

Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2023-2024 / Genap
Modul : 2 – Pointers and Functions
Nama Asisten / NIM : Isnaini Azhar Ramadhan Wijaya / 18321016
Nama Praktikan / NIM : Pradigta Hisyam Ramadhan / 18322008

BCL / Log Praktikum

Source code

```
/*EL2208 Praktikum Pemecahan Masalah dengan C 2023/2024
*Modul          : 2
*Hari dan Tanggal : Jumat, 8 Maret 2024
*Nama (NIM)      : Pradigta Hisyam Ramadhan (18322008)
*Asisten (NIM)   : Isnaini Azhar Ramadhan (18321016)
*Nama File       : main.c
*Deskripsi       : Program digunakan untuk menampilkan waktu istirahat dalam
                   format jam dan menit
*/

#include <stdio.h>
#include <stdlib.h>

// Fungsi untuk memeriksa apakah waktu masuk atau keluar valid
int isValidTime(int jMasuk, int mMasuk, int jKeluar, int mKeluar, char tanda) {
    if(jMasuk < 24 && jKeluar < 24 && mMasuk <= 60 && mKeluar <= 60 && tanda ==
    ':'){ // Masukkan valid
        return 1;
    }
    else{
        return 0;
    }
}

// Fungsi untuk menghitung dan mencetak durasi waktu istirahat
void hitungDurasiIstirahat(int jMasuk, int mMasuk, int jKeluar, int mKeluar, int
*jam, int *menit) {
    // Jika pengurangan jam keluar oleh jam masuk tidak menghasilkan angka
    negatif
    if(mKeluar >= mMasuk && jKeluar >= jMasuk){
        *menit = mKeluar - mMasuk;
        *jam = jKeluar - jMasuk;
    }
    else if(mKeluar < mMasuk && jKeluar >= jMasuk){
        *menit = mKeluar + 60 - mMasuk;
        *jam = jKeluar - 1 - jMasuk;
    }
    else if(mKeluar >= mMasuk && jKeluar < jMasuk){
        *menit = mKeluar - mMasuk;
        *jam = jKeluar + 24 - jMasuk;
    }
}
```

```

    }
    else if(mKeluar < mMasuk && jKeluar < jMasuk){
        *menit = mKeluar + 60 - mMasuk;
        *jam = jKeluar + 23 - jMasuk;
    }
    else{
        *menit = 0;
        *jam = 0;
    }
}

int main() {
    int jam_masuk, jam_keluar, menit_masuk, menit_keluar;
    int total_jam, total_menit;
    char sign;

    // Input waktu masuk dan keluar
    printf("Masukkan waktu masuk (format HH:MM): ");
    scanf("%d%c%d", &jam_masuk, &sign, &menit_masuk);
    printf("Masukkan waktu keluar (format HH:MM): ");
    scanf("%d%c%d", &jam_keluar, &sign, &menit_keluar);

    // printf("%d %d %d %d", jam_masuk, menit_masuk, jam_keluar, menit_keluar);

    // Validasi input
    if(isValidTime(jam_masuk, menit_masuk, jam_keluar, menit_keluar, sign)){
        // Memanggil fungsi untuk menghitung durasi waktu istirahat
        hitungDurasiIstirahat(jam_masuk, menit_masuk, jam_keluar, menit_keluar,
&total_jam, &total_menit);
        if (total_jam != 0 && total_menit != 0){
            printf("Durasi waktu istirahat: %d jam %d menit", total_jam,
total_menit);
        }
        else if(total_jam != 0 && total_menit == 0){
            printf("Durasi waktu istirahat: %d jam", total_jam);
        }
        else if(total_jam == 0 && total_menit != 0){
            printf("Durasi waktu istirahat: %d menit", total_menit);
        }
        else{
            printf("Durasi waktu istirahat: %d jam", 24);
        }
    }

    else{
        printf("Format waktu salah!");
    }

    return 0;
}

```

Alasan Pemilihan Soal

Setelah soal untuk modul 2 diberikan kepada penulis, penulis membaca secara sekilas tentang kedua soal yang diberikan. Kemudian, penulis memutuskan untuk memilih soal pertama karena kompleksitas algoritma yang diperlukan pada program tidak terlalu tinggi, yaitu dengan menghitung jam istirahat dengan pengurangan jam. Meskipun dalam pengerjaannya, terdapat beberapa kendala yang akan dibahas pada subbagian *Kejadian Saat Praktikum*.

Strategi Awal/Rancangan Algoritma:

1. Pembuatan Fungsi: Pada program ini, digunakan dua fungsi utama, yaitu untuk menampilkan memvalidasi masukan yang diberikan kepada program dan juga untuk menghitung durasi istirahat.
2. Pengambilan Input: Input dengan format HH:MM diberikan pada program dan akan langsung diterminasi pada setiap pengambilan input. Sehingga, dalam program tidak menggunakan *while-loop* untuk mengulangi pengambilan input, meskipun input tidak valid. Selain itu, ada pemisah berupa karakter ':' diantara jam dan menitnya.
3. Algoritma Penghitungan: Penghitungan pada bagian fungsi *hitungDurasiIstirahat* menggunakan pengurangan jam analog dan didasari oleh beberapa kondisi. Penulis juga sempat melakukan dekomposisi masalah secara manual melalui kertas coretan yang dibawa.

Kejadian Saat Praktikum

Pada awal proses pengerjaan program, penulis mengalami kesulitan pada bagian pengambilan input fungsi. Hal ini dikarenakan tipe data yang diminta oleh format source code berupa *array of characters (string)* dan penulis masih belum begitu mahir dalam mengolah tipe data string. Beberapa hal sudah dilakukan untuk mengambil input string berupa jam dan menit dengan pemisah ':'. Mulanya, penulis menggunakan fungsi

```
scanf("%s:%s", waktu_masuk[0], waktu_masuk[1]);
```

dengan anggapan bahwa nilai jam dan menit akan tersimpan pada array indeks pertama dan kedua. Akan tetapi ketika program dijalankan, nilai input tidak tersimpan pada kedua variabel. Penulis mencari beberapa referensi untuk mengambil nilai input yang dipisahkan dengan tanda tertentu dan menemukan cara untuk melakukannya, yaitu dengan menggunakan fungsi *strtok()*. Namun, setelah mencoba fungsi tersebut dan program dijalankan, terdapat pesan error sehingga masalah masih belum terselesaikan. Pada akhirnya, masalah dapat terselesaikan dengan menambahkan notasi '&' untuk akses memori array.

Kemudian masalah datang ketika program perlu untuk memvalidasi input yang diberikan. Perancangan algoritma yang penulis buat sebelumnya menggunakan tipe data *integer*, sehingga perlu header `<string.h>` untuk merubah variabel dengan tipe data string menjadi integer. Penulis awalnya menggunakan fungsi *atoi()*, namun terjadi error pada program. Penulis menyadari bahwa masalah utama dari fungsi validasi input ini adalah penggunaan pointer pada variabel bertipe *char (char *waktu)*. Meski sudah bertanya kepada asisten, penulis masih belum bisa memecahkan permasalahan pengambilan dan validasi input. Pada akhirnya, penulis melakukan pendekatan lain, yaitu dengan mengganti semua tipe data variabel menjadi bertipe *integer*, alih-alih menggunakan tipe data *string*.

Tahapan Strategi (Milestone)

1. Pengambilan input

Pengambilan input waktu dengan format "HH:MM" dilakukan dengan menggunakan fungsi *scanf* sebagai berikut:

```
scanf("%d%c%d", &jam_masuk, &sign, &menit_masuk);  
scanf("%d%c%d", &jam_keluar, &sign, &menit_keluar);
```

kedua baris kode diatas akan mengambil nilai dua input dengan tipe data *integer* dari pengguna (jam dan menit) yang dipisahkan dengan tanda unik ':'.

2. Validasi input

Fungsi *isValidTime* dilakukan untuk memvalidasi input jam, menit, dan pemisah yang digunakan. Pemrosesan dilakukan dengan menggunakan *if-statement* dengan parameter sebagai berikut.

- 1.) jMasuk, jKeluar: Apabila kedua nilai jam bernilai kurang dari 24, maka input valid.
- 2.) mMasuk, mKeluar: Apabila kedua nilai menit bernilai kurang dari 60, maka input valid.
- 3.) sign: Apabila karakter pemisah yang digunakan adalah ':', maka input valid.

Apabila ketiga syarat di atas tidak terpenuhi, maka input yang diberikan kepada program tidak valid, dan akan ditampilkan pesan kesalahan.

3. Menghitung durasi istirahat

Untuk melakukan fungsi ini, penulis membagi proses penghitungan berdasarkan lima kasus seperti yang ditampilkan pada *source code*. Akan tetapi, penulis baru menyadari setelah keluar dari ruangan bahwa proses kondisional ini dapat menjadi lebih ringkas, yaitu dengan cara menghitung menitnya saja. Contoh dari kode tersebut adalah sebagai berikut.

```
void hitungDurasiIstirahat(int jMasuk, int mMasuk, int jKeluar, int mKeluar, int
*jam, int *menit) {
    int totalMenitMasuk = jMasuk * 60 + mMasuk;
    int totalMenitKeluar = jKeluar * 60 + mKeluar;

    int diffMenit = (totalMenitKeluar - totalMenitMasuk + 24 * 60) % (24 * 60);

    *jam = diffMenit / 60;
    *menit = diffMenit % 60;
}
```

Terlihat bahwa output jam diperoleh dengan menghitung hasil pembagian dari selisih menit keluar dan menit masuk dengan 60, serta output menit dapat diperoleh dengan menghitung sisa hasil bagi dari selisih menit keluar dan masuk dengan 60.

4. Pemanggilan fungsi dan penampilan output penghitungan

Pada bagian fungsi utama, penulis menggunakan *nested if-statement* karena spesifikasi dari soal meminta validasi input dan apabila total jam/menit bernilai nol, maka jam/menit tidak perlu ditampilkan.

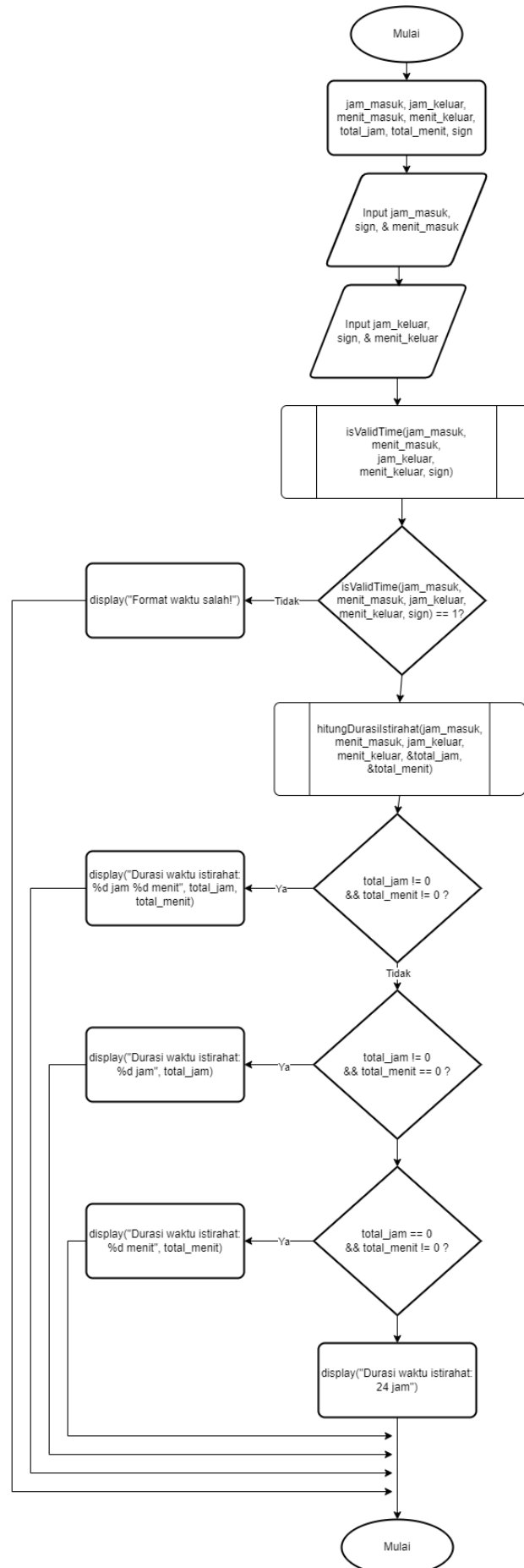
Proses Debugging

Pada saat proses *grading* melalui *github*, penulis mengalami dua kendala utama pada uji kasus ke-8 dan ke-10.

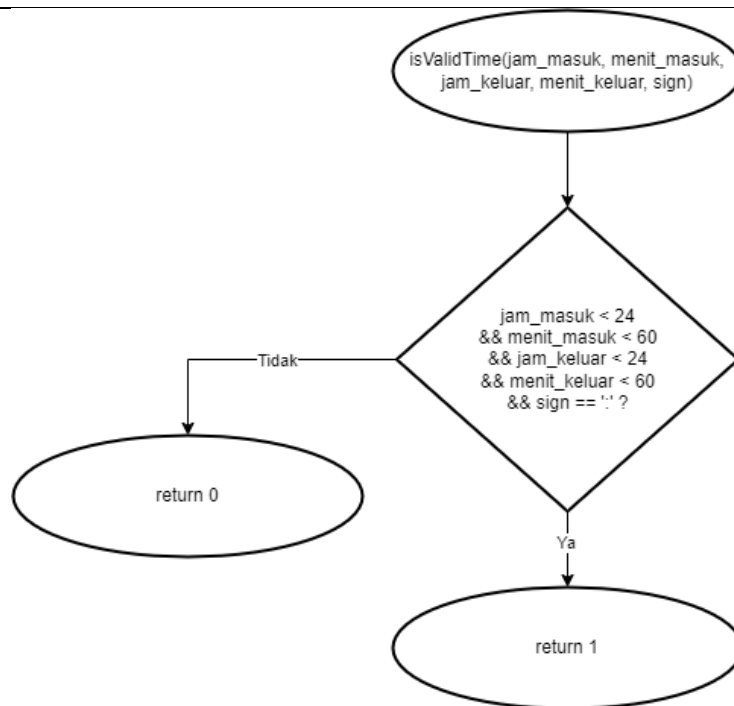
1.) Pada uji kasus ke-10, penulis tidak menyertakan durasi istirahat maksimum, yaitu 24 jam. Durasi ini diperoleh apabila pengguna memasukkan waktu masuk dan waktu keluar yang sama, misalnya dari 00:00 sampai 00:00. Untuk mengakomodir kasus ini, penulis menambahkan *if-statement* tambahan pada program main yang akan menampilkan waktu istirahat selama 24 jam apabila nilai jam dan menit bernilai nol.

2.) Pada uji kasus ke-8, penulis tidak melakukan validasi input pada karakter pemisah antara jam dan menit. Untuk mengatasi permasalahan ini, ditambahkan sebuah variabel baru bertipe *char* untuk menampung tanda pemisah. Kemudian pada bagian fungsi *isValidTime*, ditambahkan pula syarat kondisi untuk tanda pemisah harus menggunakan ':'. Apabila pengguna menggunakan tanda lain, maka akan ditampilkan pesan kesalahan format waktu.

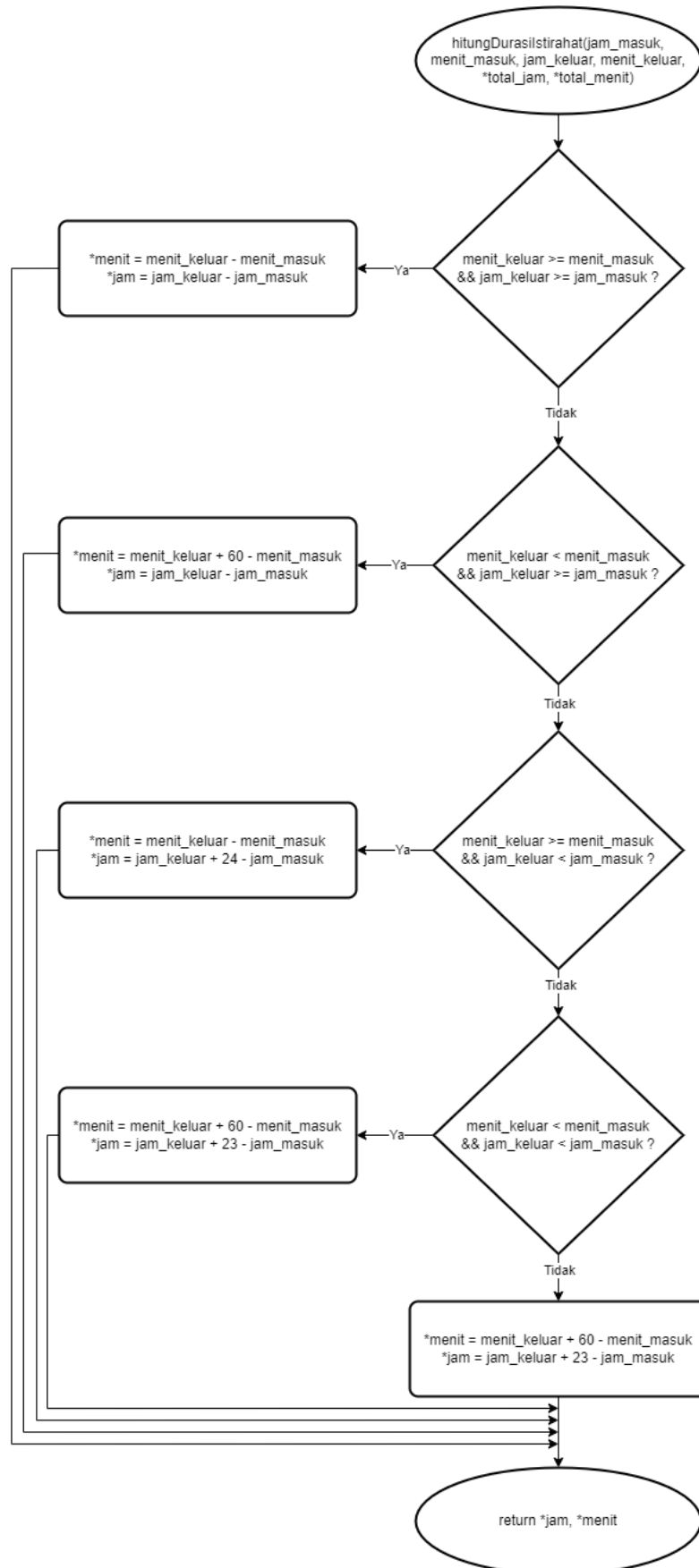
Diagram Alir



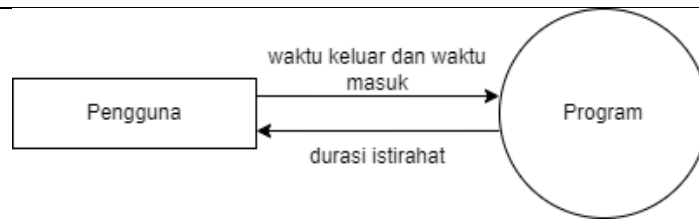
Gambar 1. Diagram alir fungsi utama program



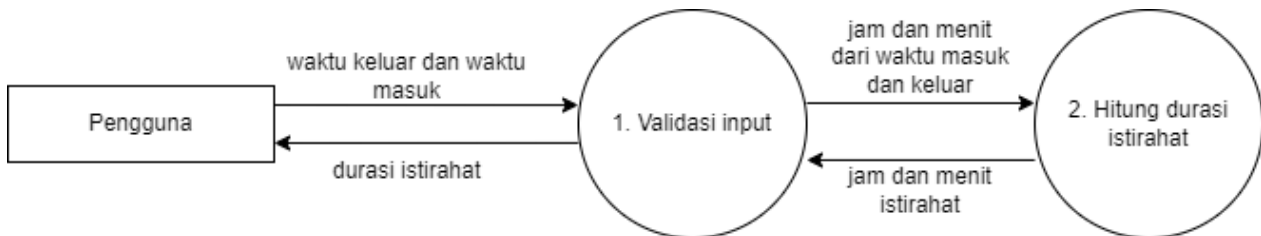
Gambar 2. Diagram alir fungsi `isValidTime()`



Gambar 3. Diagram alir fungsi hitungDurasiIstirahat



Gambar 4. Data flow diagram level 0



Gambar 5. Data flow diagram level 1