

Contoh ADT Sederhana dalam Bahasa C

IF2110 – Algoritma dan Struktur Data
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Tujuan

Mahasiswa dapat mengimplementasikan ADT dengan memakai modul program dalam bahasa C

File time.h

```
/* File : time.h */
/* spesifikasi ADT Time */
#ifndef TIME_h
#define TIME_h
#include "boolean.h"

/* Notasi Akses */
#define HOURS(t)    (t).hours
#define MINUTES(t) (t).minutes
#define SECONDS(t) (t).seconds
```

```
typedef struct Time {
    int hours;    /* 0..23 */
    int minutes; /* 0..59 */
    int seconds; /* 0..59 */
} time;
```

```
/****** Konstruktor *****/
void CreateTime(time *t, int h, int m, int s);
/* Membentuk time t dari h, m, s yg valid */
```

Notasi akses
dengan macro:
Menggantikan
fungsi/prosedur
selektor **get** dan **set**

File time.h (lanjutan)

```
/*-----validator time-----*/
boolean isTimeValid(int h, int m, int s);
/* Mengirim true jika h,m,s dapat membentuk time yang valid */

/*-----Baca & Tulis time-----*/
void readTime(time *t);
/* I.s. : t tak terdefinisi */
/* F.s. : t terdefinisi dan merupakan time valid */
/* Proses : mengulang baca komponen h,m,s sehingga membentuk t yang valid */
void displayTime(time t);
/* I.s. : t sembarang */
/* F.s. : t ditulis ke layar dengan format HH:MM:SS */
/* Proses : Menulis ke layar */

/*-----Konversi terhadap type jam-----*/
int timeToSeconds(time t);
/* Konversi time menjadi detik */
time secondsToTime(int n);
/* Konversi detik ke time */
#endif
```

File time.c

```
/* File : time.c */
/* Body ADT Time */
#include "time.h"
/* -----konstruktor Time----- */
void CreateTime(time *t, int h, int m, int s)
/* Membentuk time dari h, m, s yg valid */
{
    /* Kamus Lokal */
    /* Algoritma */
    HOURS(*t) = h;
    MINUTES(*t) = m;
    SECONDS(*t) = s;
}
/*-----validator Time-----*/
boolean isTimeValid(int h, int m, int s)
/* Mengirim true jika h,m,s dapat membentuk time yang valid */
{
    /* Kamus Lokal */
    /* Algoritma */
    return ((h >= 0 && h <= 23) && (m >= 0 && m <= 59) && (s >= 0 && s <= 59));
}
```

Cara mengeset komponen,
menggunakan notasi akses
yang sudah didefinisikan

File time.c (lanjutan)

```
/*-----Baca & Tulis time-----*/
void readTime(time *t)
/* (komentar tidak ditulis untuk menghemat tempat) */
{   /* Kamus Lokal */
    int h, m, s;
    /* Algoritma */
    do {
        printf("Masukkan time : ");scanf("%d",&h);
        printf("Masukkan menit : ");scanf("%d",&m);
        printf("Masukkan detik : ");scanf("%d",&s);
    } while(!isTimeValid(h,m,s));
    CreateTime(t, h,m,s);
}
void displayTime(time t)
/* (komentar tidak ditulis untuk menghemat tempat) */
{   /* Kamus lokal */
    /* Algoritma */
    printf("%02d:%02d:%02d",HOURS(t),MINUTES(t),SECONDS(t));
}
```

File time.c (lanjutan)

```
int timeToSeconds(time t)
/* Konversi time menjadi detik */
{
    /* Kamus Lokal */
    /* Algoritma */
    return (3600 * HOURS(t) + 60 * MINUTES(t) + SECONDS(t));
}

time secondsToTime(int n)
/* Konversi detik ke time */
{
    /* Kamus Lokal */
    int h,m,s,sisa; time t1;
    /* Algoritma */
    n = n % 86400; // pastikan n detik berada dalam rentang 24 jam
    h = n / 3600;
    sisa = n % 3600;
    m = sisa / 60;
    s = sisa % 60;
    CreateTime(&t1, h,m,s);
    return t1;
}
```

Driver

```
/* File : main_time.c */
/* Program MainTime */
/* Driver ADT Time */
#include <stdio.h>
#include "time.h"

int main()
{
    /* KAMUS */
    time t1, t2;
    /* ALGORITMA */
    readTime(&t1);
    printf("time yang dibaca: ");
    displayTime(t1); printf("\n");
    printf("Konversi menjadi detik: %d\n",timeToSeconds(t1));
    t2 = secondsToTime(5000);
    printf("5000 detik sama dengan time : ");
    displayTime(t2);

    return 0;
}
```