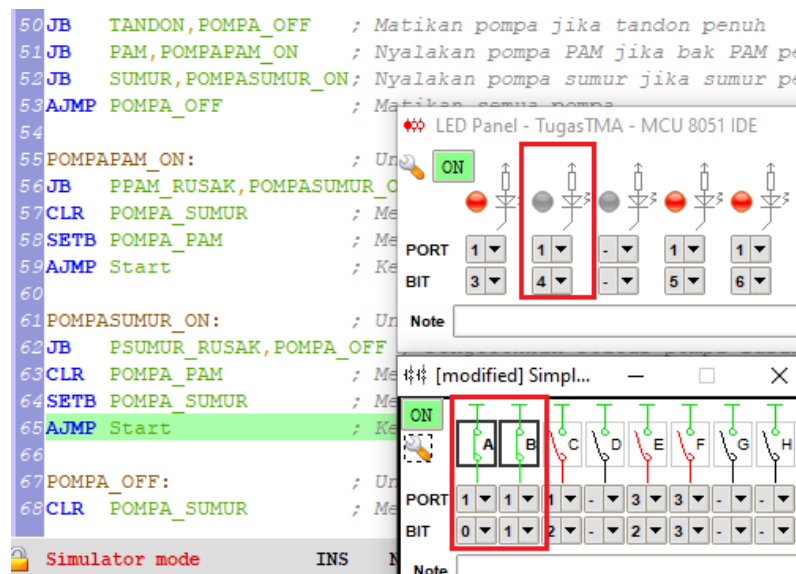
Dengan kondisi tandon kosong (0) dan bak PAM terisi (1), instruksi `JB TANDON, POMPA_OFF` (baris 50) akan dilewati dan instruksi `JB PAM, POMPAPAM_ON` (baris 51) akan dieksekusi. Karena belum terdapat *interrupt* untuk kerusakan pompa, maka pengecekan pompa rusak pada baris 56 akan dilewati dan pompa PAM dinyalakan (baris 58). Karena *switch* dan LED bersifat *active low*, maka *switch* P1.0 tertutup dan LED P1.3 mati. **Keluaran yang sama akan tercapai meskipun sumur terisi dan/atau pompa sumur rusak karena pompa bak PAM selalu dicek pertama kali.**



Gambar 5. Hasil eksekusi kondisi pertama

## 2. Kondisi Kedua: Tandon (0), Bak PAM (0), Sumur (1), (Tanpa *interrupt*)

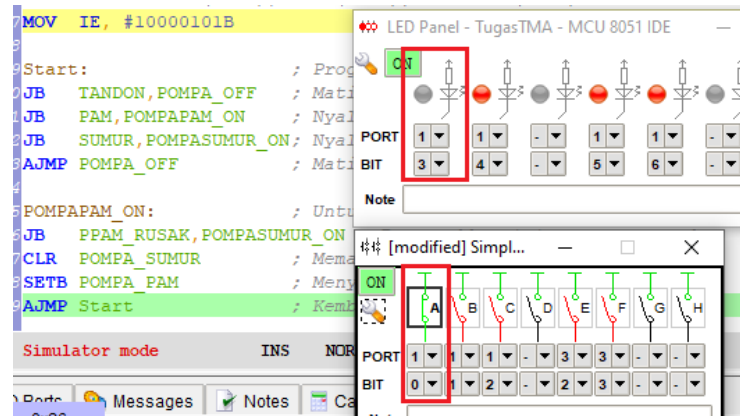
Dengan kondisi tandon kosong (0), bak PAM kosong (0), dan sumur terisi, instruksi pada baris 50 dan 51 akan dilewati dan akan dieksekusi perintah *jump* pada baris 52 yang melompat ke fungsi untuk menyalakan pompa sumur. Karena belum ada *interrupt* pompa rusak, maka pompa sumur dapat dinyalakan dan LED (P1.4) menjadi mati.



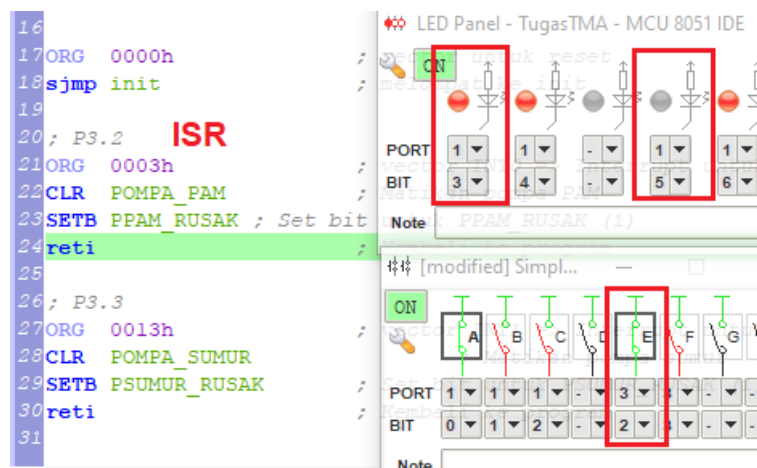
Gambar 6. Hasil eksekusi kondisi kedua

### 3. Kondisi Ketiga: Tandon (0), Bak PAM (1), Sumur (1), (Setelah *interrupt* EX0)

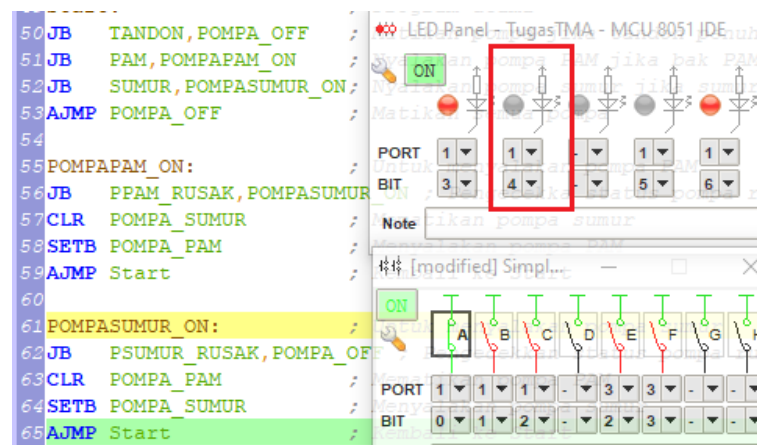
Dengan kondisi tandon 0 (kosong) dan bak PAM terisi (1), dan terdapat sinyal *interrupt* dari P3.2 (EX0), program akan langsung mengeksekusi ISR yang beralamat 0003h untuk *interrupt* tersebut yang berupa mematikan pompa PAM (**P1.3**) dan melakukan *set* untuk status PPAM\_RUSAK (**P1.5**). Setelahnya ketika program lompat ke fungsi POMPAPAM\_ON, program akan dialihkan ke fungsi POMPASUMUR\_ON yang menyalakan pompa sumur.



Gambar 7. Eksekusi #3 sebelum interrupt



Gambar 8. Eksekusi #3 saat terjadi interrupt (pompa PAM mati dan rusak)

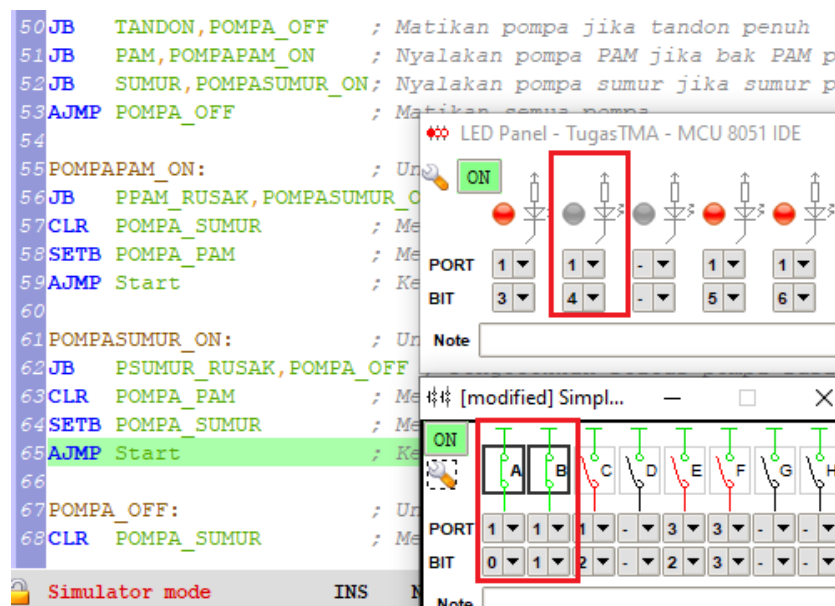


Gambar 9. Eksekusi #3 setelah interrupt (pompa sumur menyala)

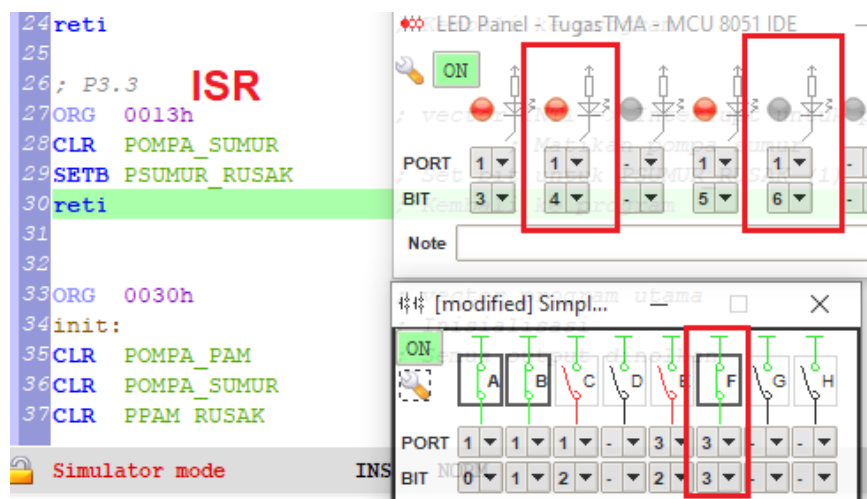


#### 4. Kondisi Keempat: Tandon (0), Bak PAM (0), Sumur (1), (Setelah Interrupt EX1)

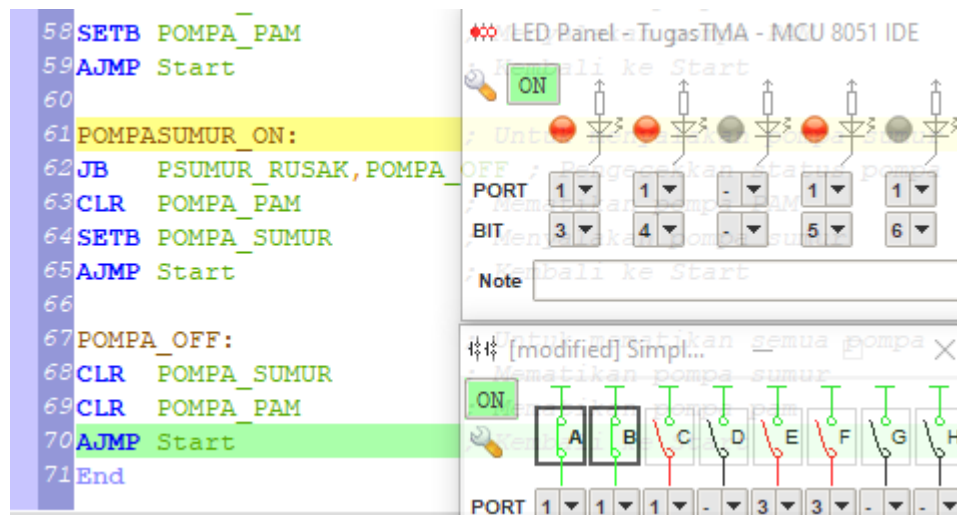
Dengan kondisi tandon 0 (kosong), bak PAM kosong (0), dan sumur terisi, instruksi pada baris 50 dan 51 akan dilewati dan akan dieksekusi perintah *jump* pada baris 52 yang melompat ke fungsi untuk menyalakan pompa sumur. Ketika terdapat sinyal *interrupt* pada pin P3.3 (EX1), program akan langsung mengeksekusi ISR untuk pin EX1 (P3.3) yang beralamat 0013h. Instruksi yang dijalankan adalah mematikan pompa sumur (P1.4) dan melakukan *set* untuk PSUMUR\_RUSAK (P1.6). Setelah kembali ke program utama, program akan mencoba untuk menyalakan pompa sumur, namun akan dialihkan ke POMPA\_OFF karena pompa sumur rusak dan bak PAM kosong. Setelahnya, semua pompa akan dimatikan.



Gambar 10. Eksekusi #4 sebelum interrupt



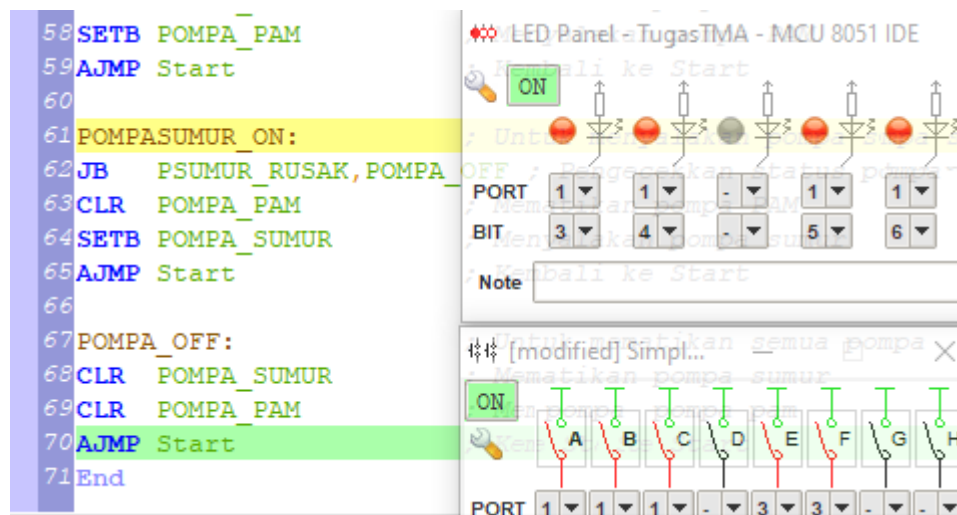
Gambar 11. Eksekusi #4 saat terjadi interrupt (pompa sumur mati dan rusak)



Gambar 12. Eksekusi #4 setelah interrupt (semua pompa mati)

## 5. Kondisi Kelima: Tandon (1)

Dengan kondisi tandon terisi (1), perintah JB TANDON, POMPA\_OFF akan langsung terksekusi dan tidak akan ada pompa yang dinyalakan. Selama tandon terisi, tidak akan ada pompa yang menyala walaupun dengan kombinasi kondisi apapun.



Gambar 13. Eksekusi #5

Terjadi perubahan isi memori untuk alamat 0x08 setelah terjadi *interrupt*.

	x0	x1	x2	x3	x4	x5	x6	x7
00	00	00	00	00	00	00	00	00
08	60	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00
18	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00
28	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00

Gambar 14. Perubahan isi memori

Folder project:

[https://drive.google.com/file/d/1QYsbKyVGZMMWKSfzVDNiF1Uwa9PAQbg/view?usp=share\\_link](https://drive.google.com/file/d/1QYsbKyVGZMMWKSfzVDNiF1Uwa9PAQbg/view?usp=share_link)