
PROJEK AKHIR

TRACKER LANGKAH

ANGGOTA KELOMPOK

DWI AGUS MAULANA
L0325022

AGIS LENTERA PRATAMA
L0325016

TIARA BELVA PRATISTHA
L0325014

PENDAHULUAN

A. LATAR BELAKANG

Olahraga adalah salah satu aktivitas penting yang bisa menjaga kesehatan di manusia. Olahraga banyak jenisnya salah satunya berjalan kaki. Terkadang seseorang kesulitan dalam memantau aktivitas jalan kaki yang telah ia lakukan, seperti pengguna kesulitan dalam mengingat berapa banyak langkah yang telah dicapai dalam satu hari, membandingkan progres harian dengan target yang telah ditentukan oleh pengguna dan melihat riwayat progres dengan performa terbaik atau terendah. Dengan program ini, pengguna dapat memantau langkah hariannya.

B. TUJUAN

- Membantu pengguna untuk mencatat jumlah langkah kaki harian.
- Memberikan target langkah harian dan mengevaluasi pencapaian langkah sesuai dengan target harian.
- Melihat progres harian dengan persentasi (%) sesuai dengan target langkah kaki yang di inputkan pengguna.
- Melihat status apakah sudah mencapai target atau belum.
- Memberikan ringkasan berjalan kaki dalam satu minggu.



PENDAHULUAN

C. RUANG LINGKUP

Program ini berfokus pada pencatatan langkah harian, yang mencakup fitur utama:

- Pencatatan langkah perhari
- Menghitung progres harian
- Menghitung hari yang mencapai target
- Menentukan hari terburuk (langkah paling sedikit) dan hari terbaik (langkah paling banyak)
- Menampilkan rangkuman progres dalam satu minggu

D. ALUR KERJA PROGRAM

- **Inisiasi Stuktur Data**

Saat program dijalankan, stuktur data akan mengisi 7 hari dalam seminggu (senin-minggu).

- **Input Langkah Perhari**

Pengguna diminta untuk memasukkan langkahnya dari hari senin hingga minggu secara berurutan



PENDAHULUAN

D. ALUR KERJA PROGRAM

- **Progress Harian**

Laporan harian yang terdiri dari hari, langkah, target harian, progress harian (%), dan status akan diberikan. Target harian adalah 10.000 langkah. Progress harian (%) adalah persentase langkah terhadap target harian. Status adalah keterangan "Target tercapai" atau "Target tidak tercapai".

- **Progress Mingguan**

Laporan mingguan yang terdiri dari total langkah, target mingguan, rata-rata harian, jumlah hari yang mencapai target, hari terbaik/terburuk, total kekurangan, dan status minggu tersebut. Status mingguan menggunakan nilai kekurangan. Semakin kecil nilai kekurangan, semakin baik status mingguannya. Status mingguan terdiri dari "SANGAT BAIK", "CUKUP BAIK", "BAIK", "CUKUP BURUK", "BURUK", dan "SANGAT BURUK".

- **Akhir Program**

Program akan berhenti saat progress harian dan progress mingguan sukses ditampilkan.



PENDAHULUAN

E. LIBRARY YANG DIGUNAKAN

- **Stdio.h**

Library ini adalah library standar dalam bahasa pemrograman C yang berfungsi untuk menangani operasi input dan output.



SOURCE CODE

Library

```
#include <stdio.h>
```

Stuktur Data

```
typedef struct DataHarian{  
    char hari[10];  
    int jmlLangkah;  
    int progress;  
} DataHarian;
```

Constant

```
#define JUMLAH_HARI 7
```

Global Variable

```
int target = 10000;
```



SOURCE CODE

Function Prototype

```
void inputData(DataHarian *data);

void calcProgress(int target, DataHarian *data);
int calcTotalLangkah(DataHarian *data);
int jmlHariTercapai(int target, DataHarian *data);
int totalKekurangan(int target, DataHarian *data);

void printStatus(int target, DataHarian data);
void printProgressHarian(int target, DataHarian *data);

void printHariTercapai(int target, DataHarian *data);
void printBestAndWorstDay(DataHarian *data);
void printStatusMingguIni(int target, DataHarian *data);
void printProgressMingguan(int target, DataHarian *data);
```



SOURCE CODE

Function inputData

```
void inputData(DataHarian *data) {  
    for (int i = 0; i < JUMLAH_HARI; i++) {  
        printf("%-8s: ", data[i].hari);  
        scanf("%d", &data[i].jmlLangkah);  
    }  
}
```

Function calcProgress

```
void calcProgress(int target, DataHarian *data) {  
    for (int i = 0; i < JUMLAH_HARI; i++) {  
        data[i].progress = ((data[i].jmlLangkah) * 100) / target ;  
    }  
}
```



SOURCE CODE

Function calcTotalLangkah

Function jmlHariTercapai

```
int calcTotalLangkah(DataHarian *data) {  
    int total = 0;  
    for (int i = 0; i < JUMLAH_HARI; i++) {  
        total += (data[i].jmlLangkah);  
    }  
    return total;  
}
```

```
int jmlHariTercapai(int target, DataHarian *data) {  
    int daysReached = 0;  
    for (int i = 0; i < JUMLAH_HARI; i++) {  
        if (data[i].jmlLangkah >= target) {  
            daysReached++;  
        }  
    }  
    return daysReached;  
}
```



SOURCE CODE

Function totalKekurangan

```
int totalKekurangan(int target, DataHarian *data) {  
    int langkahKekurangan = 0;  
    for (int i = 0; i < JUMLAH_HARI; i++) {  
        if (data[i].jmlLangkah < target) {  
            langkahKekurangan += (target - data[i].jmlLangkah);  
        }  
    }  
    return langkahKekurangan;  
}
```



SOURCE CODE

Function printStatus

```
void printStatus(int target, DataHarian data) {  
    if (data.jmlLangkah < target) {  
        printf("Kurang %d langkah\n", (target - (data.jmlLangkah)));  
    } else if (data.jmlLangkah == target) {  
        printf("TARGET TERCAPAI! tepat 10.000 langkah\n");  
    } else {  
        printf("TARGET TERCAPAI! +%d langkah\n", ((data.jmlLangkah) - target));  
    }  
}
```



SOURCE CODE

Function printProgressHarian

```
void printProgressHarian(int target, DataHarian *data) {  
    printf("\n===== PROGRES HARIAN =====\n");  
    for (int i = 0; i < JUMLAH_HARI; i++) {  
        printf("\n%s\n", data[i].hari);  
        printf("Langkah   : %d\n", data[i].jmlLangkah);  
        printf("Target    : %d\n", target);  
        printf("Progress  : %d%%\n", data[i].progress);  
        printf("Status    : ");  
        printStatus(target, data[i]);  
    }  
}
```



SOURCE CODE

Function printHariTercapai

```
void printHariTercapai(int target, DataHarian *data) {  
    int first = 1;  
  
    for (int i = 0; i < JUMLAH_HARI; i++) {  
        if (data[i].jmlLangkah >= target) {  
            if (!first) {  
                printf(", ");  
            }  
            printf("%s", data[i].hari);  
            first = 0;  
        }  
    }  
    printf("\n");  
}
```



SOURCE CODE

Function printBestAndWorstDay

```
void printBestAndWorstDay(DataHarian *data) {  
    int indexBest = 0;  
    int indexWorst = 0;  
  
    for (int i = 1; i < JUMLAH_HARI; i++) {  
        if (data[i].jmlLangkah > data[indexBest].jmlLangkah) {  
            indexBest = i;  
        }  
        if (data[i].jmlLangkah < data[indexWorst].jmlLangkah) {  
            indexWorst = i;  
        }  
    }  
  
    printf("\nHari terbaik           : %s (%d langkah)\n",  
        data[indexBest].hari,  
        data[indexBest].jmlLangkah);  
    printf("Hari terburuk           : %s (%d langkah)\n",  
        data[indexWorst].hari,  
        data[indexWorst].jmlLangkah);  
}
```



SOURCE CODE

Function printProgressMingguan

```
void printProgressMingguan(int target, DataHarian *data) {
    int totalLangkah = calcTotalLangkah(data);
    printf("\n===== RINGKASAN MINGGU INI =====\n");

    printf("Total langkah          : %d\n", totalLangkah);
    printf("Target mingguan           : %d\n", target * 7);
    printf("Rata-rata harian          : %d\n", totalLangkah / 7);

    printf("\nHari target tercapai    : %d/7 hari (%d%%)\n", jmlHariTercapai(target, data), (jmlHariTercapai(target, data) * 100) / 7);
    printHariTercapai(target, data);
    printBestAndWorstDay(data);

    printf("\nTotal kekurangan        : %d\n", totalKekurangan(target, data));
    printStatusMingguIni(target, data);
}
```



SOURCE CODE

Main Function

```
int main() {  
  
    DataHarian data[7] = {  
        {"Senin"},  
        {"Selasa"},  
        {"Rabu"},  
        {"Kamis"},  
        {"Jumat"},  
        {"Sabtu"},  
        {"Minggu"}  
    };  
}
```

```
    printf("\n===== TRACKER LANGKAH HARIAN =====\n");  
    printf("Target Harian: 10.000 langkah\n");  
    printf("INPUT LANGKAH HARIAN:\n");  
    inputData(data);  
    calcProgress(target, data);  
    printProgressHarian(target, data);  
    printProgressMingguan(target, data);  
  
    return 0;  
}
```



PENJELASAN FITUR

FITUR YANG TERDAPAT DALAM PROGRAM TRACKER

Pengelolaan dan Input Data Langkah Harian
Kalkulasi Progres dan Status Harian
Analisis Statistik Mingguan (Total dan Rata-rata)
Analisis Performa Terbaik dan Terburuk
Sistem Evaluasi Kinerja Mingguan (Rating)



PENJELASAN FITUR

PENGELOLAAN DAN INPUT DATA LANGKAH HARIAN

Fungsi:

Pengelolaan data harian menggunakan struktur data (struct DataHarian) untuk mengelompokkan informasi per hari. Menyimpan komponen data:

1. Nama hari (Senin-Minggu).
2. Jumlah langkah yang ditempuh.
3. Persentase progres pencapaian.

Saat program dimulai, pengguna diminta memasukkan data jumlah langkah untuk setiap hari dalam satu minggu secara berurutan.



PENJELASAN FITUR

PENGELOLAAN DAN INPUT DATA LANGKAH HARIAN

Fungsi:

Pengelolaan data harian menggunakan struktur data (struct DataHarian) untuk mengelompokkan informasi per hari. Menyimpan komponen data:

1. Nama hari (Senin-Minggu).
2. Jumlah langkah yang ditempuh.
3. Persentase progres pencapaian.

Saat program dimulai, pengguna diminta memasukkan data jumlah langkah untuk setiap hari dalam satu minggu secara berurutan.

Implementasi:

inputData() melakukan iterasi (perulangan) sebanyak 7 kali (sesuai JUMLAH_HARI).

Menggunakan array of struct DataHarian data[7] untuk menampung input pengguna yang dibaca menggunakan scanf.

PENJELASAN FITUR

KALKULASI PROGRES DAN STATUS HARIAN

Fungsi:

Menghitung persentase ketercapaian langkah berdasarkan target tetap (10.000 langkah). Memberikan umpan balik status harian kepada pengguna:

- Menampilkan sisa langkah jika target belum tercapai (Contoh: "Kurang 2000 langkah").
- Menampilkan ucapan selamat jika target tercapai atau terlampaui (Contoh: "TARGET TERCAPAI! +500 langkah").

Implementasi:

`calcProgress()` menggunakan rumus matematika sederhana: $(\text{langkah} * 100) / \text{target}$ untuk mengisi variabel `progress`. Fungsi `printStatus()` menggunakan logika percabangan (if-else) untuk menentukan pesan status yang sesuai (Kurang/Pas/Lebih).

PENJELASAN FITUR

ANALISIS STATISTIK MINGGUAN (TOTAL DAN RATA-RATA)

Fungsi:

Menyajikan ringkasan performa pengguna selama satu minggu penuh, meliputi:

1. Total langkah akumulatif.
2. Target mingguan (70.000 langkah).
3. Rata-rata langkah per hari.
4. Jumlah dan daftar nama hari di mana target harian berhasil dicapai.

Implementasi:

Fungsi `calcTotalLangkah()` menjumlahkan seluruh langkah dalam array. Fungsi `jmlHariTercapai()` menghitung frekuensi keberhasilan mencapai target. Fungsi `printHariTercapai()` mencetak nama-nama hari yang memenuhi kondisi `steps >= target`.

PENJELASAN FITUR

ANALISIS PERFORMA TERBAIK DAN TERBURUK

Fungsi:

Mengidentifikasi hari dengan kinerja paling optimal dan paling rendah dalam satu minggu:

- Hari Terbaik: Hari dengan jumlah langkah terbanyak.
- Hari Terburuk: Hari dengan jumlah langkah paling sedikit.

Implementasi:

Fungsi `printBestAndWorstDay()` menggunakan algoritma pencarian nilai minimum dan maksimum (Min-Max). Melakukan perbandingan nilai `jmlLangkah` antar indeks array untuk menemukan indeks dengan nilai tertinggi dan terendah.

PENJELASAN FITUR

SISTEM EVALUASI KINERJA MINGGUAN (RATING)

Fungsi:

Memberikan penilaian kualitatif ("Raport") terhadap performa mingguan pengguna berdasarkan total kekurangan langkah ("hutang langkah"). Kategori penilaian:

- SANGAT BAIK (Kekurangan ≤ 5.000).
- CUKUP BAIK (Kekurangan ≤ 10.000).
- BAIK (Kekurangan ≤ 20.000).
- CUKUP BURUK (Kekurangan ≤ 30.000).
- BURUK (Kekurangan ≤ 50.000).
- SANGAT BURUK (Kekurangan > 50.000).

Implementasi:

`totalKekurangan()` menghitung selisih antara target dan langkah hanya pada hari-hari yang targetnya tidak tercapai. Fungsi `printStatusMingguIni()` menggunakan struktur kontrol bertingkat (if-else if) untuk mencetak predikat status sesuai rentang nilai kekurangan.

OUTPUT PROGRAM

INPUT LANGKA HARIAN

```
===== TRACKER LANGKAH HARIAN =====  
Target Harian: 10.000 langkah  
INPUT LANGKAH HARIAN:  
Senin      : 9200  
Selasa     : 11500  
Rabu       : 8500  
Kamis      : 12000  
Jumat      : 9800  
Sabtu      : 7500  
Minggu     : 10500
```


OUTPUT PROGRAM

LAPORAN HARIAN

```
===== PROGRES HARIAN =====
```

Senin

Langkah : 9200

Target : 10000

Progress : 92%

Status : Kurang 800 langkah

Selasa

Langkah : 11500

Target : 10000

Progress : 115%

Status : TARGET TERCAPAI! +1500 langkah

Rabu

Langkah : 8500

Target : 10000

Progress : 85%

Status : Kurang 1500 langkah

Kamis

Langkah : 12000

Target : 10000

Progress : 120%

Status : TARGET TERCAPAI! +2000 langkah

Jumat

Langkah : 9800

Target : 10000

Progress : 98%

Status : Kurang 200 langkah

Sabtu

Langkah : 7500

Target : 10000

Progress : 75%

Status : Kurang 2500 langkah

Minggu

Langkah : 10500

Target : 10000

Progress : 105%

Status : TARGET TERCAPAI! +500 langkah

OUTPUT PROGRAM

LAPORAN MINGGUAN

===== RINGKASAN MINGGU INI =====

Total langkah : 69000

Target mingguan : 70000

Rata-rata harian : 9857

Hari target tercapai : 3/7 hari (42%)

Selasa, Kamis, Minggu

Hari terbaik : Kamis (12000 langkah)

Hari terburuk : Sabtu (7500 langkah)

Total kekurangan : 5000

Status minggu ini : SANGAT BAIK

KESIMPULAN

Setelah menyelesaikan proyek pemrograman Tracker Langkah ini, dapat disimpulkan bahwa program ini sukses mencapai tujuan yang direncanakan. Penggunaan struktur data Struct memungkinkan data disimpan dengan rapi dan pengolahan data yang efisien.

Program berhasil mengimplementasikan fitur-fitur utama seperti memasukkan langkah harian, memberi laporan harian, dan memberi laporan mingguan.

Program dapat menentukan status harian dan status mingguan memanfaatkan data yang diberikan pengguna dan mengolahnya dengan membagi-bagi proses pengolahan menjadi beberapa fungsi dengan perintah yang berbeda-beda.
