

# Modularitas Program dalam Bahasa C

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

# Tujuan

Mahasiswa memahami kegunaan modul program

Mahasiswa memahami konsep reusability dalam pembuatan program

Mahasiswa memahami pembuatan program C dengan beberapa modul program (dalam beberapa file)

Mahasiswa dapat mengimplementasikan program dengan memakai modul program dalam bahasa C

# Modularitas Program

Sebuah program yang “utuh”, seringkali terdiri dari beberapa modul program.

Modul program dapat mewakili:

- Sekumpulan **rutin** (prosedur & fungsi) sejenis
- **ADT** (Abstract Data Type): definisi type dan primitifnya
- **Mesin** : definisi *state variable* dari mesin dan primitifnya

# Pembuatan program

Program terdiri dari :

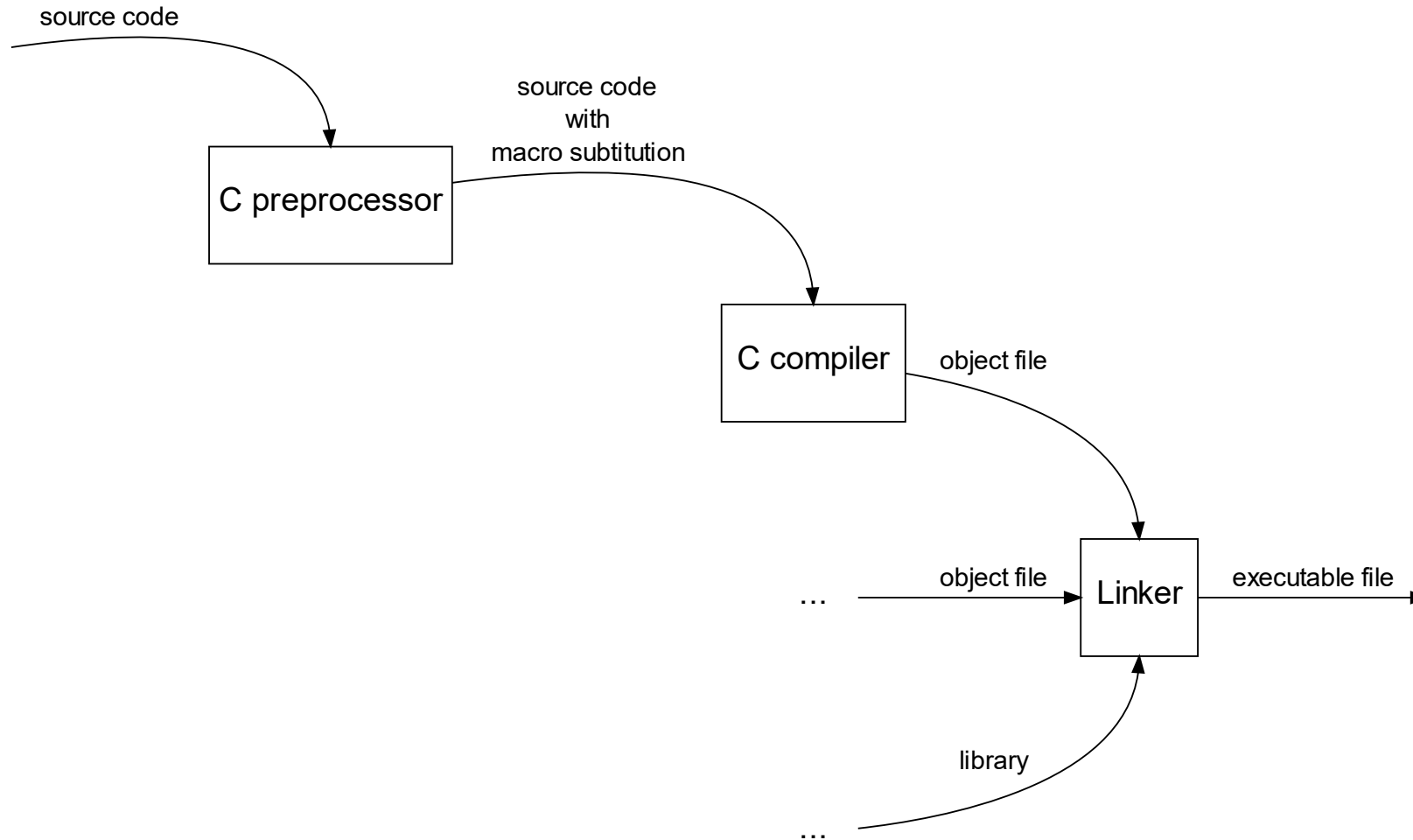
- Satu program utama (*main program*)
- Beberapa modul yang lain

Program yang dibagi-bagi menjadi beberapa file seharusnya dapat dikompilasi terpisah. Setiap modul membentuk sebuah *object code*.

Pembuatan sebuah *executable code* dilakukan dgn melakukan *linking* terhadap sejumlah *object code* yang sudah dikompilasi.

Penghematan waktu dan duplikasi usaha (*reusability*).

# Pemrosesan kode sumber dalam Bahasa C



# Modul program dalam C (1/3)

Program utuh terdiri dari 4 kelompok file

1. File *header* dengan nama `xxx.h` di folder `src/`

Untuk setiap type dan primitifnya, ada sebuah file *header*.

Contoh: untuk ADT Time dan ADT Date ada 2 buah file header, yaitu `time.h` dan `date.h`

Fungsi selektor (`get*`, `set*`) dapat digantikan dengan macro berparameter. Misalnya untuk selektor `HOURS(t)`, `MINUTES(t)`, dan `SECONDS(t)` dituliskan sebagai:

```
#define HOURS(t) (t).hours  
#define MINUTES(t) (t).minutes  
#define SECONDS(t) (t).seconds
```

# Modul program dalam C (2/3)

2. File body dengan nama `xxx.c` di folder `src/`

Berisi realisasi dari prototype yang didefinisikan dalam file *header*.

Akan ada sebuah `xxx.c` untuk setiap `xxx.h`

Contoh : untuk file header `Time.h` dan `Date.h` akan ada file body `Time.c` dan `Date.c`

# Modul program dalam C (3/3)

## 3. File main (driver) di folder `src/`

Berisi program utama dan prosedur/fungsi lain yang hanya dibutuhkan oleh main

Misalnya diberi nama `main.c`

## 4. File unit test di folder `tests/`

Berisi beberapa *test case* untuk setiap prosedur/fungsi yang ada di header

Misalnya, menggunakan library check, diberi nama `check_time.c`

Program Uteh akan terdiri dari sebuah `main.c`, sebuah `xxx.h`, `xxx.c`, dan `check_xxx.c`



# File Header

```
/* File : xxx.h */
/* Deskripsi : keterangan isi file header */
/* Isi : deklarasi konstanta, type dan prototype */
/* File header TIDAK BOLEH mengandung deklarasi variabel! */

#ifndef xxx_h
#define xxx_h
/* Bagian I : berisi deklarasi konstanta */

/* Bagian II : berisi deklarasi type */

/* Bagian III : berisi deklarasi prototype prosedur & fungsi */
/* yg merupakan primitif type tsb */
/* Kelompokkan fungsi dan prosedur sesuai standar di kelas */
/* Mis.: konstruktor, selektor, predikat, operator relasional*/
/* operator aritmatika, operator lain, dsb */

#endif
```

# File Body

```
/* File : xxx.c */  
/* Deskripsi : keterangan isi file body */  
/* Isi : realisasi/ kode program dari semua prototype */  
/* yg didefinisikan pada xxx.h */  
/* Untuk sebuah mesin akan mengandung deklarasi variabel */  
/* state dari mesin tsb */  
  
# include "xxx.h"  
  
/* Realisasi kode program, sesuai urutan pada xxx.h */  
  
/* Copy dari xxx.h, kemudian edit */
```

# File Main Program

```
/* File : main_xxx.c */
/* Deskripsi: program utama & semua nama lokal thd persoalan */

#include "xxx.h"
/* include file lain yg diperlukan */
/* Bagian I : berisi kamus GLOBAL dan prototype */
/* deklarasi semua nama dan prosedur/fungsi global */

/* Bagian II : program utama */
int main() {
    /* Kamus lokal terhadap main */

    /* Algoritma */

    return 0;
}

/* Bagian III : berisi realisasi kode program yang merupakan */
/* BODY dari semua prototype yg didefinisikan pada file ini */
/* yaitu pada bagian I, dengan urutan-urutan yang sama */
/* Copy prototype, kemudian edit */
```

# File Unit Test

```
/* File : check_xxx.c */
/* Deskripsi: unit test untuk modul xxx */

#include <check.h>
#include "../src/xxx.h"
/* include file lain yg diperlukan */

START_TEST(test_xxx_namafungsiyangdittest) {
    /* Bagian I: berisi test untuk sebuah fungsi dari modul xxx */
} END_TEST

Suite *xxx_suite(void) {
    /* Bagian II: mengumpulkan semua test menjadi satu test suite */
}

int main(void) {
    /* Bagian III: menjalankan unit test */
}
```

# Penyimpanan Modul Program

Untuk setiap modul xxx.h dan xxx.c dibuat unit test untuk menguji setiap fungsi/prosedur yg dibuat.

Setiap paket yg terdiri dari xxx.h, xxx.c, unit test (misal: check\_xxx.c), dan hasil test disimpan dalam satu direktori.

**Pelajari** [Check 0.15.2: 3 Tutorial: Basic Unit Testing](#).

