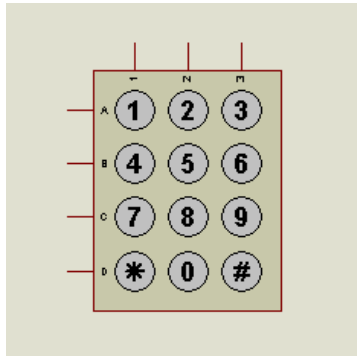


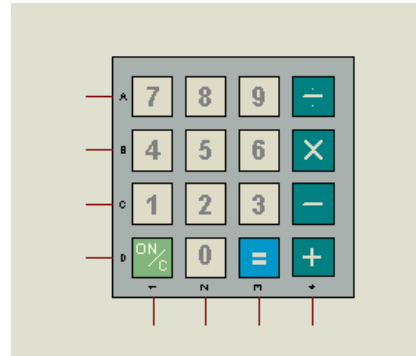
## KEYPAD

### “Oupthu Seven Segment dengan Input KeyPad”

Keypad adalah saklar-saklar push button yang disusun secara matriks yang berfungsi untuk menginput data seperti, input pintu otomatis, input absensi, input datalogger dan sebagainya. Push Button yang tersusun tersebut dapat berupa 4x4 (4 kolom, 4 baris) seperti Gambar 1a. atau 3x4 (3 kolom 4 baris) seperti Gambar 1b.

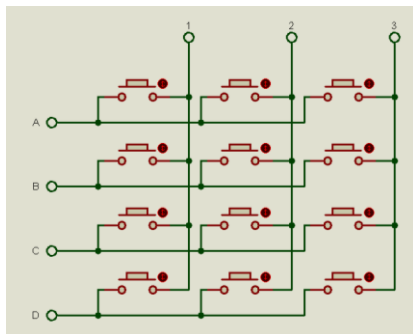


Gambar 1a.

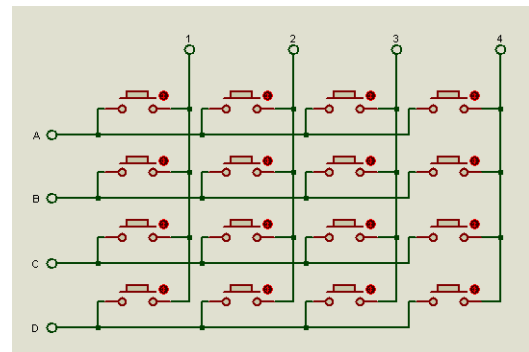


Gambar 1b.

Susunan Push Button pada masing – masing Keypad tersebut dapat digambarkan pada Gambar 2a (susunan Push Button pada Keypad 3x4) dan Gambar 2b (susunan Push Button pada Keypad 4x4).



Gambar 2a.



Gambar 2b.

Pada Keypad 3x4 hanya membutuhkan pin I/O sebanyak 7 pada microcontroller (3 pin output menuju kolom 1,2,3 dan 4 Input dari baris A,B,C,D) dimana jika dengan menggunakan Push Button biasa membutuhkan 12 I/O, sedangkan pada Keypad 4x4 hanya membutuhkan pin I/O sebanyak 8 pada microcontroller (4 pin output menuju kolom 1,2,3 dan 4 Input dari baris A,B,C,D) dimana jika dengan menggunakan Push Button biasa membutuhkan 16 I/O.

Keypad dapat digunakan untuk perangkat input dengan menggunakan teknik Scanning yaitu dengan memberikan Sinyal High (+5 volt) dari Microcontroller ke masing-masing kolom (Keypad) secara bergantian sembari mendeteksi apakah pin input microcontroller yang menerima output baris A,B,C,D (Keypad) menerima Sinyal High.

Pada Keypad 3x4 dapat digunakan Algoritma berikut untuk proses Scanning nya (pada Microcontroller) :

1. Mulai
2. Berikan sinyal high ke kolom 1,2,3

3. Cek apakah salah satu pin input (dari keypad A,B,C,D) menerima sinyal High ?, jika iya maka Proses scanning dilakukan (no.4)
4. Proses Scanning
  - Berikan sinyal high hanya ke kolom 1
  - Jika input dari baris A bernilai High, maka berikan aksi (nilai 1)
  - Jika input dari baris B bernilai High, maka berikan aksi (nilai 4)
  - Jika input dari baris C bernilai High, maka berikan aksi (nilai 7)
  - Jika input dari baris D bernilai High, maka berikan aksi (nilai \*)
  - Berikan sinyal high hanya ke kolom 2
  - Jika input dari baris A bernilai High, maka berikan aksi (nilai 2)
  - Jika input dari baris B bernilai High, maka berikan aksi (nilai 5)
  - Jika input dari baris C bernilai High, maka berikan aksi (nilai 8)
  - Jika input dari baris D bernilai High, maka berikan aksi (nilai 0)
  - Berikan sinyal high hanya ke kolom 3
  - Jika input dari baris A bernilai High, maka berikan aksi (nilai 3)
  - Jika input dari baris B bernilai High, maka berikan aksi (nilai 6)
  - Jika input dari baris C bernilai High, maka berikan aksi (nilai 9)
  - Jika input dari baris D bernilai High, maka berikan aksi (nilai #)
  - delay
5. Selesai

Bahasa Assembly untuk algoritma di atas adalah seperti contoh script di bawah ini

```

loop
    movlw    b'111'        ;
    movwf    PORTE         ; PORTE aktif (menuju keypad kolom 1,2,3)

                                ; Cek apakah ada salah satu Push Button yang ditekan
    btfsc    PORTA,0        ; Jika diantara PB baris A ditekan
    call     scan_keypad    ; maka lakukan proses scanning
    btfsc    PORTA,1        ; Jika diantara PB baris B ditekan
    call     scan_keypad    ; maka lakukan proses scanning
    btfsc    PORTA,2        ; Jika diantara PB baris C ditekan
    call     scan_keypad    ; maka lakukan proses scanning
    btfsc    PORTA,4        ; Jika diantara PB baris D ditekan
    call     scan_keypad    ; maka lakukan proses scanning
    nop      ; jika tidak ada yang ditekan maka tidak melakukan proses scanning

    goto     loop

scan_keypad
    movlw    b'000'        ;
    movwf    PORTE         ; PORTE nonaktif

    movlw    b'001'        ; kolom 1 aktif
    movwf    PORTE
    btfsc    PORTA,0        ; baris A (keypad 1)
    call     tampil_1
    btfsc    PORTA,1        ; baris B (keypad 4)
    call     tampil_4
    btfsc    PORTA,2        ; baris C (keypad 7)
    call     tampil_7
    btfsc    PORTA,4        ; baris D (keypad *)
    nop

    movlw    b'010'        ; kolom 2 aktif
    movwf    PORTE
    btfsc    PORTA,0        ; baris A (keypad 2)
    call     tampil_2
    btfsc    PORTA,1        ; baris B (keypad 5)
    call     tampil_5
  
```

```

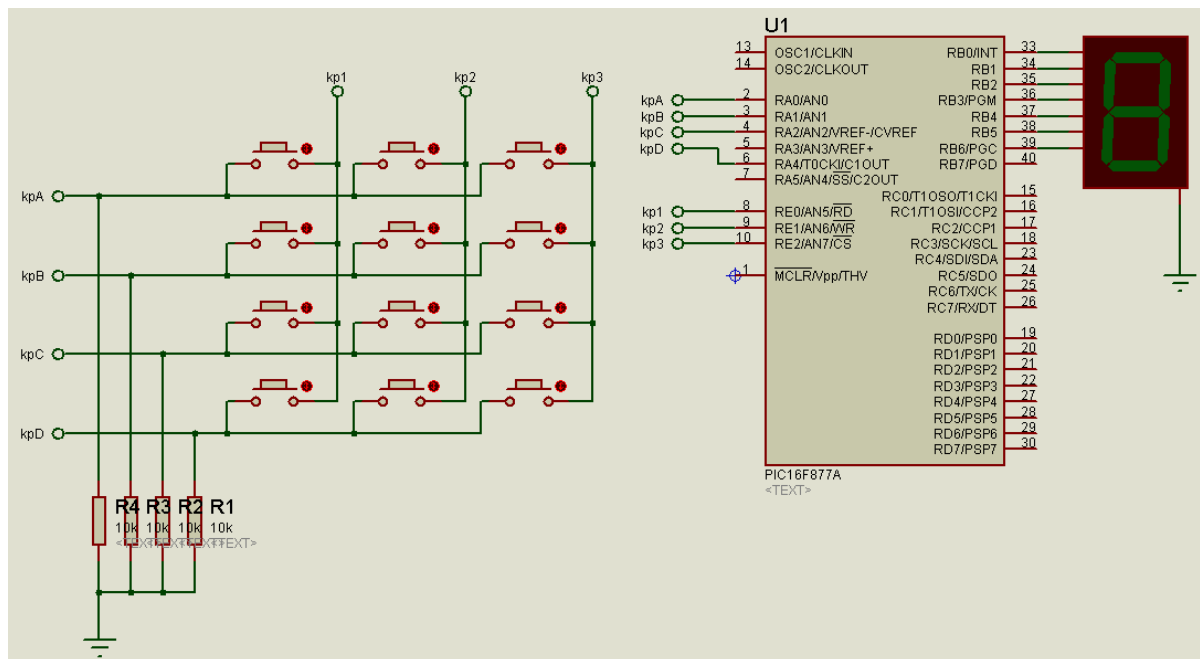
btfsc    PORTA,2          ; baris C (keypad 8)
call     tampil_8
btfsc    PORTA,4          ; baris D (keypad 0)
call     tampil_0
nop

movlw    b'100'          ; kolom 3 aktif
movwf    PORTE
btfsc    PORTA,0          ; baris A (keypad 3)
call     tampil_3
btfsc    PORTA,1          ; baris B (keypad 6)
call     tampil_6
btfsc    PORTA,2          ; baris C (keypad 9)
call     tampil_9
btfsc    PORTA,4          ; baris D (keypad #)
nop

call     delay
return

```

Sebagai latihan maka akan dibuat sistem untuk menampilkan angka 0 s/d 9 sesuai dengan tombol KeyPad yang ditekan seperti rangkaian dibawah ini ,



Rangkaian dengan kondisi seperti dijelaskan di atas dapat direalisasikan menggunakan pemrograman Bahasa Assembly pada Aplikasi MPLAB dengan script di bawah ini

```

STATUS    equ 03h          ; STATUS merujuk ke alamat memori 03h
PORTA     equ 05h          ; PORTA merujuk ke alamat memori 05h
PORTB     equ 06h          ; PORTB merujuk ke alamat memori 06h
PORTC     equ 07h          ; PORTC merujuk ke alamat memori 07h
PORTE     equ 09h          ; PORTE merujuk ke alamat memori 09h
TRISA     equ 85h          ; TRISA merujuk ke alamat memori 85h
TRISB     equ 86h          ; TRISB merujuk ke alamat memori 86h
TRISC     equ 87h          ; TRISC merujuk ke alamat memori 87h
TRISE     equ 89h          ; TRISE merujuk ke alamat memori 89h
ADCON1    equ 9Fh          ; ADCON1 merujuk ke alamat memori 9Fh

```

```

cblock h'20'
delay_1
delay_2
endc

```

```

org 00                ; origin register
clr                  ; Clear the working register
bsf    STATUS,5      ; ke Bank1
movlw  0x06          ; konfigurasi seluruh PIN
movwf  ADCON1        ; sebagai digital inputs
movlw  0x1F          ;
movwf  TRISA         ; TRISA bit 4-0 bernilai 1
                        ; menyebabkan PORTA bit 4-0 menjadi INPUT

movlw  0x00          ;
movwf  TRISB         ; TRISB seluruh bitnya diberi nilai 0
                        ; menyebabkan Seluruh Pin PORTB menjadi OUTPUT

movlw  b'000'        ;
movwf  TRISE         ; TRISC bit<0-2> bernilai 0 bit<3-7> bernilai 1
                        ; menyebabkan PORTC bit<0-2> sbg OUTPUT
                        ; dan PORTC bit<3-7> sbg INPUT
                        ; ke Bank0 (karena PORTA dan PORTB ada di Bank0)
bcf    STATUS,5      ; PORTA di-clear
clrf   PORTA         ; PORTB di-clear
clrf   PORTB         ; PORTC di-clear
;clrf  PORTC
call   tampil_0

loop
    movlw  b'111'    ;
    movwf  PORTE     ; PORTE aktif (menuju keypad kolom 1,2,3)
                        ; Cek apakah ada salah satu Push Button yang ditekan
    btfsc  PORTA,0   ; Jika diantara PB baris A ditekan
    call   scan_keypad ; maka lakukan proses scanning
    btfsc  PORTA,1   ; Jika diantara PB baris B ditekan
    call   scan_keypad ; maka lakukan proses scanning
    btfsc  PORTA,2   ; Jika diantara PB baris C ditekan
    call   scan_keypad ; maka lakukan proses scanning
    btfsc  PORTA,4   ; Jika diantara PB baris D ditekan
    call   scan_keypad ; maka lakukan proses scanning
    nop        ; jika tidak ada yang ditekan maka tidak melakukan proses scanning

    goto   loop

scan_keypad
    movlw  b'000'    ;
    movwf  PORTE     ; PORTE nonaktif

    movlw  b'001'    ; kolom 1 aktif
    movwf  PORTE
    btfsc  PORTA,0   ; baris A (keypad 1)
    call   tampil_1
    btfsc  PORTA,1   ; baris B (keypad 4)
    call   tampil_4
    btfsc  PORTA,2   ; baris C (keypad 7)
    call   tampil_7
    btfsc  PORTA,4   ; baris D (keypad *)
    nop

    movlw  b'010'    ; kolom 2 aktif
    movwf  PORTE
    btfsc  PORTA,0   ; baris A (keypad 2)
    call   tampil_2
    btfsc  PORTA,1   ; baris B (keypad 5)
    call   tampil_5
    btfsc  PORTA,2   ; baris C (keypad 8)
    call   tampil_8
    btfsc  PORTA,4   ; baris D (keypad 0)
    call   tampil_0
    nop

    movlw  b'100'    ; kolom 3 aktif
    movwf  PORTE
    btfsc  PORTA,0   ; baris A (keypad 3)
    call   tampil_3
    btfsc  PORTA,1   ; baris B (keypad 6)
    call   tampil_6
    btfsc  PORTA,2   ; baris C (keypad 9)
    call   tampil_9

```

```

        btfsc    PORTA,4      ; baris D (keypad #)
        nop

        call     delay
        return

tampil_0
        movlw   b'00111111'
        movwf   PORTB
        return

tampil_1
        movlw   b'00000110'
        movwf   PORTB
        return

tampil_2
        movlw   b'01011011'
        movwf   PORTB
        return

tampil_3
        movlw   b'01001111'
        movwf   PORTB
        return

tampil_4
        movlw   b'01100110'
        movwf   PORTB
        return

tampil_5
        movlw   b'01101101'
        movwf   PORTB
        return

tampil_6
        movlw   b'01111101'
        movwf   PORTB
        return

tampil_7
        movlw   b'00000111'
        movwf   PORTB
        return

tampil_8
        movlw   b'01111111'
        movwf   PORTB
        return

tampil_9
        movlw   b'01101111'
        movwf   PORTB
        return

delay
        movlw   d'255'        ; isi w-reg dengan desimal 255 (nilai maksimal 255)
                                ; menentukan waktu delay.
        movwf   delay_1       ; isi memori delay_1 dengan nilai w-reg
        movwf   delay_2       ; isi memori delay_2 dengan nilai w-reg

delay_loop
        decfsz  delay_1, f; delay_1 = delay_1 - 1, jika delay_1 = 0 maka lompat
        goto   delay_loop    ; kembali kurangi delay_1
        decfsz  delay_2, f; delay_2 = delay_2 - 1, jika delay_2 = 0 maka lompat
        goto   delay_loop    ; kembali kurangi delay_2
        return

end

```