9

FUNGSI TIME, DATE DAN FUNGSI-FUNGSI YANG BERHUBUNGAN DENGAN SISTEM
Bagian ini mencakup fungsi-fungsi di atas yang dalam kodisi tertentu akan lebih sensitif bagi sistem operasi dari pada yang lain. Fungsi ini mencakup fungsi `time` dan `date` serta fungsi-fungsi yang berhubungan dengan lokasi geografis tempat komputer digunakan.


```c
struct tm {
    int tm_sec; /* seconds, 0 - 59 */
    int tm_min; /* minutes, 0 - 59 */
    int tm_hour; /* hours, 0 - 23 */
    int tm_mday; /* day of the month, 1 - 31 */
    int tm_mon; /* months since Jan, 0 - 11 */
    int tm_year; /* years from 1900 */
    int tm_wday; /* since Sundah, 0 - 6 */
    int tm_yday; /* days since Jan 1, 0 - 365 */
    int tm_isdst /* Daylight Savings Time indicator */
}
```

Nilai dari `tm_isdst` akan positif bila waktu simpan daylight aktif, 0 bila dia tidak aktif, dan negatif bila tidak terdapat informasi. Standard ANSI C menunjuk bentuk waktu dan tanggal ini sebagai `broken-down time`.

Disamping itu, `TIME.H` mendefinisikan makro `CLK TCK` yang merupakan jumlah detik jam sistem tiap detik.

Fungsi-fungsi lokasi geografis memerlukan header `LOCALE.H`.

Sebagian besar compiler juga akan memberikann fungsi-fungsi sistem operasi dan fungsi-fungsi khusus komputer; baca kembali buku manual user untuk jenis-jenis fungsi ini.
**ASCTIME**

```c
#include "time.h"
char *asctime(struct tm *ptr);
```

Fungsi `asctime()` akan mengembalikan pointer ke string. String ini akan mengubah informasi yang disimpan dalam struktur yang ditunjuk oleh `ptr` kedalam bentuk sebagai berikut:

```
day month date hours:minutes:seconds year
```

Misalnya:

**Wed Jun 19 12:05:34 1989**

Pointer struktur yang melewati `asctime()` biasanya didapat dari ` localtime` atau `gmtime()`.

Buffer yang digunakan oleh `asctime()` untuk menampung string output yang sudah diformat merupakan array yang dialokasikan secara statistik dan akan ditulis ulang tiap kali fungsi ini dipanggil. Bila kita hendak menyimpan isi string ini, kita harus mencopynya di tempat lain.

Fungsi-fungsi yang berhubungan adalah `localtime()`, `gmtime()`, `time()`, dan `ctime()`.

---

**CLOCK**

```c
#include "time.h"
clock_t clock(void);
```

Fungsi `clock()` akan mengembalikan nilai yang menunjukkan jumlah lamanya program pemanggilan dijalankan. Untuk mengubah nilai ini ke dalam detik, maka bagiilah dengan `CLK_TCK`. Nilai -1 akan dikembalikan bila waktu ini tidak ada.

Fungsi-fungsi yang berhubungan adalah `asctime()`, `time()`, dan `ctime()`.
CTIME

#include "time.h"

char *ctime(const time_t *time);

Fungsi ctime() akan mengembalikan pointer ke dalam string dari bentuk:

day month year hours:minutes:seconds year

pointer ke dalam kalender waktu. Waktu kalender biasanya diperoleh melalui panggilan time().

Buffer yang digunakan ctime() untuk menampung string output merupakan array karakter yang dialokasikan secara statis, dan ditulis ulang tiap kali fungsi tersebut dipanggil. Bila kita ingin menyimpan isi string, kita harus mencopypnya di tempat lain.

Fungsi-fungsi yang berhubungan adalah localtime(), gmtime(), time(), dan asctime().

DIFFTIME

#include "time.h"

double difftime(time_t time2, time_t time);

Fungsi difftime() akan mengembalikan perbedaan antara time1 dan time2, dalam detik. Yaitu time2 - time1.

Fungsi-fungsi yang berhubungan adalah localtime(), gmtime(), time(), dan asctime().
**GMTIME**

```c
#include "time.h"
struct tm *gmtime(time_t *time);
```

Fungsi `gmtime()` akan mengembalikan pointer ke dalam bentuk broken-down dari `time` dalam bentuk struktur `tm`. Waktu ini disajikan dalam Greenwich mean time. Nilai `time` biasanya diperoleh melalui panggilan `time()`.

Struktur yang digunakan oleh `gmtime()` untuk menampung waktu broken-down secara statistik dialokasikan dan ditulis ulang tiap kali fungsi tersebut dipanggil. Bila kita hendak menyimpan isi struktur ini, maka kita harus mengcopynya di tempat lain.

Fungsi-fungsi yang berhubungan adalah `localtime()`, `time()`, dan `asctime()`.

**LOCALECONV**

```c
#include "locale.h"
struct lconv *localeconv(void);
```

Fungsi `localeconv()` akan mengembalikan pointer ke struktur tipe `lconv` yang berisikan berbagai informasi lingkungan negara tertentu yang berhubungan dengan cara memformat bilangan. Struktur `lconv` diorganisaskan sebagai berikut.

```c
struct lconv {
    char *decimal_point; /* decimal point character for non-monetary values */
    char *thousands_sep; /* thousands separator for non-monetary values */
    char *grouping; /* specifies grouping for non-monetary values */
    char int_curr_symbol; /* international currency symbol */
    char *currency_symbol; /* local currency symbol */
};
```
char *mon_decimal_point; /* decimal point character for monetary values */
char *mon_thousands_sep; /* thousands separator for monetary values */
char *mon_grouping_sign; /* specifies grouping for monetary values */
char *positive_sign; /* positive value indicator for monetary values */
char *negative_sign; /* negative value indicator for monetary values */
char int_frac_digits; /* number of digits displayed to the right of the decimal point for monetary values displayed using international format */
char frac_digits; /* number of digits displayed to the right of the decimal point for monetary values displayed using local format */
char p_cs_precedes; /* 1 if currency symbol precedes positive value, 0 if currency symbol follows value */
char p_sep_by_space; /* 1 if currency symbol is separated from value by a space, 0 otherwise */
char n_cs_precedes; /* 1 if currency symbol precedes a negative value, 0 if currency symbol follows value */
char n_sep_by_space; /* 1 if currency symbol is separated from a negative value by a space, 0 if currency symbol follows value */
char p_sign_posn; /* indicates position of positive value symbol */
char n_sign_posn; /* indicates position of negative value symbol */
Fungsi `localeconv()` akan mengembalikan pointer ke dalam stuktur `conv`. Kita tidak boleh mengubah isi dari struktur ini.

Baca kembali manual compiler untuk informasi implementasi khusus yang berhubungan dengan fungsi ini.

Fungsi yang berhubungan adalah `setlocale()`.

### LOCALTIME

```c
#include "time.h"

struct tm *localtime(const time_t *time);
```

Fungsi `localtime()` akan mengembalikan pointer ke dalam bentuk broken-down (bagian-bagian) `time` dalam susunan `tm`. Waktu disajikan dalam waktu lokal. Nilai `time` biasanya didapatkan melalui panggilan `time()`.

Struktur yang digunakan oleh `localtime()` untuk menampung bagian-bagian waktu secara statistik dialokasikan dan ditulis ulang tiap kali fungsi ini dipanggil. Bila kita hendak menyimpan isi struktur ini, maka kita harus mencopypnya di tempat lain.

Fungsi-fungsi yang berhubungan adalah `gmtime()`, `time()`, dan `asctime()`.

### MKTIME

```c
#include "time.h"

time_t mktime(struct tm *time);
```

Fungsi dari `mktime()` akan mengembalikan persamaan waktu kalender dari bagian-bagian waktu yang ditemukan dalam struktur yang ditunjuk oleh `time`. Fungsi ini terutama digunakan untuk memulai waktu sistem. Unsur `tm_wday` dan `tm_yday` diset oleh fungsi ini sehingga mereka tidak perlu didefinisikan pada saat pemanggilan.

Bila `mktime()` tidak dapat menyajikan informasi sebagai waktu kalender yang benar, maka `-1` akan dikembalikan.

Fungsi-fungsi yang berhubungan adalah `gmtime()`, `time()`, `asctime` dan `ctime()`.
SETLOCALE

#include "locale.h"

char *setlocale(int type, const char *locale);

Fungsi setlocale() memungkinkan parameter tertentu yang menunjukkan lokasi geografis dari pelaksanaan program untuk diset atau ditanyakan. Di Eropa, misalnya, nilai uang menggunakan koma, sedangkan di Amerika Serikat digunakan tanda titik.

Bila locale adalah nol, maka setlocale() akan mengembalikan pointer ke dalam string lokalisasi yang sedang aktif. Bila yang terjadi sebaliknya maka usahakanlah setlocale() untuk menggunakan string lokalisasi yang ditunjuk untuk menyet parameter locale seperti yang ditunjuk oleh type.

Pada saat pemanggilan, type harus merupakan salah satu dari makro berikut:

LC_ALL
LC_COLLATE
LC_CTYPE
LC_MONETARY
LC_NUMERIC
LC_TIME

LC_ALL menunjuk pada semua kategori lokalisasi. LC_COLLATE akan mempengaruhi operasi fungsi strftime(). LC_CTYPE mengubah cara fungsi karakter bekerja. LC_MONETARY akan menentukan format moneter. LC_NUMERIC akan mengubah karakter desimal-poin untuk fungsi input/output yang diformat. Akhirnya LC_TIME menentukan fungsi strftime().

Standard ANSI C mendefinisikan dua string yang mungkin untuk locale. Yang pertama adalah “C”, yang menentukan lingkungan minimal untuk kompilasi C. Yang kedua adalah “”, string nol yang menentukan lingkungan default yang ditentukan dalam implementasi. Semua nilai lain untuk locale merupakan nilai yang ditentukan implementasi dan akan mempengaruhi portabilitas.

Fungsi setlocale() akan mengembalikan pointer ke dalam string yang berhubungan dengan parameter type.

Fungsi-fungsi yang berhubungan adalah localeconv(), time(), strftime(), dan strftime().
Fungsi `strftime()` adalah untuk menempatkan waktu, tanggal dan informasi lain ke dalam string yang ditunjuk oleh `str` menurut command format yang ditemukan dalam string yang ditunjuk oleh `fmt`, dengan menggunakan bagian waktu `time`. Maksimum dari karakter `maxsize` akan ditempatkan dalam `str`.

Fungsi ini bekerja hampir seperti `printf()` dalam hal bahwa fungsi ini mengenali satu set command yang dimulai dengan tanda persen (%) dan menempatkan outputnya yang telah diformat ke dalam string. Command format digunakan untuk menunjuk cara yang tepat tentang penyajian berbagai informasi tanggal dan waktu dalam `str`. Karakter lainnya yang ditemukan dalam string format ini ditempatkan dalam `str` dengan tidak mengalami perubahan. Waktu dan tanggal yang ditampilkan adalah waktu setempat. Command format ini ditunjukkan dalam tabel berikut. Perhatikan bahwa banyak command yang case-sensitive.

<table>
<thead>
<tr>
<th>Command</th>
<th>Diganti oleh</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>%a</code></td>
<td>Nama hari-hari kerja yang disingkat</td>
</tr>
<tr>
<td><code>%A</code></td>
<td>Nama hari-hari dalam satu minggu penuh</td>
</tr>
<tr>
<td><code>%b</code></td>
<td>Nama bulan yang disingkat</td>
</tr>
<tr>
<td><code>%B</code></td>
<td>Nama bulan lengkap</td>
</tr>
<tr>
<td><code>%c</code></td>
<td>String waktu dan tanggal yang standard</td>
</tr>
<tr>
<td><code>%d</code></td>
<td>Hari dalam bulan dalam desimal (1-31)</td>
</tr>
<tr>
<td><code>%H</code></td>
<td>Jam (0-23)</td>
</tr>
<tr>
<td><code>%I</code></td>
<td>Jam (1-12)</td>
</tr>
<tr>
<td><code>%j</code></td>
<td>Hari dalam tahun dengan desimal (1-366)</td>
</tr>
<tr>
<td><code>%m</code></td>
<td>Bulan dalam desimal (1-12)</td>
</tr>
<tr>
<td><code>%M</code></td>
<td>Menit dalam desimal (0-59)</td>
</tr>
<tr>
<td><code>%p</code></td>
<td>Persamaan lokal dari AM atau PM</td>
</tr>
<tr>
<td><code>%S</code></td>
<td>Detik dalam desimal (0-59)</td>
</tr>
<tr>
<td><code>%u</code></td>
<td>Minggu dalam tahun, Hari Minggu sebagai hari pertama (0-52)</td>
</tr>
<tr>
<td>%w</td>
<td>Hari-hari kerja dalam desimal (0-6, hari Minggu adalah 0)</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>%W</td>
<td>Minggu dalam tahun, hari Minggu sebagai hari pertama (0-52)</td>
</tr>
<tr>
<td>%x</td>
<td>String tanggal standard</td>
</tr>
<tr>
<td>%X</td>
<td>String waktu standard</td>
</tr>
<tr>
<td>%y</td>
<td>Tahun dalam desimal tanpa abab (00-99)</td>
</tr>
<tr>
<td>%Y</td>
<td>Tahun termasuk abad dalam desimal</td>
</tr>
<tr>
<td>%Z</td>
<td>Nama daerah waktu</td>
</tr>
<tr>
<td>%%</td>
<td>Tanda persen</td>
</tr>
</tbody>
</table>

Fungsi `strftime()` akan mengembalikan jumlah karakter yang ditempatkan dalam string yang ditunjukkan oleh `str`, atau 0 bila terjadi kesalahan.

Fungsi-fungsi yang berhubungan adalah `localtime()`, `gmtime()`, dan `ctime()`.

### TIME

```c
#include "time.h"

time_t time(time_t *time);
```

Fungsi `time()` adalah untuk mengembalikan waktu kalender yang sedang berlaku dari sistem tersebut. Bila sistem ini tidak memiliki waktu, maka -1 akan dikembalikan.

Fungsi `time()` dapat dipanggil baik dengan pointer nol atau dengan pointer untuk variabel dari jenis `time_t`. Bila pointer yang kedua ini digunakan maka argumen juga akan disi dengan waktu kalender.

Fungsi-fungsi yang berhubungan adalah `localtime()`, `gmtime()`, `strftime()`, dan `ctime()`.