



Innovative Electronics

Keypad and 7 Segment Display DT51 KND

USER'S GUIDE



manual
book



Daftar Isi

1.	Pendahuluan	1
	1.1 Spesifikasi DT51 KND.....	1
	1.2 Tata Letak DT51 KND.....	1
2.	Memasang DT51 KND	3
	2.1 Perlengkapan DT51 KND.....	3
	2.2 Sistem yang Dianjurkan.....	3
	2.3 Hubungan DT51 Ver 3.0 dengan DT51 KND.....	3
	2.4 Pengujian DT51 KND.....	4
3.	Perangkat Lunak DT51 KND	4
	3.1 Modul dan Rutin	5
	3.2 Contoh Program.....	10
	Appendix	
	Skema DT51 KND.....	19

Trademarks & Copyright

AT is a trademark of International Business Machines Corp.

IBM, PC, and PC-DOS are trademarks of International Business Machines Corp.

MS-DOS is a registered trademark of Microsoft Corporation.

Pentium is a registered trademark of Intel Corporation.

ALDS is copyright by Himnunan Mahasiswa Elektronika Sekolah Tinggi Teknik Surabaya.

1. Pendahuluan

DT51 KND merupakan Keypad And 7 Segment Display add-on board untuk 89C51 Development Tools DT51 Ver 3.0. DT51 KND berfungsi untuk menerima input tombol keypad serta menampilkan digit / karakter yang diinginkan ke 7 Segment Display. Dengan dilengkapi rutin-rutin untuk keypad maupun display maka DT51 KND sangat mudah digunakan dalam sistem kontrol yang membutuhkan input tombol dan tampilan.

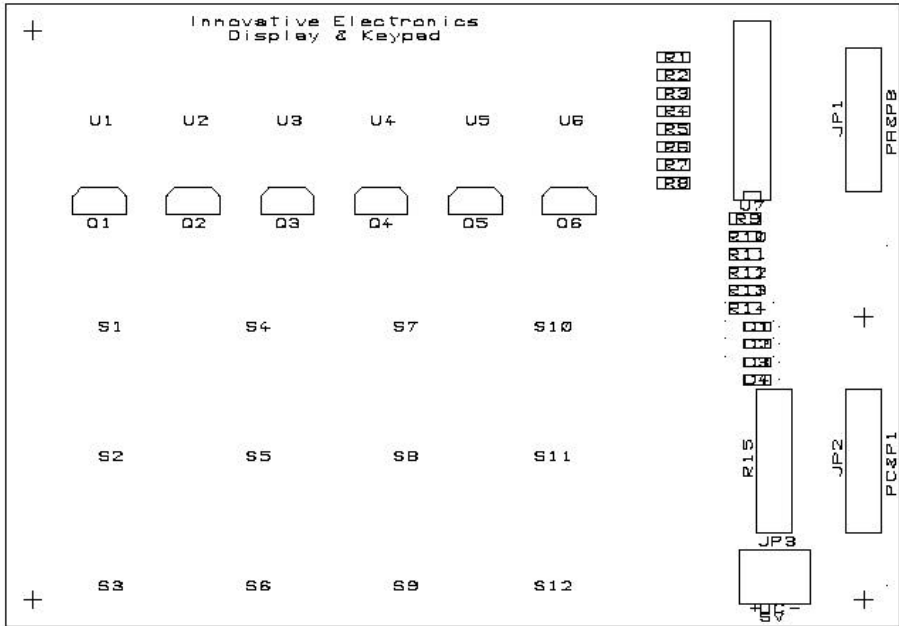
1.1 Spesifikasi DT51 KND

Spesifikasi DT51 KND sebagai berikut :

- Kompatibel penuh dengan DT51 Ver 3.0.
- 12 tombol Keypad dengan susunan matriks 3 x 4..
- 6 digit 7 Segment Display dengan tinggi 0.5 inch jenis super bright (sangat terang).
- Menggunakan sistem scanning.
- Kemudahan interface, mudah dihubungkan dengan mikroprosesor / mikrokontroler lain.
- Tersedia rutin siap pakai untuk Keypad dan 7 Segment Display.

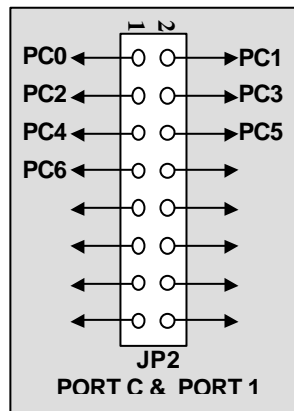
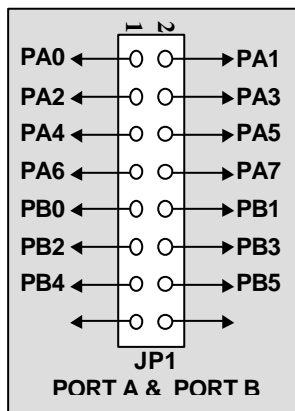
1.2 Tata Letak DT51 KND

Tata letak DT51 KND ditunjukkan pada gambar 1-1. Keypad dan 7 Segment Display diletakkan pada sisi yang berlainan, sehingga bila diinginkan pemasangan pada panel akan menjadi mudah dan rapi.



Gambar 1-1
Tata Letak DT51 KND

Detail urutan masing-masing konektor sebagai berikut :



2. Memasang DT51 KND

Berikut ini akan dijelaskan persiapan serta langkah-langkah yang diperlukan untuk menghubungkan DT51 KND dengan DT51 Ver 3.0 atau sistem yang lain.

2.1 Perlengkapan DT51 KND

Setiap kemasan DT51 KND berisi :



1 buah board DT51



1 disket program



1 buku manual

2.2 Sistem yang Dianjurkan

Perangkat keras :

- PC XT / AT / Pentium™ IBM Compatible dengan port serial (COM1 / COM2).
- Board DT51 Ver 3.0
- Floppy Disk 3.5", kapasitas 1,44 Mbytes.
- Hard Disk dengan kapasitas minimum 500 Kbytes

Perangkat lunak :

- Sistem operasi MS-DOS™ atau PC-DOS™.
- File-file yang ada pada disket program.

2.3 Hubungan DT51 Ver 3.0 dengan DT51 KND

DT51 KND memang di-desain sebagai add on board DT51 Ver 3.0, tetapi tidak menutup kemungkinan untuk dihubungkan dengan sistem mikroprosesor / mikrokontroler yang lain. Apabila Anda ingin menghubungkan DT51 KND dengan sistem yang lain kami sarankan untuk mempelajari skema DT51 KND (lihat Appendix).

Untuk menghubungkan DT51 KND dengan DT51 Ver 3.0 dianjurkan untuk menggunakan kabel pita (flat ribbon cable). Hubungannya ditunjukkan pada tabel-tabel berikut :

DT51 [Port A & Port B]	PA0	PA1	PA2	PA3	PA4	PA5	PA6
Pin	1	2	3	4	5	6	7
DT51 KND [JP1]	PA0	PA1	PA2	PA3	PA4	PA5	PA6
Pin	1	2	3	4	5	6	7

DT51 [Port A & Port B]	PA7	PB0	PB1	PB2	PB3	PB4	PB5
Pin	8	9	10	11	12	13	14
DT51 KND [JP1]	PA7	PB0	PB1	PB2	PB3	PB4	PB5
Pin	8	9	10	11	12	13	14

DT51 [Port C & Port 1]	PC0	PC1	PC2	PC3	PC4	PC5	PC6
Pin	1	2	3	4	5	6	7
DT51 KND [JP2]	PC0	PC1	PC2	PC3	PC4	PC5	PC6
Pin	1	2	3	4	5	6	7

DT51 KND memerlukan catu daya 5V DC yang dihubungkan dengan konektor JP3. Perhatikan polaritasnya jangan sampai terbalik, karena dapat mengakibatkan kerusakan.

2.4 Pengujian DT51 KND

Pengujian DT51 KND dapat dilakukan dengan menghubungkan board DT51 KND dengan board DT51 Ver 3.0 (baca bagian 2.3). Hubungkan catu daya ke masing-masing board. Kemudian jalankan program TESKND1.EXE (memori eksternal) atau TESKND2.EXE (memori internal) yang terdapat pada disket program. 7 Segment akan menampilkan angka 0-9 dan penekanan tombol keypad akan menampilkan angka 0-9 serta fungsi geser ke kanan atau ke kiri.

3. Perangkat Lunak DT51

Disket DT51 KND berisi file-file berikut :

- **Test File**
 TESKND1.EXE, TESKND1.HEX, dan TESKND1.ASM
 (menggunakan memori eksternal).
 TESKND2.EXE, TESKND2.HEX, dan TESKND2.ASM
 (menggunakan memori internal).
- **Sample File**
 SAMPLE1.ASM, SAMPLE2.ASM, SAMPLE3.ASM, dan
 SAMPLE4.ASM.
- **Modul dan Library.**
 DATADIS.ASM : display library berisi kode karakter untuk
 tampilan 7 Segment.
 DATAKEY.ASM : keypad library berisi kode untuk
 penekanan tombol keypad.
 KNDEXT.ASM : modul menggunakan memori eksternal
 KNDINT.ASM : modul menggunakan memori internal
- **Datasheet**
 ULN2803A.PDF : datasheet ULN2803A

3.1 Modul dan Rutin

Pada DT51 KND telah disediakan 2 buah modul program yaitu KNDEXT.ASM dan KNDINT.ASM. Modul-modul ini berisi rutin-rutin yang mempermudah Anda dalam mengembangkan program untuk Keypad dan 7 Segment Display dengan DT51 Ver 3.0.

A. KNDEXT.ASM

Modul ini menggunakan memori eksternal (EEPROM) sebagai memori display, modul ini cocok digunakan bila pada memori internal 89C51 sudah tidak tersedia cukup ruang lagi (biasanya dipakai untuk stack memory dan variabel). Kelebihan dari modul ini adalah karakter tampilan tersimpan pada Non Volatile memory sehingga tidak hilang saat catu daya mati dan dapat dipakai lagi saat catu daya menyala. Resources DT51 yang dipakai (otomatis tidak dapat dipakai untuk keperluan lain) adalah :

- Memori internal dengan alamat 18H-20H.
- Timer 0 dan interuptnya
- PPI 8255

Adapun rutin-rutin pada modul ini sebagai berikut :

InitKND

Fungsi : Menginisialisasi PPI dan Interupt untuk keperluan scanning display dan keypad.

Input : -

Output : -

Remark : Sebelum memakai rutin-rutin yang lain pada modul KNDXT.ASM, Anda harus memanggil rutin ini.

Write7S

Fungsi : Menuliskan karakter ke memori display untuk ditampilkan.

Input : ACC diisi dengan karakter yang akan ditulis.

Output : -

Karakter yang tersedia pada library sebagai berikut :

Kode	Karakter
0H	'0'
1H	'1'
2H	'2'
3H	'3'
4H	'4'
5H	'5'
6H	'6'
7H	'7'
8H	'8'
9H	'9'
0AH	'.'
0BH	'A'

Kode	Karakter
0CH	'C'
0DH	'd'
0EH	'E'
0FH	'F'
10H	'G'
11H	'H'
12H	'i'
13H	'J'
14H	'L'
15H	'o'
16H	'P'
17H	'U'

Contoh penggunaan :

```
MOV     A,#0H
LCALL  Write7S ; menuliskan '0' pada memory display serta
              ditampilkan
```

ClrDisMem

Fungsi : Menghapus seluruh karakter pada memori display dan tampilan.

Input : -

Output : -

Remark : -

CursorON

Fungsi : Menampilkan kursor

Input : -

Output : -

Remark : Apabila rutin **ClrDisMem** dijalankan , maka pada tampilan akan tampak kursor yang berkedip

CursorOFF

Fungsi : Mematikan kursor

Input : -

Output : -

Remark : -

RealDis

Fungsi : Menset memori display sebesar 6 byte sesuai dengan jumlah 7 Segment pada tampilan.

Input : -

Output : -

Remark : Jika anda memakai rutin **RealDis** (real display), maka karakter maksimum yang dapat Anda tuliskan hanya 6, penulisan karakter yang ke 7 akan menyebabkan seluruh tampilan terhapus dan ditampilkan sebagai karakter ke 1.

VirtualDis

Fungsi : Menset memori display sebesar 18 byte sesuai dengan 3 x jumlah 7 Segment (6 digit) pada tampilan.

Input : -

Output : -

Remark : Jika Anda memakai **VirtualDis** (virtual display), maka seakan-akan Anda memiliki tampilan sebanyak 18 karakter, penulisan karakter yang ke 7 akan menggeser tampilan sebelumnya ke kiri 1 kali demikian seterusnya sampai penulisan karakter yang ke 18.

ShftDisRight

Fungsi : Menggeser tampilan ke kanan satu kali (tanpa mengubah memori display).

Input : -

Output : -

Remark : Rutin ini hanya berlaku jika display diset pada kondisi virtual.

ShftDisLeft

Fungsi : Menggeser tampilan ke kiri satu kali (tanpa mengubah memori display).

Input : -

Output : -

Remark : Rutin ini hanya berlaku jika display diset pada kondisi virtual.

EnbKeyInt

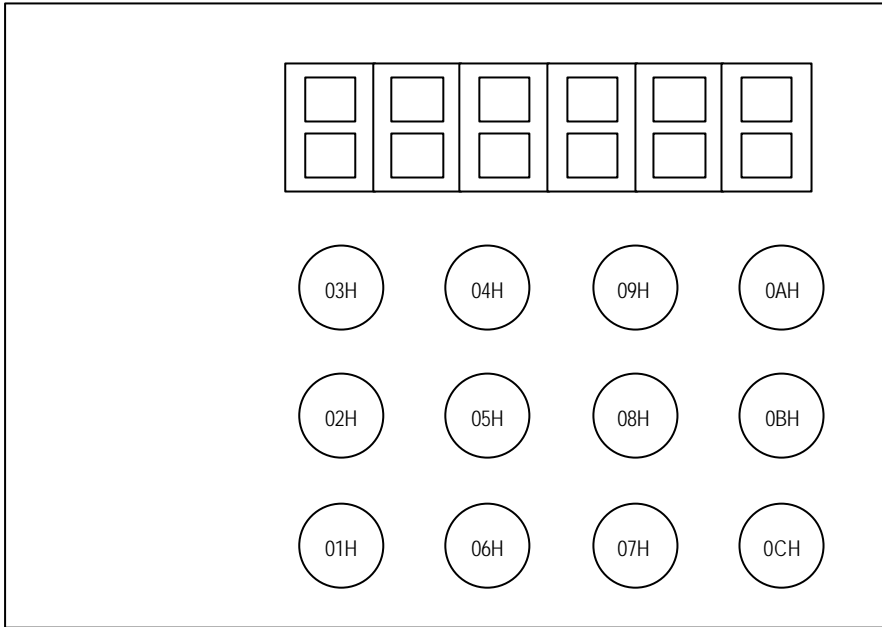
Fungsi : Mengaktifkan External Interrupt 1 agar menghasilkan sinyal interrupt setiap kali ada keypad yang ditekan.

Input : -

Output : -

Remark : Dengan mengaktifkan keypad interrupt Anda tidak perlu melakukan polling untuk mengetahui apakah ada tombol yang ditekan, yang perlu Anda siapkan adalah interrupt service routine untuk external interrupt 1 (lihat contoh program sample2.asm), Kode tombol yang ditekan akan berada pada variabel KeyCode.

Kode (heksadesimal) masing-masing tombol keypad ditunjukkan pada gambar berikut :



DisKeyInt

- Fungsi : Mematikan External Interrupt 1 (kondisi default) sebagai keypressed interrupt
- Input : -
- Output : -
- Remark : Untuk mengetahui apakah ada tombol keypad yang ditekan Anda harus melakukan polling pada variabel KeyPressed, jika berlogika “1” maka ada tombol yang ditekan, kode tombol yang ditekan akan berada pada variabel KeyCode (lihat contoh sample1.asm).

B. KNDINT.ASM

Modul ini menggunakan memori internal 89C51 alamat 40H-53H sebagai memori display, modul ini cocok digunakan bila pada memori internal 89C51 masih tersedia cukup ruang (biasanya dipakai untuk stack memory dan variabel). Kelebihan dari modul ini adalah kecepatan prosesnya, sehingga tidak menimbulkan blink/kedip pada 7 Segment Display. Resources DT51 yang dipakai (otomatis tidak dapat dipakai untuk keperluan lain) adalah

- Memori internal dengan alamat 18H-20H.
- Memori internal dengan alamat 40H-53H (perhatikan bahwa SP minimum diset pada address 53H).
- Timer 0 dan interuptnya.
- PPI 8255

Adapun rutin-rutin untuk modul ini persis sama dengan rutin-rutin pada KNDEXT.ASM seperti yang telah dijelaskan sebelumnya.

3.2 Contoh Program

Berikut ini beberapa contoh program untuk DT51 KND (diasumsikan menggunakan Assembler ALDS[®]).

SAMPLE1.ASM :

Contoh program menggunakan KNDEXT.ASM tanpa interupt.

```
-----  
;Sample Program Using DT51 KND Routine  
; Without External Interrupt 1  
-----  
                .CODE  
                ORG    4000H  
                LJMP   Start  
  
                ORG    400BH  
                LJMP   Scanning  
  
                ORG    4100H  
                INCLUDE KNDEXT.ASM
```

JMPTABLE:	LJMP	S1Act
	LJMP	S2Act
	LJMP	S3Act
	LJMP	S4Act
	LJMP	S5Act
	LJMP	S6Act
	LJMP	S7Act
	LJMP	S8Act
	LJMP	S9Act
	LJMP	S10Act
	LJMP	S11Act
	LJMP	S12Act
S1Act:	MOV	A,#0
	ACALL	Write7S
	LJMP	Loop
S2Act:	MOV	A,#1
	ACALL	Write7S
	LJMP	Loop
S3Act:	MOV	A,#2
	ACALL	Write7S
	LJMP	Loop
S4Act:	MOV	A,#3
	ACALL	Write7S
	LJMP	Loop
S5Act:	MOV	A,#4
	ACALL	Write7S
	LJMP	Loop
S6Act:	MOV	A,#5
	ACALL	Write7S
	LJMP	Loop
S7Act:	MOV	A,#6
	ACALL	Write7S
	LJMP	Loop
S8Act:	MOV	A,#7
	ACALL	Write7S
	LJMP	Loop
S9Act:	MOV	A,#8
	ACALL	Write7S
	LJMP	Loop

```

S10Act:    MOV     A,#9
           ACALL  Write7S
           LJMP   Loop
S11Act:    ACALL  ShftDisRight
           LJMP   Loop
S12Act:    ACALL  ShftDisLeft
           LJMP   Loop

Start:     MOV     P1,#0FBH
           MOV     SP,#40H
           LCALL  InitKND
           LCALL  CursorOn
           LCALL  VirtualDis
           LCALL  ClrDisMem

Loop:      JNB     KeyPressed,$
           CLR     KeyPressed
           MOV     A,KeyCode
           MOV     DPTR,#JMPTABLE
           DEC     A
           MOV     B,#3
           MUL    AB
           JMP     @A+DPTR
           END

```

SAMPLE2.ASM :

Contoh program menggunakan KNDEXT.ASM dengan interrupt.

```

;-----
;Sample Program Using DT51 KND Routine
;   With External Interrupt 1
;-----
           .CODE
           ORG     4000H
           LJMP   Start

           ORG     400BH
           LJMP   Scanning
           ORG     4013H
           LJMP   ISR_EI1

```

```

                ORG     4100H
                INCLUDE KNDEXT.ASM

ISR_EI1:       MOV     A,KeyCode
                MOV     DPTR,#JMPTABLE
                DEC     A
                MOV     B,#3
                MUL     AB
                JMP     @A+DPTR

JMPTABLE:      LJMP    S1Act
                LJMP    S2Act
                LJMP    S3Act
                LJMP    S4Act
                LJMP    S5Act
                LJMP    S6Act
                LJMP    S7Act
                LJMP    S8Act
                LJMP    S9Act
                LJMP    S10Act
                LJMP    S11Act
                LJMP    S12Act

S1Act:         MOV     A,#0
                ACALL  Write7S
                RETI

S2Act:         MOV     A,#1
                ACALL  Write7S
                RETI

S3Act:         MOV     A,#2
                ACALL  Write7S
                RETI

S4Act:         MOV     A,#3
                ACALL  Write7S
                RETI

S5Act:         MOV     A,#4
                ACALL  Write7S
                RETI

S6Act:         MOV     A,#5
                ACALL  Write7S
                RETI

```

```

S7Act:    MOV     A,#6
          ACALL  Write7S
          RETI

S8Act:    MOV     A,#7
          ACALL  Write7S
          RETI

S9Act:    MOV     A,#8
          ACALL  Write7S
          RETI

S10Act:   MOV     A,#9
          ACALL  Write7S
          RETI

S11Act:   ACALL  ShftDisRight
          RETI

S12Act:   ACALL  ShftDisLeft
          RETI

Start:    MOV     P1,#0FBH
          MOV     SP,#40H
          LCALL  InitKND
          LCALL  CursorOn
          LCALL  VirtualDis
          LCALL  ClrDisMem
          LCALL  EnbKeyInt
          AJMP   $

          END

```

SAMPLE3.ASM :

Contoh program menggunakan KNDINT.ASM tanpa interupt.

```

;-----
;Sample Program Using DT51 KND Routine
;  Without External Interrupt 1
;-----
          .CODE
          ORG     4000H
          LJMP   Start

          ORG     400BH
          LJMP   Scanning

```

```
ORG 4100H
INCLUDE KNDINT.ASM
```

```
JMPTABLE: LJMP S1Act
           LJMP S2Act
           LJMP S3Act
           LJMP S4Act
           LJMP S5Act
           LJMP S6Act
           LJMP S7Act
           LJMP S8Act
           LJMP S9Act
           LJMP S10Act
           LJMP S11Act
           LJMP S12Act

S1Act:    MOV A,#0
           ACALL Write7S
           LJMP Loop
S2Act:    MOV A,#1
           ACALL Write7S
           LJMP Loop
S3Act:    MOV A,#2
           ACALL Write7S
           LJMP Loop
S4Act:    MOV A,#3
           ACALL Write7S
           LJMP Loop
S5Act:    MOV A,#4
           ACALL Write7S
           LJMP Loop
S6Act:    MOV A,#5
           ACALL Write7S
           LJMP Loop
S7Act:    MOV A,#6
           ACALL Write7S
           LJMP Loop
S8Act:    MOV A,#7
           ACALL Write7S
           LJMP Loop
S9Act:    MOV A,#8
           ACALL Write7S
           LJMP Loop
```

```

S10Act:    MOV     A,#9
           ACALL  Write7S
           LJMP   Loop
S11Act:    ACALL  ShftDisRight
           LJMP   Loop
S12Act:    ACALL  ShftDisLeft
           LJMP   Loop

Start:     MOV     P1,#0FBH
           MOV     SP,#53H
           LCALL  InitKND
           LCALL  CursorOn
           LCALL  VirtualDis
           LCALL  ClrDisMem

Loop:      JNB     KeyPressed,$
           CLR     KeyPressed
           MOV     A,KeyCode
           MOV     DPTR,#JMPTABLE
           DEC     A
           MOV     B,#3
           MUL    AB
           JMP     @A+DPTR
           END

```

SAMPLE4.ASM :

Contoh program menggunakan KNDINT.ASM dengan interrupt.

```

;-----
;Sample Program Using DT51 KND Routine
;   With External Interrupt 1
;-----
           .CODE
           ORG    4000H
           LJMP   Start

           ORG    400BH
           LJMP   Scanning
           ORG    4013H
           LJMP   ISR_EI1

```

```

                ORG     4100H
                INCLUDE KNDINT.ASM

ISR_EI1:       MOV     A,KeyCode
                MOV     DPTR,#JMPTABLE
                DEC     A
                MOV     B,#3
                MUL     AB
                JMP     @A+DPTR

JMPTABLE:     LJMP    S1Act
                LJMP    S2Act
                LJMP    S3Act
                LJMP    S4Act
                LJMP    S5Act
                LJMP    S6Act
                LJMP    S7Act
                LJMP    S8Act
                LJMP    S9Act
                LJMP    S10Act
                LJMP    S11Act
                LJMP    S12Act

S1Act:        MOV     A,#0
                ACALL  Write7S
                RETI

S2Act:        MOV     A,#1
                ACALL  Write7S
                RETI

S3Act:        MOV     A,#2
                ACALL  Write7S
                RETI

S4Act:        MOV     A,#3
                ACALL  Write7S
                RETI

S5Act:        MOV     A,#4
                ACALL  Write7S
                RETI

S6Act:        MOV     A,#5
                ACALL  Write7S
                RETI

```

```

S7Act:    MOV    A,#6
          ACALL  Write7S
          RETI
S8Act:    MOV    A,#7
          ACALL  Write7S
          RETI
S9Act:    MOV    A,#8
          ACALL  Write7S
          RETI
S10Act:   MOV    A,#9
          ACALL  Write7S
          RETI
S11Act:   ACALL  ShftDisRight
          RETI
S12Act:   ACALL  ShftDisLeft
          RETI

Start:    MOV    P1,#0FBH
          MOV    SP,#53H
          LCALL  InitKND
          LCALL  CursorOn
          LCALL  VirtualDis
          LCALL  ClrDisMem
          LCALL  EnbKeyInt
          AJMP   $

          END

```

Bila Anda menemui kesulitan dalam menggunakan DT51 KND, hubungi technical support kami melalui e-mail :

 tech-sup@mitra.net.id

