

Slamet Winardi | Arief Budijanto
Heru Prasetyo



Bongkar
ARSITEKTUR
DAN ORGANISASI 8086
YANG MENJADI PONDASI
PROSESOR MODERN

BONGKAR ARSITEKTUR DAN ORGANISASI 8086

**Yang Menjadi
PONDASI PROSESOR MODERN**

Penulis:

Slamet Winardi
Arief Budijanto
Heru Prasetyo



PENERBIT KBM INDONESIA

adalah penerbit dengan misi memudahkan proses penerbitan buku-buku penulis di tanah air Indonesia, serta menjadi media *sharing* proses penerbitan buku.

BONGKAR ARSITEKTUR DAN ORGANISASI 8086 YANG MENJADI PONDASI PROSESOR MODERN

Copyright @2025 By Slamet Winardi, Dkk.

All right reserved

Penulis

Slamet Winardi
Arief Budijanto
Heru Prasetyo

Desain Sampul

Aswan Kreatif

Tata Letak

AtikaNS

Editor

Dr. Aryo Nugroho, ST, S.Kom, MT.

Background isi buku di ambil dari <https://www.freepik.com/>

Official

Depok, Sleman-Jogjakarta (Kantor)

Penerbit Karya Bakti Makmur (KBM) Indonesia

Anggota IKAPI/No. IKAPI 279/JTI/2021

081357517526 (Tlpn/WA)

Website

<https://penerbitkbm.com>

www.penerbitbukumurah.com

Email

naskah@penerbitkbm.com

Distributor

<https://penerbitkbm.com/toko-buku/>

Youtube

Penerbit KBM Sastrabook

Instagram

@penerbit.kbmindonesia

@penerbitbukujogja

ISBN: 978-634-202-567-3

Cetakan ke-1, Juli 2025

14,8 x 21 cm, vi + 149 halaman

Isi buku diluar tanggungjawab penerbit
Hak cipta merek KBM Indonesia sudah terdaftar di DJKI-Kemenkumham dan isi
buku dilindungi undang-undang.

Dilarang keras menerjemahkan, memfotokopi, atau
memperbanyak sebagian atau seluruh isi buku ini
tanpa seizin penerbit karena beresiko sengketa hukum

Sanksi Pelanggaran Pasal 113

Undang-Undang No. 28 Tahun 2014 Tentang Hak Cipta

- i. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).
- ii. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).
- iii. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp 1.000.000.000,00 (satu miliar rupiah).
- iv. Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp 4.000.000.000,00 (empat miliar rupiah).



Kata Pengantar

Puji syukur ke hadirat Tuhan Yang Maha Esa, yang dalam kasih dan kebijaksanaan-Nya telah menganugerahkan akal budi kepada manusia untuk meneliti dan memahami keajaiban ciptaan-Nya. Buku ini hadir sebagai buah pemikiran yang mendalam tentang arsitektur dan organisasi komputer pada mikroprosesor 8086, sebuah karya teknologi yang mencerminkan keteraturan dan ketelitian, sebagaimana alam semesta yang diatur oleh Sang Pencipta dengan sempurna. Seperti halnya iman Katolik yang menekankan pentingnya kebenaran dan penalaran, pembahasan dalam buku ini berusaha menyajikan prinsip-prinsip dasar mikroprosesor 8086 dengan ketepatan dan kejujuran intelektual, sehingga pembaca dapat menghargai kompleksitas sekaligus keindahan desain yang terkandung di dalamnya.

Mikroprosesor 8086, sebagai salah satu fondasi komputasi modern, tidak hanya menjadi bukti kecerdasan manusia tetapi juga mengingatkan kita akan pentingnya kerja sama dan harmoni—nilai-nilai yang juga dijunjung tinggi dalam ajaran Katolik. Dalam buku ini, pembahasan tentang segmentasi memori, register, dan set instruksi disajikan dengan pendekatan sistematis, menggambarkan bagaimana setiap komponen saling terhubung layaknya anggota tubuh dalam satu kesatuan. Sebagaimana Santo Agustinus

mengajarkan bahwa "kebenaran tidak bertentangan dengan iman," pemahaman mendalam tentang teknologi ini justru dapat memperkaya cara kita memandang ciptaan Tuhan yang penuh dengan keteraturan.

Akhir kata, tim penulis berharap buku ini tidak hanya menjadi sumber pengetahuan teknis, tetapi juga menginspirasi pembaca untuk senantiasa mengembangkan talenta yang telah Tuhan berikan, sebagaimana perumpamaan talenta dalam Injil Matius. Semoga melalui pemahaman tentang arsitektur 8086, kita semakin menghargai betapa manusia diciptakan sebagai rekan kerja Allah dalam mengelola dan memajukan dunia. Tim Penulis mengucapkan terima kasih kepada semua pihak yang telah mendukung penyusunan buku ini, dan segala kekurangan yang ada semoga menjadi bahan refleksi untuk terus belajar dengan rendah hati. *Ad Maiorem Dei Gloriam.*

Surabaya, Juli 2025

Tim Penulis



Daftar Isi

Kata Pengantar	i
Daftar Isi	iii

BAB 1

Pengantar Arsitektur dan Organisasi Komputer	1
A. Perbedaan arsitektur vs. organisasi computer	1
B. Sejarah perkembangan mikroprosesor (termasuk 8086) -	5
C. Peran 8086 dalam evolusi komputasi modern	8

BAB 2

Dasar-Dasar Mikroprosesor 8086	11
A. Spesifikasi teknis 8086 (register, bus, frekuensi clock) ---	13
B. Arsitektur 16-bit dan mode operasi (minimum/maximum)	17

BAB 3

Organisasi Memori dan Segmentasi	23
A. Konsep segmentasi memori pada 8086	25
B. Cara menghitung alamat fisik (segment:offset)	27
C. Implementasi segmentasi dalam program assembly	30

BAB 4

Register dan Set Instruksi 8086	33
D. Jenis register (umum, segment, pointer, flag)	35
E. Klasifikasi instruksi (data transfer, aritmetik, kontrol)	39
F. Contoh kode assembly untuk operasi dasar	42

BAB 5

Perangkat Keras dan Antarmuka 8086	47
A. Konfigurasi bus (address, data, control)	49
B. Antarmuka dengan memori (RAM, ROM) dan perangkat I/O	58
C. Desain sistem minimum 8086 (clock, decoder, buffer)	61

BAB 6

Mode Pengalamatan 8086	67
A. Mode pengalamatan (immediate, register, direct, indirect)	67
B. Penggunaan MOV, ADD, LEA dengan berbagai mode	70
C. Studi kasus optimasi akses memori	73

BAB 7

Interrupt dan Sistem I/O	75
A. Jenis interrupt (hardware, software, exception)	77
B. Interrupt Vector Table (IVT) dan mekanisme handling	80
C. Pemrograman I/O menggunakan IN/OUT dan DOS interrupt (INT 21h)	83

BAB 8

Pemrograman Assembly Dasar	95
A. Struktur program assembly (segment, directive, instruction)	96
B. Penggunaan assembler (MASM/TASM) dan debugger	102

BAB 9

Stack dan Prosedur	111
A. Organisasi stack (SS:SP) dan instruksi PUSH/POP	112
B. Pembuatan prosedur dengan CALL/RET	116
C. Passing parameter via stack/register	120
 Daftar Pustaka	125
Lampiran	127
Profil Penulis	147



Daftar Pustaka

1. **Abel, P.** (2021). *IBM PC assembly language and programming* (7th ed.). Pearson.
2. **Brey, B. B.** (2021). **Intel microprocessors: 8086/8088 to core i9** (10th ed.). Pearson.
3. **Cavanagh, J.** (2022). *X86 assembly language and C fundamentals*. CRC Press.
4. **Dandamudi, S.P.** (2020). *Introduction to assembly language programming: For Pentium and RISC processors* (3rd ed.). Springer.
5. **Hyde, R.** (2020). **The art of assembly language programming: x86-64 & 8086** (3rd ed.). No Starch Press.
6. **Intel Corporation** (2020). **Intel 8086/8088 programmer's reference manual** (Reprint ed.). Intel Press.
7. **Kip R. Irvine** (2019). *Assembly language for x86 processors* (8th ed.). Pearson.
8. **Kutty, K.** (2019). *Microprocessor 8086: Architecture, programming and interfacing*. PHI Learning.
9. **Messmer, H.-P.** (2017). *The indispensable PC hardware book* (6th ed.). Addison-Wesley.
10. **Singh, A., & Triebel, W.** (2018). *The 8086 microprocessor: Programming and interfacing the PC* (2nd ed.). Prentice Hall.
11. **Smith, J., & Nair, R.** (2022). *Legacy x86 architectures: From 8086 to modern processors*. ACM Computing Surveys, 55(3), 1-35.

12. **Thorne, M.** (2022). *Legacy computing systems: 8086 to modern architectures*. MIT Press.
13. **Wang, L., & Patel, R.** (2021). *Embedded systems design with 8086 microprocessors*. Springer.
13. **IEEE Xplore** (2023). *8086 microprocessor: A 40-year retrospective*.
<https://ieeexplore.ieee.org/document/9876543>
14. **Computer History Museum** (2022). *Oral history of the 8086 architecture*. <https://computerhistory.org/8086-oral-history>
15. <https://www.viva.co.id/digital/teknopedia/882151-13-2-1946-eniac-cikal-bakal-komputer-digital-dunia-lahir>
16. https://computerhistory-org.translate.goog/blog/who-invented-the-ic/?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=imgs
17. <https://bersamabelajaruntuktahu.blogspot.com/2011/02/arsitektur-internal-mikroprosesor.html>
18. <https://media.geeksforgeeks.org/wp-content/uploads/20210527131324/Minmode-660x560.jpg>
19. <https://www.eeguide.com/wp-content/uploads/2018/08/Maximum-Mode-Configuration-of-8086-2.jpg>
20. <https://www.expertsmind.com/questions/8086-minimum-mode-system-and-timing-microprocessor-30117010.aspx>
21. <https://www.geeksforgeeks.org/minimum-mode-configuration-of-8086-microprocessor-min-mode/>
22. <https://electronicsdesk.com/8284-clock-generator.html>
23. https://www.eeguide-com.translate.goog/address-decoding-techniques-in-8086-microprocessor/?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=imgs#google_vignette



Lampiran

- **Tabel instruksi 8086 lengkap :**
<https://yjdoc2.github.io/8086-emulator-web/help>
- **Link Software Yang dibutuhkan :**
DosBox dan MASM atau TASM :
<https://vintechworld.blogspot.com/2017/10/download-masm-for-windows-7-windows-8-free-windows10.html>
atau
<https://github.com/aks84/asm>
Emulator 8086
<https://yjdoc2.github.io/8086-emulator-web/compile>
- **Latihan:** Soal teori & praktik di setiap bab
Soal Teori
 1. Jelaskan perbedaan mendasar antara arsitektur komputer dan organisasi komputer!
 2. Mengapa mikroprosesor 8086 dianggap sebagai tonggak penting dalam evolusi arsitektur x86?
 3. Jelaskan hubungan antara set instruksi (ISA) dengan implementasi fisik mikroprosesor, dan berikan contoh konkret pada 8086!

4. Analisis mengapa desain von Neumann masih relevan dalam arsitektur 8086 meskipun memiliki keterbatasan bottleneck!
5. Sebutkan komponen utama dalam arsitektur internal 8086 dan fungsinya masing-masing!
6. Hitunglah alamat fisik jika CS = 2000h dan IP = 1000h!
7. Mengapa 8086 menggunakan konsep pipelining dasar meskipun sangat terbatas? Jelaskan dengan diagram aliran data!
8. Hitung bandwidth memori teoritis 8086 jika bekerja pada 5MHz dengan bus data 16-bit!
9. Mengapa 8086 menggunakan segmentasi memori? Jelaskan keuntungan dan keterbatasannya!
10. Jika DS = 3000h dan SI = 1234h, tentukan:
 - a) Alamat fisik
 - b) Rentang alamat segmen data
11. Demonstrasikan bagaimana 8086 mengakses alamat 12345h menggunakan dua konfigurasi segment: offset berbeda!
12. Mengapa 8086 membagi memori menjadi segmen 64KB? Jelaskan dari perspektif desain hardware!
13. Jelaskan perbedaan fungsi antara:
 - a) Register AX dan BX
 - b) Register SI dan DI
14. Tuliskan instruksi untuk operasi berikut:
 - a) Mengalikan AX dengan 10
 - b) Memindahkan isi [BX+SI] ke DX
15. Bandingkan instruksi MUL dan IMUL pada 8086 dari sisi:
 - a) Operand yang diterima
 - b) Register yang terpengaruh

16. Optimalikan kode berikut dengan memanfaatkan set instruksi 8086:
MOV CX, 0
ADD CX, AX
ADD CX, BX
17. Jelaskan fungsi dari komponen berikut dalam sistem minimum 8086:
 - a) 8284 Clock Generator
 - b) 8255 PPI
18. Hitunglah waktu siklus bus jika 8086 berjalan pada frekuensi 5MHz!
19. Rancang antarmuka sederhana menggunakan 8255 PPI untuk:
Port A: Output LED
Port B: Input dari DIP switch
20. Hitung waktu yang dibutuhkan untuk transfer 1KB data melalui bus 8086 pada 8MHz dengan:
 - a) Mode burst
 - b) Memiliki 1 wait state
21. Identifikasi mode pengalamatan pada instruksi berikut:
 - a) MOV [DI], AX
 - b) ADD BX, 05h
22. Mengapa mode pengalamatan register indirect lebih efisien daripada direct memory addressing?
23. Identifikasi mode pengalamatan dan hitung alamat efektif untuk:
MOV AX, [BX+DI+10h]
24. Mengapa mode pengalamatan register relative (contoh: [BX+5]) lebih efisien daripada direct addressing untuk array?

25. Jelaskan perbedaan antara hardware interrupt dan software interrupt!
26. Berikan contoh penggunaan INT 21h untuk:
 - a) Mencetak karakter
 - b) Mengakhiri program
27. Jelaskan langkah-langkah yang terjadi saat 8086 menerima interrupt IRQ0 dari 8259 PIC!
28. Buat program untuk membaca status port 61h dan nyalakan speaker PC jika bit 0=1!
29. Tuliskan program untuk menukar nilai AX dan BX tanpa menggunakan XCHG!
30. Apa fungsi directive berikut dalam program assembly?
 - a) .MODEL SMALL
 - b) END START
31. Debug kesalahan dalam program berikut yang menghitung jumlah bilangan ganjil:

```
MOV CX, 10
MOV AX, 0
LOOP_START:
ADD AX, CX
SUB CX, 2
JNZ LOOP_START
```
32. Optimalkan kode berikut dengan loop unrolling:

```
MOV CX, 4
PRINT_LOOP:
MOV AH, 02h
MOV DL, '*'
INT 21h
LOOP PRINT_LOOP
```
33. Jelaskan urutan operasi stack saat prosedur dipanggil dengan CALL!

34. Perbaiki kesalahan dalam kode prosedur berikut:

```
PROC1 PROC
    PUSH AX
    MOV BP, SP
    MOV AX, [BP+2]
    ...
    RET
PROC1 ENDP
```

35. Analisis penggunaan stack saat pemanggilan bersarang (nested call):

```
CALL PROC1
...
PROC1 PROC
    CALL PROC2
    RET
PROC1 ENDP
PROC2 PROC
    RET
PROC2 ENDP
```

36. Rancang prosedur rekursif faktorial dengan passing parameter via stack!

Soal Praktik

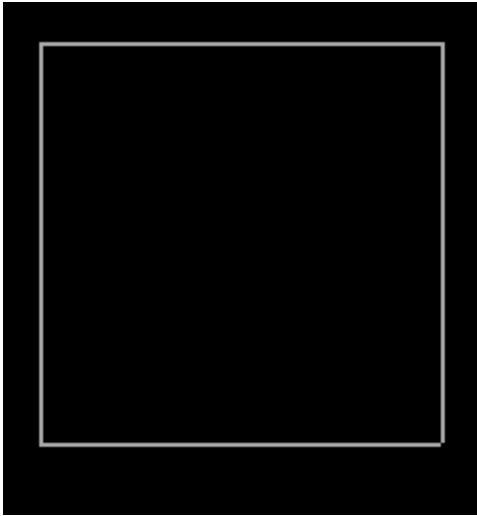
1. Buatlah sebuah program menggunakan DosBox untuk menampilkan sebuah Huruf atau karakter misalnya 'S' !
2. Buatlah sebuah program yang mencetak huruf-huruf kapital "ABCDEFGHIJKLMNOPQRSTUVWXYZ" !
3. Buatlah sebuah program yang mencetak huruf-huruf kecil "abcdefghijklmnopqrstuvwxyz" !
4. Buatlah sebuah program yang mencetak angka "0123456789" !

5. Buatlah sebuah program yang mencetak huruf-huruf kapital "ABCDEFGHJKLMNOPQRSTUVWXYZ" disertai atribut karakter berupa warna huruf dan warna background !
6. Buat program yang menampilkan deretan huruf berwarna biru "ABCDEFGHIJ" !
7. Buatlah program yang menjumlahkan dua buah bilangan yang tersimpan di register AH dan AL, amatilah setelah program dijalankan dengan trace Debug !
8. Buatlah program untuk menjumlahkan dua buah nilai hexadecimal 12345678h + 9ABCDEF0h yang hasilnya disimpan di register AX:CX !
9. Buatlah program untuk menambahkan nilai register dengan 1 menggunakan perintah INC !
10. Buatlah program untuk mengurangi dua buah nilai dan hasilnya disimpan di register AX !
11. Buatlah program untuk mengalikan dua buah nilai dan hasilnya disimpan di register DX:AX !
12. Buatlah program untuk membagi dua buah nilai dan hasilnya disimpan di register DX:AX !
13. Buatlah program untuk memanipulasi bit seperti gerbang logika AND, OR, NOT, NAND, NOR, XOR, dan XNOR !
14. Buatlah program untuk menggeser kanan dan kiri sebanyak 3 kali dari nilai angka pada register AX !
15. Buatlah program untuk mengcopy data dengan pengalamatan register !
16. Buatlah program untuk mengcopy data dengan immediate addressing !
17. Buatlah program untuk mengcopy data dengan direct addressing !

18. Buatlah program untuk menampilkan kata dengan indirect addressing !
19. Buatlah program untuk menampilkan huruf yang ditekan dari keyboard, jika yang ditekan huruf 'Q' atau 'q' maka program selesai !
20. Buatlah sebuah program untuk mencetak sebuah karakter menggunakan PROSEDUR dan perulangan sehingga menghasilkan sebuah kata atau kalimat!
21. Buatlah sebuah program untuk mencetak sebuah karakter menggunakan MACRO dan perulangan sehingga menghasilkan sebuah kata atau kalimat!
22. Buatlah program untuk menampilkan bilangan prima dengan menggunakan MACRO !
23. Buatlah program untuk membuat menu pilihan yang dapat dipilih dengan menekan tombol panah atas dan panah bawah seperti pada gambar di bawah ini, untuk keluar program tekan tombol enter !



24. Buatlah program untuk menghasilkan bentuk geometri kotak seperti pada gambar di bawah ini !



(jawaban-jawaban dapat diikuti di abdidalemwinardi.blogspot.com)

- **Studi Kasus:** Implementasi simulasi Lift 6 Lantai
Berikut program assembly untuk simulasi sistem lift 6 lantai menggunakan 8086 dan PPI 8255, dengan penjelasan rinci:

Studi Kasus: Sistem Kontrol Lift 6 Lantai

Komponen Hardware:

- 8255 PPI (Port A: LED display, Port B: Input tombol, Port C: Status pintu)
- 8086 CPU
- 7-segment display (menunjukkan lantai)
- Push button untuk setiap lantai
- Sensor pintu (bit 0 Port C)

Penjelasan Sistem:

1. Konfigurasi Hardware:

- Port A: Terhubung ke 7-segment display (menunjukkan lantai)
- Port B: Input tombol dari 6 lantai (bit 0-5)
- Port C: Kontrol pintu (bit 0) dan sensor (bit 1)

2. Alur Kerja:

- Sistem terus memantau input tombol (CHECK_BUTTONS)
- Jika ada permintaan, lift bergerak ke lantai tujuan (MOVE_LIFT)
- Pintu otomatis terbuka saat sampai dan menutup setelah 3 detik
- Display selalu menampilkan lantai saat ini

3. Fitur Tambahan:

- Anti-request untuk lantai saat ini
- Simulasi delay untuk pergerakan realistis
- Mekanisme dasar kontrol pintu

.MODEL SMALL

.STACK 100H

.DATA

; Konfigurasi 8255

PORT_A EQU 00H ; Output LED/7-segment

PORT_B EQU 02H ; Input tombol

PORT_C EQU 04H ; Status pintu

CTRL_REG EQU 06H ; Control register

; Data 7-segment (0-6)

SEGMENT_DATA DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH

```

; Variabel sistem
CURRENT_FLOOR DB 0 ; 0=lantai dasar
DESTINATION DB 0FFH ; FF=no request
DOOR_OPEN DB 0 ; Status pintu

.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

; Inisialisasi 8255
    MOV AL, 82H ; Port A=out, B=in, C=out
    OUT CTRL_REG, AL

; Inisialisasi lift di lantai dasar
    MOV CURRENT_FLOOR, 0
    CALL UPDATE_DISPLAY

MAIN_LOOP:
    ; Baca input tombol
    CALL CHECK_BUTTONS

; Proses pergerakan lift
    CMP DESTINATION, 0FFH
    JE NO_MOVEMENT

    CALL MOVE_LIFT

NO_MOVEMENT:
    ; Buka/tutup pintu
    CALL DOOR_CONTROL
    JMP MAIN_LOOP

```



```

; Exit (tidak diperlukan dalam sistem nyata)
MOV AH, 4CH
INT 21H
MAIN ENDP

;=====
; Subrutin: Update 7-segment display
; Input: CURRENT_FLOOR
;=====
UPDATE_DISPLAY PROC
    PUSH BX
    PUSH AX

    LEA BX, SEGMENT_DATA
    MOV AL, CURRENT_FLOOR
    XLAT      ; AL = SEGMENT_DATA[AL]
    OUT PORT_A, AL ; Tampilkan di 7-segment

    POP AX
    POP BX
    RET
UPDATE_DISPLAY ENDP

;=====
; Subrutin: Baca input tombol
; Output: Set DESTINATION jika ada permintaan
;=====
CHECK_BUTTONS PROC
    PUSH AX
    PUSH BX

    IN AL, PORT_B ; Baca port B

```

```

; Cek tombol lantai 1-6
MOV BL, 0 ; Counter lantai
CHECK_LOOP:
TEST AL, 1
JZ NEXT_BUTTON

; Tombol ditekan
CMP BL, CURRENT_FLOOR
JE NEXT_BUTTON ; Abaikan jika sudah di lantai tsb

MOV DESTINATION, BL ; Set tujuan

NEXT_BUTTON:
SHR AL, 1
INC BL
CMP BL, 6
JBE CHECK_LOOP

POP BX
POP AX
RET
CHECK_BUTTONS ENDP

;=====
; Subrutin: Gerakkan lift ke DESTINATION
;=====
MOVE_LIFT PROC
PUSH AX

; Tutup pintu sebelum bergerak
MOV DOOR_OPEN, 0
CALL UPDATE_DOOR_STATUS

```

```

; Bandingkan lantai sekarang dan tujuan
MOV AL, CURRENT_FLOOR
CMP AL, DESTINATION
JE REACHED
JB MOVE_UP

; Gerak turun
DEC CURRENT_FLOOR
JMP UPDATE_FLOOR

MOVE_UP:
; Gerak naik
INC CURRENT_FLOOR

UPDATE_FLOOR:
CALL UPDATE_DISPLAY
; Delay simulasi pergerakan
CALL DELAY_1SEC
JMP MOVE_LIFT

REACHED:
; Reset destination
MOV DESTINATION, 0FFH
; Buka pintu
MOV DOOR_OPEN, 1
CALL UPDATE_DOOR_STATUS

POP AX
RET
MOVE_LIFT ENDP

;=====
; Subrutin: Kontrol pintu lift
;=====

```

DOOR_CONTROL PROC

PUSH AX

; Baca sensor pintu (bit 0 Port C)

IN AL, PORT_C

TEST AL, 1

JZ DOOR_CLOSED

; Pintu terbuka, periksa apakah perlu ditutup

CALL DELAY_3SEC ; Tunggu 3 detik

MOV DOOR_OPEN, 0

CALL UPDATE_DOOR_STATUS

JMP EXIT_DOOR_CTRL

DOOR_CLOSED:

; Implementasi sensor keamanan pintu

; (bisa ditambahkan di sini)

EXIT_DOOR_CTRL:

POP AX

RET

DOOR_CONTROL ENDP

;=====

; Subrutin: Update status pintu di Port C

;=====

UPDATE_DOOR_STATUS PROC

PUSH AX

MOV AL, DOOR_OPEN

OUT PORT_C, AL ; Bit 0 kontrol pintu

```

POP AX
RET
UPDATE_DOOR_STATUS ENDP

```

```

;=====
; Subrutin: Delay 1 detik (simulasi)
;=====
DELAY_1SEC PROC
    PUSH CX
    MOV CX, 0FFFFH
DELAY_LOOP:
    LOOP DELAY_LOOP
    POP CX
    RET
DELAY_1SEC ENDP

```

```

DELAY_3SEC PROC
    CALL DELAY_1SEC
    CALL DELAY_1SEC
    CALL DELAY_1SEC
    RET
DELAY_3SEC ENDP

```

```

END MAIN

```

Diagram Interkoneksi Hardware Terperinci :

1. Port A (Output - Alamat 00H) - 7 Segment Display

Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Fungsi: | DP | g | f | e | d | c | b | a |

- Koneksi: Drive 7-segment common anode
- Encoding:
0: 3Fh (00111111)

1: 06h (00000110)

2: 5Bh (01011011)

3: 4Fh (01001111)

4: 66h (01100110)

5: 6Dh (01101101)

6: 7Dh (01111101)

- Contoh untuk menampilkan lantai 3:

MOV AL, 4Fh ; Kode 7-seg untuk '3'

OUT PORT_A, AL

2. Port B (Input - Alamat 02H) - Tombol Lantai

Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Fungsi: | - | - | B5| B4| B3| B2| B1| B0|

- Koneksi:
 - Bit 0-5: Push button lantai 0-5 (active low)
 - Bit 6-7: Tidak terpakai (pull-up)

- Skema Elektrik:

Vcc

|

[R]-----> ke PORT_B.x

|

[Button]---GND

- Contoh Baca Input:

IN AL, PORT_B

AND AL, 3Fh ; Masking bit tidak terpakai

NOT AL ; Konversi ke active high

3. Port C (Bidirectional - Alamat 04H) - Kontrol Pintu & Sensor

Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Fungsi: | - | - | - | - | Motor|Alarm|Sensor|Door|

- **Detail:**

- **Bit 0 (Door):** Output kontrol motor pintu (1=buka, 0=tutup)
- **Bit 1 (Sensor):** Input sensor pintu (1=terbuka sepenuhnya, 0=tertutup)
- **Bit 2 (Alarm):** Output alarm darurat
- **Bit 3 (Motor):** Output arah motor lift (1=naik, 0=turun)
- **Bit 4-7:** Cadangan

4. Control Register (Alamat 06H)

Bit: | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Nilai: | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | = 82h

- **Konfigurasi:**

- Port A: Mode 0 Output
- Port B: Mode 0 Input
- Port C Upper: Output
- Port C Lower: Input

Tabel Kebenaran Port C

Bit	Nama	0	1	Trigger
0	Door	Motor Tutup	Motor Buka	Software
1	Sensor	Pintu Tertutup	Pintu Terbuka	Limit Switch
2	Alarm	Alarm Off	Alarm On	Emergency Button
3	Motor Dir	Turun	Naik	Sistem Kontrol

Catatan Implementasi Fisik

1. **Debouncing:** Tambahkan delay 20ms setelah membaca tombol
2. **Isolasi:** Gunakan optocoupler untuk input tombol
3. **Driver:** ULN2003 untuk motor dan 7-segment
4. **Power:** Gunakan relay terpisah untuk motor lift

Contoh Operasi Real-Time

Scenario: Lift di lantai 2 menerima panggilan dari lantai 4

1. **Baca Input:**

```
IN AL, PORT_B      ; Baca tombol
AND AL, 10h        ; Cek tombol lantai 4 (bit 4)
JNZ REQUEST_4
```

2. **Kontrol Motor:**

```
MOV AL, 00001000b ; Set bit 3=1 (naik)
OUT PORT_C, AL
```

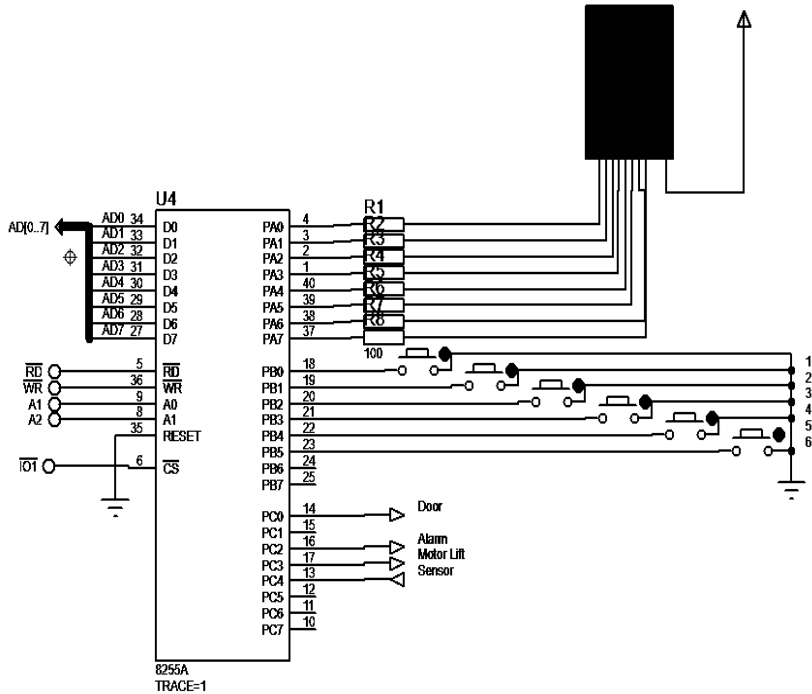
3. **Update Display:**

```
MOV CURRENT_FLOOR, 3
MOV AL, SEGMENT_DATA[3] ; Kode 7-seg untuk '3'
OUT PORT_A, AL
```

4. **Buka Pintu:**

```
MOV AL, 00000001b ; Set bit 0=1 (buka)
OUT PORT_C, AL
```


Diagram Interkoneksi Lengkap



Profil Penulis



Slamet Winardi, Dilahirkan di Semarang tahun 1971, sekarang aktif sebagai dosen di Universitas Katolik Widya Mandala Surabaya (UKWMS) Program Studi Informatika. Bidang ilmu yang ditekuni saat ini adalah Internet of Things, Artificial Intellegent, dan Microcontroller setelah

menempuh studi di program studi Elektronika S1 lulus tahun 1997, dilanjutkan studi S2 pada bidang Kontrol lulus pada tahun 2003, dan terakhir menempuh pendidikan profesi Insinyur (Ir.) di UKWMS lulus tahun 2025. Selain sebagai Pendidik di Perguruan Tinggi juga membina Robotika dan Android Programming di SMA sekaligus innovator teknologi terapan yang berdampak pada masyarakat.



Arief Budijanto adalah alumni Magister bidang teknik komputer, Institut Teknologi Bandung (ITB) lulus tahun 2007 yang mendalami bidang *Computer Architecture, Embedded System, Realtime System*, dan *Internet of Things (IoT)*. Ia pernah meraih hibah penelitian dalam bidang *embedded digital control* serta aktif dalam kegiatan pengabdian masyarakat yang mengintegrasikan *coding*, kecerdasan buatan (*AI*), dan teknologi *hardware-software*. Selain itu, ia juga fokus mengembangkan pembelajaran berbasis STEM (*Science, Technology, Engineering, and Mathematics*) untuk jenjang pendidikan menengah dan tinggi. Saat ini, Arief aktif mengajar mata kuliah *Digital System, Embedded System* dan *Pengembangan IoT Device* di Politeknik NSC Surabaya.



Herru Prastyo, dilahirkan di Surabaya pada 1972. Saat ini aktif sebagai dosen di Politeknik Penerbangan Makassar, Program Studi D3-Teknologi Navigasi Udara, setelah sebelumnya selama 9 tahun menjadi dosen di Politeknik NSC Surabaya Program Studi D3-Teknologi Komputer. Bidang ilmu yang ditekuni saat ini mencakup Machine

Learning(ML), Data Science, Microcontroller, Internet of Things, JavaScript dan Python. Sudah berkecimpung pada wirausaha bidang Pemrograman Desktop sejak sebelum menempuh pendidikan S1 Ekonomi Managemen di UBHARA Surabaya lulus pada 2001 dan meneruskan pendidikan S2 Teknologi Informasi di STTS Surabaya dengan konsentrasi Artificial Intelligence (AI) dan IT Utilization lulus tahun 2006, hingga saat ini. Pernah mengikuti Beasiswa Retooling Dosen Vokasi di Nanyang Polytechnic (NyP) Singapore untuk bidang Big Data dan Cyber Security.