

LAPORAN TUGAS BESAR
APLIKASI MANAJEMEN DAN PENCATATAN WORKOUT
HARIAN



Yang di susun oleh:

Kelompok : 17
Anggota : 1. Fauzan Rafif | 103112400227
2. Gilar Saputra | 103112400253

TELKOM UNIVERSITY PURWOKERTO
S1 TEKNIK INFORMATIKA
2024/2025

Pendahuluan

Tugas besar kali ini bertujuan untuk mengembangkan atau membuat suatu aplikasi yang dapat membantu user atau pengguna untuk memantau aktivitas workout pengguna. Aplikasi ini bertujuan agar user dapat melihat aktivitas workoutnya dengan fitur-fitur seperti menambah, mengubah, menghapus dan menampilkan data aktivitas, serta akan membantu pengguna untuk memantau kalori selama beraktivitas. Aplikasi ini juga akan menampilkan data statistik anda selama menggunakan aplikasi, dan sistem dapat memberikan rekomendasi latihan kepada pelanggan. Tujuan lainnya jua agar pelanggan atau user dapat memantau kemajuan aktivitas workout mereka selama menggunakan aplikasi ini.

Deskripsi

Aplikasi ini digunakan untuk mencatat aktivitas workout pengguna dan memberikan rekomendasi latihan. Data utama yang digunakan adalah daftar latihan, durasi, jumlah kalori, dan jadwal latihan. Pengguna aplikasi adalah individu yang ingin melacak kemajuan olahraga mereka, dengan fitur-fitur yang disediakan, yaitu

1. **Manajemen data** : menambah, mengubah, menghapus, dan menampilkan data aktivitas workout.
2. **Searching** : user dapat mencari latihan yang sudah user lakukan.
3. **Sorting** : user dapat menyorting atau mengurutkan kalori dan durasi, dimana ini berguna untuk mempermudah user dapat melihat suatu data latihan menghabiskan berapa lama waktu dan kalori yang di bakar saat olahraga tersebut
4. **Rekomendasi Latihan** : system akan membantu user memberikan rekomendasi latihan berdasarkan pola sebelumnya .
5. **Statistik/laporan** : sistem dapat mengelola data aktivitas user selama menggunakan aplikasi.

Berikut Deskripsi Kode :




```

1  import (
2      "fmt"
3      "strings"
4      "time"
5  )
6

```

Gambar 1.0 – Import Library



```

1  // struktur dari tanggal, kategori, nama, durasi, kalori latihan (opsi 1-bagian 2)
2  type Workout struct {
3      tanggalWorkout time.Time
4      kategoriWorkout string
5      namaLatihan     string
6      durasiLatihan   int
7      kalori          float64
8  }
9
10 // struktur dari jenis kategori dan kalori yang dibakar per detik (opsi 1-bagian 1)
11 type kategoriWorkout struct {
12     jenisLatihan string
13     kaloriPerMenit float64
14 }
15
16 // Opsi jenis latihan dimana kalori didapat dalam hitungan 'kalori per menit'
17 var jenisLatihan_kategori = map[string][]kategoriWorkout{
18     "Bahu-dan-Punggung": {
19         {"pull-up", 13.8},
20         {"bent-over row", 15.0},
21         {"face-pull", 12.0},
22         {"shrug", 10.8},
23         {"superman-hold", 9.6},
24     },
25     "Lengan": {
26         {"bicep-curl", 10.2},
27         {"tricep-dip", 12.0},
28         {"hammer-curl", 10.8},
29         {"overhead-tricep-extension", 11.4},
30         {"concentration-curl", 9.6},
31     },
32     "Dada": {
33         {"push-up", 12.6},
34         {"chest-fly", 13.2},
35         {"bench-press", 15.0},
36         {"incline-push-up", 12.0},
37         {"cable-crossover", 14.4},
38     },
39     "Perut": {
40         {"plank", 15.0},
41         {"sit-up", 10.2},
42         {"leg-raise", 12.0},
43         {"russian-twist", 13.2},
44         {"mountain-climber", 15.6},
45     },
46     "Kaki": {
47         {"squat", 13.8},
48         {"lunges", 13.2},
49         {"jumping-squat", 16.2},
50         {"calf-raise", 10.8},
51         {"wall-sit", 11.4},
52     },
53 }
54
55 // membuat array (slice) global untuk menampung daftar latihan
56 var daftarLatihan []Workout
57

```

Gambar 1.1 – Struktur dan array

```

1 // Data Dummy (Fake Data) untuk riwayat workout 2 hari sebelumnya dengan manual
2 func dataDummy() {
3     tanggal1 := time.Now().AddDate(0, 0, -2)
4     tanggal2 := time.Now().AddDate(0, 0, -1)
5
6     dummy1 := []Workout{
7         {tanggal1, "Dada", "push-up", 6, 12.6 * 6},
8         {tanggal1, "Lengan", "hammer-curl", 7, 10.8 * 7},
9         {tanggal1, "Bahu-dan-Punggung", "shrug", 7, 10.8 * 7},
10        {tanggal1, "Dada", "cable-crossover", 5, 14.4 * 5},
11        {tanggal1, "Lengan", "tricep-dip", 6, 12.0 * 6},
12        {tanggal1, "Bahu-dan-Punggung", "face-pull", 6, 12.0 * 6},
13        {tanggal1, "Dada", "push-up", 6, 12.6 * 6},
14    }
15
16    dummy2 := []Workout{
17        {tanggal2, "Perut", "plank", 8, 15.0 * 8},
18        {tanggal2, "Kaki", "squat", 10, 13.8 * 10},
19        {tanggal2, "Perut", "russian-twist", 6, 13.2 * 6},
20        {tanggal2, "Kaki", "jumping-squat", 6, 16.2 * 6},
21        {tanggal2, "Perut", "mountain-climber", 5, 15.6 * 5},
22    }
23
24    daftarLatihan = append(daftarLatihan, dummy1...)
25    daftarLatihan = append(daftarLatihan, dummy2...)
26 }
27

```

Gambar 1.2 – Data Dummy (Fake data)

```

1 // tampilkan data latihan
2 func tampilkanLatihan() {
3     if len(daftarLatihan) == 0 {
4         fmt.Println("Belum ada data latihan yang tersimpan.")
5         return
6     }
7
8     fmt.Println("-----")
9     fmt.Printf("| %-3s | %-10s | %-20s | %-20s | %-10s | %-10s |\n", "No", "Tanggal", "Jenis Latihan", "Kategori", "Durasi", "Kalori")
10    fmt.Println("-----")
11
12    for i := 0; i < len(daftarLatihan); i++ {
13        latihan := daftarLatihan[i]
14        fmt.Printf("| %-3d | %-10s | %-20s | %-20s | %-4d menit | %-10.2f |\n",
15            i+1,
16            latihan.tanggalWorkout.Format("02-01-2006"),
17            latihan.namaLatihan,
18            latihan.kategoriWorkout,
19            latihan.durasiLatihan,
20            latihan.kalori)
21    }
22
23    fmt.Println("----- \n")
24 }

```

Gambar 1.3 – Fitur Tampilkan Data

```

1 // Fitur menu 1 -> menambah data latihan
2 func tambahLatihan() {
3     var kategori = []string{"Bahu-dan-Punggung", "Lengan", "Dada", "Perut", "Kaki"}
4
5     // User menginput kategori workout
6     var pilihKategori int
7     fmt.Println("Kategori Workout: ")
8     for i := 0; i < len(kategori); i++ {
9         fmt.Printf("%d. %s \n", i+1, kategori[i])
10    }
11    fmt.Print("Pilih Kategori Workout (1 - 5): ")
12    fmt.Scan(&pilihKategori)
13
14    jenisKategori := kategori[pilihKategori-1] //untuk program tahu index kategori mana yang di pilih
15    listLatihan_kategori := jenisLatihan_kategori[jenisKategori] //menampilkan list latihan yang ada pada kategori yang dipilih
16
17    // user menginput nama latihan yang tersedia di tiap kategori
18    var pilihLatihan int
19    for i := 0; i < len(listLatihan_kategori); i++ {
20        fmt.Printf("%d. %s \n", i+1, listLatihan_kategori[i].jenisLatihan)
21    }
22    fmt.Print("Pilih jenis latihan yang ingin anda lakukan: ")
23    fmt.Scan(&pilihLatihan)
24
25    namaLatihan := listLatihan_kategori[pilihLatihan-1].jenisLatihan
26    kalori_perMenit := listLatihan_kategori[pilihLatihan-1].kaloriPerMenit
27
28    // user menginput , durasi dan program menghitung kalori
29    tanggal := time.Now()
30    var durasi int
31    fmt.Printf("Berapa lama anda latihan \"%s\" (dalam menit): ", namaLatihan)
32    fmt.Scan(&durasi)
33
34    kaloriTerbakar := kalori_perMenit * float64(durasi)
35
36    dataLatihan := Workout{
37        tanggalWorkout: tanggal,
38        kategoriWorkout: jenisKategori,
39        namaLatihan: namaLatihan,
40        durasiLatihan: durasi,
41        kalori: kaloriTerbakar,
42    }
43
44    daftarLatihan = append(daftarLatihan, dataLatihan)
45 }
46

```

Gambar 1.4 – Fitur Tambah Data

```

1 // Fitur menu 3 -> hapus data latihan
2 func hapusLatihan() {
3     if len(daftarLatihan) == 0 {
4         fmt.Println("Tidak data latihan.")
5         return
6     }
7
8     tampilkanLatihan()
9
10    var nomorLatihan int
11    fmt.Print("Masukkan nomor latihan yang ingin dihapus: ")
12    fmt.Scan(&nomorLatihan)
13    if nomorLatihan < 1 || nomorLatihan > len(daftarLatihan) {
14        fmt.Println("Nomor yang dimasukkan tidak ada.")
15        return
16    }
17
18    for i := nomorLatihan - 1; i < len(daftarLatihan)-1; i++ {
19        daftarLatihan[i] = daftarLatihan[i+1]
20    }
21    daftarLatihan = daftarLatihan[:len(daftarLatihan)-1]
22    fmt.Println("\nData latihan berhasil dihapus. \n")
23 }
24

```

Gambar 1.5 – Fitur Hapus Data

```

1 // Fitur menu 2 -> mengubah data latihan
2 func ubahLatihan() {
3     tampilkanLatihan()
4
5     var pilihNomor int
6     fmt.Print("Pilih nomor latihan yang ingin diubah: ")
7     fmt.Scan(&pilihNomor)
8
9     fmt.Println("Masukan data baru :")
10    var kategori = []string{"Bahu-dan-Punggug", "Lengan", "Dada", "Perut", "Kaki"}
11    var pilihKategori int
12    fmt.Println("Kategori Workout: ")
13    for i := 0; i < len(kategori); i++ {
14        fmt.Printf("%d. %s \n", i+1, kategori[i])
15    }
16    fmt.Print("Pilih Kategori Workout (1-5): ")
17    fmt.Scan(&pilihKategori)
18
19    jenisKategori := kategori[pilihKategori-1]
20    listLatihan_kategori := jenisLatihan_kategori[jenisKategori]
21
22    var pilihLatihan int
23    fmt.Println("\nJenis Latihan:")
24    for i := 0; i < len(listLatihan_kategori); i++ {
25        fmt.Printf("%d. %s \n", i+1, listLatihan_kategori[i].jenisLatihan)
26    }
27    fmt.Print("Pilih jenis latihan (1-", len(listLatihan_kategori), "): ")
28    fmt.Scan(&pilihLatihan)
29
30    namaLatihan := listLatihan_kategori[pilihLatihan-1].jenisLatihan
31    kalori_menit := listLatihan_kategori[pilihLatihan-1].kaloriPerMenit
32
33    var durasi int
34    fmt.Printf("\nBerapa lama anda latihan \"%s\" (dalam menit): ", namaLatihan)
35    fmt.Scan(&durasi)
36
37    kaloriTerbakar := kalori_menit * float64(durasi)
38
39    daftarLatihan[pilihNomor-1] = Workout{
40        tanggalWorkout: time.Now(), // Tanggal diupdate ke hari ini
41        kategoriWorkout: jenisKategori,
42        namaLatihan:      namaLatihan,
43        durasiLatihan:    durasi,
44        kalori:           kaloriTerbakar,
45    }
46
47    fmt.Println("\nData latihan berhasil diubah!")
48 }
49

```

Gambar 1.6 – Fitur Ubah Data

```

1 // < --- START SEARCHING ---> //
2
3 // Fungsi untuk mencari latihan berdasarkan jenis menggunakan sequential search
4 func Sequential(data []Workout, target string) int { // fungsi sequential search (Digunakan dengan memeriksa satu per satu dari awal hingga akhir)
5     for i := 0; i < len(data); i++ {
6         if data[i].namaLatihan == target {
7             return i
8         }
9     }
10    return -1
11 }
12
13 // Fungsi untuk mencari latihan berdasarkan jenis menggunakan binary search
14 func Binary(data []Workout, target string) int { // fungsi binary search (Digunakan dengan memeriksa data yang berada ditengah)
15     awal := 0
16     akhir := len(data) - 1
17
18     for awal <= akhir {
19         pembagi := (awal + akhir) / 2
20
21         if strings.ToLower(data[pembagi].namaLatihan) == target {
22             return pembagi
23         } else if strings.ToLower(data[pembagi].namaLatihan) < target {
24             awal = pembagi + 1
25         } else {
26             akhir = pembagi - 1
27         }
28     }
29     return -1
30 }
31
32 func UrutkanNama(data []Workout) { // Fungsi mengurutkan nama latihan
33     n := len(data)
34     for i := 0; i < n-1; i++ {
35         for j := 0; j < n-1-i; j++ {
36             if strings.ToLower(data[j].namaLatihan) > strings.ToLower(data[j+1].namaLatihan) {
37                 data[j], data[j+1] = data[j+1], data[j]
38             }
39         }
40     }
41 }
42
43 // Fitur menu 4 -> cari data latihan
44 func LatihanYangDicari() {
45     if len(daftarLatihan) == 0 {
46         fmt.Println("Belum ada data latihan untuk dicari.")
47         return
48     }
49
50     fmt.Print("Masukkan nama latihan: ")
51     var Dicari string
52     fmt.Scan(&Dicari)
53     Dicari = strings.ToLower(Dicari)
54
55     fmt.Println("1. Metode Sequential")
56     fmt.Println("2. Metode Binary")
57     fmt.Print("Pilih metode pencarian: ")
58     var metode int
59     fmt.Scan(&metode)
60
61     var hasil []Workout
62
63     if metode == 1 {
64         for i := 0; i < len(daftarLatihan); i++ {
65             if strings.ToLower(daftarLatihan[i].namaLatihan) == Dicari {
66                 hasil = append(hasil, daftarLatihan[i])
67             }
68         }
69     } else if metode == 2 {
70         salinan := make([]Workout, len(daftarLatihan))
71         copy(salinan, daftarLatihan)
72         UrutkanNama(salinan)
73
74         index := Binary(salinan, Dicari)
75         if index != -1 {
76             hasil = append(hasil, salinan[index])
77         }
78     } else {
79         fmt.Println("Metode pencarian tidak valid.")
80         return
81     }
82
83     // Tampilkan hasil
84     if len(hasil) > 0 {
85         fmt.Println("Hasil Pencarian:")
86         fmt.Println("-----")
87         fmt.Printf("| %-3s | %-10s | %-20s | %-20s | %-10s | %-10s |\n", "No", "Tanggal", "Jenis Latihan", "Kategori", "Durasi", "Kalori")
88         fmt.Println("-----")
89         for i := 0; i < len(hasil); i++ {
90             fmt.Printf("| %-3d | %-10s | %-20s | %-20s | %-10.2f menit | %-10.2f |\n",
91                 i+1,
92                 hasil[i].tanggalWorkout.Format("02-01-2006"),
93                 hasil[i].namaLatihan,
94                 hasil[i].kategoriWorkout,
95                 hasil[i].durasiLatihan,
96                 hasil[i].kalori)
97         }
98         fmt.Println("-----")
99     } else {
100        fmt.Println("Latihan dengan nama tersebut tidak ditemukan.")
101    }
102 }
103
104 // < --- END SEARCHING ---> //
105

```

Gambar 1.7 – Fitur Searching

```

1 // < --- START SORTING ---> //
2
3 var daftarLatihanBackup []Workout // backup data asli sebelum di sorting
4
5 // Selection sort "Durasi"
6 func selectionSort(data []Workout, ascending bool) {
7     n := len(data)
8     for i := 0; i < n-1; i++ {
9         idx := i
10        for j := i + 1; j < n; j++ {
11            if ascending {
12                if data[j].durasilatihan < data[idx].durasilatihan {
13                    idx = j
14                }
15            } else {
16                if data[j].durasilatihan > data[idx].durasilatihan {
17                    idx = j
18                }
19            }
20        }
21        data[i], data[idx] = data[idx], data[i]
22    }
23 }
24
25 // Selection sort "Kalori"
26 func insertionSortKalori(data []Workout, ascending bool) {
27     n := len(data)
28     for i := 1; i < n; i++ {
29         temp := data[i]
30         j := i - 1
31
32         if ascending {
33             for j >= 0 && data[j].kalori > temp.kalori {
34                 data[j+1] = data[j]
35                 j--
36             }
37         } else {
38             for j >= 0 && data[j].kalori < temp.kalori {
39                 data[j+1] = data[j]
40                 j--
41             }
42         }
43
44         data[j+1] = temp
45     }
46 }
47
48 // Fitur menu 5 -> sorting kalori dan durasi
49 func sorting() {
50
51     if len(daftarLatihanBackup) == 0 {
52         daftarLatihanBackup = make([]Workout, len(daftarLatihan))
53         copy(daftarLatihanBackup, daftarLatihan)
54     }
55     var pilihJenis int
56     var urutan int
57
58     fmt.Println("Pilih jenis sorting:")
59     fmt.Println("1. Durasi (Selection Sort)")
60     fmt.Println("2. Kalori (Insertion Sort)")
61     fmt.Println("3. Kembali ke data awal")
62     fmt.Print("Pilih menu di atas: ")
63     fmt.Scan(&pilihJenis)
64
65     if pilihJenis == 3 {
66         if len(daftarLatihanBackup) > 0 {
67             daftarLatihan = make([]Workout, len(daftarLatihanBackup))
68             copy(daftarLatihan, daftarLatihanBackup)
69         }
70         return
71     }
72
73     fmt.Println("Urutan:")
74     fmt.Println("1. Dari yang terkecil")
75     fmt.Println("2. Dari yang terbesar")
76     fmt.Print("Masukkan pilihan urutan: ")
77     fmt.Scan(&urutan)
78
79     ascending := (urutan == 1)
80
81     salinanData := make([]Workout, len(daftarLatihan))
82     copy(salinanData, daftarLatihan)
83
84     if pilihJenis == 1 {
85         selectionSort(salinanData, ascending)
86     } else if pilihJenis == 2 {
87         insertionSortKalori(salinanData, ascending)
88     }
89
90     daftarLatihan = salinanData
91 }
92
93 // < --- END SORTING ---> //
94

```

Gambar 1.8 – Fitur Sorting


```

1 // Fitur menu 6 -> rekomendasi latihan
2 func rekomendasi() {
3     tanggalTerakhir := daftarLatihan[len(daftarLatihan)-1].tanggalWorkout.Format("02-01-2006")
4
5     var bodyAtas, bodyBawah int
6     for _, data := range daftarLatihan {
7         if data.tanggalWorkout.Format("02-01-2006") == tanggalTerakhir {
8             kategori := strings.ToLower(data.kategoriWorkout)
9             if kategori == "bahu-dan-punggung" || kategori == "dada" || kategori == "lengan" {
10                 bodyAtas++
11             } else if kategori == "perut" || kategori == "kaki" {
12                 bodyBawah++
13             }
14         }
15     }
16
17     var targetKategori []string
18     if bodyAtas > bodyBawah {
19         targetKategori = []string{"Bahu-dan-Punggung", "Dada", "Lengan"}
20         fmt.Println("Hari ini dominan latihan upper body, rekomendasi dari kategori: ", targetKategori)
21     } else {
22         targetKategori = []string{"Perut", "Kaki"}
23         fmt.Println("Hari ini dominan latihan lower body, rekomendasi dari kategori: ", targetKategori)
24     }
25
26     //Gabungkan seluruh jenis latihan
27     var semuaLatihan []kategoriWorkout
28     for _, kategori := range targetKategori {
29         semuaLatihan = append(semuaLatihan, jenisLatihan_kategori[kategori]...)
30     }
31
32     // Urutan berdasarkan kalori per menit tertinggi pada data kartegori
33     for i := 0; i < len(semuaLatihan)-1; i++ {
34         maxIdx := i
35         for j := i + 1; j < len(semuaLatihan); j++ {
36             if semuaLatihan[j].kaloriPerMenit > semuaLatihan[maxIdx].kaloriPerMenit {
37                 maxIdx = j
38             }
39         }
40         semuaLatihan[i], semuaLatihan[maxIdx] = semuaLatihan[maxIdx], semuaLatihan[i]
41     }
42
43     // Rekomendasi hingga 500 kalori
44     var totalKalori float64
45     var rekomendasi []Workout
46
47     for _, item := range semuaLatihan {
48         if totalKalori >= 500 {
49             break
50         }
51         sisaKalori := 500 - totalKalori
52
53         durasi := int(sisaKalori / item.kaloriPerMenit)
54         if durasi < 3 {
55             durasi = 3
56         } else if durasi > 15 {
57             durasi = 15
58         }
59         kalori := item.kaloriPerMenit * float64(durasi)
60
61         // Cari kategori latihan
62         var kategorilatihan string
63         for _, kategori := range targetKategori {
64             for _, latihan := range jenisLatihan_kategori[kategori] {
65                 if latihan.jenisLatihan == item.jenisLatihan {
66                     kategorilatihan = kategori
67                 }
68             }
69         }
70
71         rekomendasi = append(rekomendasi, Workout{
72             namaLatihan: item.jenisLatihan,
73             kategoriWorkout: kategorilatihan,
74             durasilatihan: durasi,
75             kalori: kalori,
76         })
77         totalKalori += kalori
78     }
79
80     // Output
81     fmt.Printf("\n██ Rekomendasi Latihan (Target ~500 Kalori)\n")
82     fmt.Println("-----")
83     fmt.Printf("| %-20s | %-20s | %-10s | %-10s |\n", "Jenis Latihan", "Kategori", "Durasi", "Kalori")
84     fmt.Println("-----")
85     for _, item := range rekomendasi {
86         fmt.Printf("| %-20s | %-20s | %-4d menit | %-10.2f |\n",
87             item.namaLatihan,
88             item.kategoriWorkout,
89             item.durasilatihan,
90             item.kalori)
91     }
92     fmt.Println("-----")
93     fmt.Printf("Total estimasi kalori: %.2f kalori\n\n", totalKalori)
94 }
95

```

Gambar 1.9 – Fitur Rekomendasi Latihan

```

1 // Fitur menu 8 -> tampilkan statistik
2 func statistik() {
3     fmt.Println("Statistik akan menampilkan 10 aktivitas terakhir dan juga data latihan per hari")
4     fmt.Println("dan anda dapat melihat total kalori dalam periode tertentu \n")
5
6     var tanggalAwal, tanggalAkhir string
7     fmt.Print("\nMasukkan tanggal mulai (DD-MM-YYYY): ")
8     fmt.Scan(&tanggalAwal)
9     fmt.Print("Masukkan tanggal akhir (DD-MM-YYYY): ")
10    fmt.Scan(&tanggalAkhir)
11
12    tMulai, err1 := time.Parse("02-01-2006", tanggalAwal)
13    tAkhir, err2 := time.Parse("02-01-2006", tanggalAkhir)
14
15    if err1 != nil || err2 != nil {
16        fmt.Println("Format tanggal salah. Harus DD-MM-YYYY.")
17        return
18    }
19
20    // --- Tampilkan 10 Aktivitas Terakhir ---
21    fmt.Println("10 Aktivitas Terakhir:")
22    fmt.Println("-----")
23    fmt.Printf("%-3s | %-10s | %-20s | %-20s | %-10s | %-10s | \n", "No", "Tanggal", "Jenis Latihan", "Kategori", "Durasi", "Kalori")
24    fmt.Println("-----")
25
26    sumberData := daftarLatihanBackup
27    if len(sumberData) == 0 {
28        sumberData = daftarLatihan
29    }
30
31    jumlahData := len(sumberData)
32    start := jumlahData - 10
33    if start < 0 {
34        start = 0
35    }
36
37    no := 1
38    for i := start; i < jumlahData; i++ {
39        data := sumberData[i]
40        fmt.Printf("%-3d | %-10s | %-20s | %-20s | %-4d menit | %-10.2f | \n",
41            no,
42            data.tanggalWorkout.Format("02-01-2006"),
43            data.namaLatihan,
44            data.kategoriWorkout,
45            data.durasiLatihan,
46            data.kalori)
47    }
48    fmt.Println("-----")
49
50    var totalKalori float64
51    for _, latihan := range daftarLatihan {
52        if !latihan.tanggalWorkout.Before(tMulai) && !latihan.tanggalWorkout.After(tAkhir) {
53            totalKalori += latihan.kalori
54        }
55    }
56
57    fmt.Printf("\nTotal Kalori dalam periode %s sampai %s: %.2f kalori\n\n",
58        tMulai.Format("02-01-2006"), tAkhir.Format("02-01-2006"), totalKalori)
59
60    // --- Statistik per Hari ---
61    statistikPerHari := make(map[string]struct {
62        jumlahLatihan int
63        totalKalori    float64
64        totalDurasi    int
65    })
66
67    for _, data := range daftarLatihan {
68        keyTanggal := data.tanggalWorkout.Format("02-01-2006")
69        stat := statistikPerHari[keyTanggal]
70        stat.jumlahLatihan++
71        stat.totalDurasi += data.durasiLatihan
72        stat.totalKalori += data.kalori
73        statistikPerHari[keyTanggal] = stat
74    }
75
76    fmt.Println("\nStatistik Per Hari:")
77    fmt.Println("-----")
78    for tanggal, data := range statistikPerHari {
79        fmt.Println("Tanggal      :", tanggal)
80        fmt.Println("Jumlah Latihan :", data.jumlahLatihan)
81        fmt.Printf("Total Durasi   : %d menit\n", data.totalDurasi)
82        fmt.Printf("Jumlah Kalori  : %.2f kalori\n", data.totalKalori)
83        fmt.Println()
84        no++
85    }
86    fmt.Println("-----")
87
88 }
89

```

Gambar 2.0 – Fitur Statistik

```

1 func main() {
2     var pilihMenu int
3
4     dataDummy()
5
6     for {
7         tampilkanLatihan()
8
9         fmt.Println("===== Aplikasi Workout =====")
10        fmt.Println("1. Tambah Data Latihan")
11        fmt.Println("2. Ubah Data Latihan")
12        fmt.Println("3. Hapus Data Latihan")
13        fmt.Println("4. Cari Jenis Latihan") // ojan yg service
14        fmt.Println("5. Urutkan Data Latihan")
15        fmt.Println("6. Rekomendasi Latihan")
16        fmt.Println("7. Tampilkan Statistik")
17        fmt.Println("8. Keluar")
18
19        fmt.Print("Silahkan pilih menu yang ingin anda gunakan: ")
20        fmt.Scan(&pilihMenu)
21        fmt.Println()
22
23        switch pilihMenu {
24            case 1:
25                tambahLatihan()
26            case 2:
27                ubahLatihan()
28            case 3:
29                hapusLatihan()
30            case 4:
31                LatihanYangDicari()
32            case 5:
33                sorting()
34            case 6:
35                rekomendasi()
36            case 7:
37                statistik()
38            case 8:
39                fmt.Print("Terima kasih sudah menggunakan aplikasi ini :) ")
40                return
41            default:
42                fmt.Println("Pilihan tidak valid, coba lagi")
43        }
44    }
45 }
46
47 }
48

```

Gambar 2.1 – Fungsi untuk menampilkan menu

Untuk lebih jelasnya, dapat melihat [ada tautan link yang sudah di sediakan.

Tantangan dan Solusi

- **Tantangan :**

1. Menentukan bagaimana konsep aplikasi workout ini disajikan, dalam konsep ini kita selaku pengembang bingung memilih data dalam satu hari atau dalam periode panjang.
2. Membuat agar tampilan data Riwayat olahraga menarik dan mudah dipahami user.
3. Bagaimana fitur rekomendasi dapat bekerja.

- **Solusi :**

1. Setelah berdiskusi bersama asisten praktikum, dan menelaah lebih lanjut deskripsi dan spesifikasinya, kita selaku pengembang memilih agar system aplikasi ini mencatat jadwal berkepanjangan. Pada kasus ini kita membuat data Dummy(data palsu) sebagai riwayat data latihan yang sudah dilakukan pengguna(user) sebelum memulai program
2. Pengembang memutuskan walaupun hasil data hanyalah berupa data pada terminal, tabel merupakan Solusi dimana user akan dapat mudah memahami data Riwayat tersebut.
3. Fitur rekomendasi disini bekerja agar setelah user menginputkan data Latihan yang dia lakukan, sistem akan melihat apakah kalori yang terbakar sudah memenuhi target atau belum, yaitu 500 kalori, dapat dilihat pada data dummy, system sudah menetapkan agar dalam 1 hari kalori yang di dapat bisa mencapai 500 kalori.

Kesimpulan dan Rekomendasi

Aplikasi ini sudah di sesuaikan dengan deskripsi dan spesifikasi yang diminta. Aplikasi ini akan dapat memanajemen data dalam menambah, mengubah, menghapus, dan menampilkan riwayat workout yang sudah dilakukan pengguna. Di dalam aplikasi ini juga terdapat fitur sorting dan searching untuk memudahkan pengguna, lalu memberikan data statistika dan memberikan rekomendasi latihan kepada pengguna.

Rekomendasi pengembangan aplikasi :

1. Sistem dapat diaplikasikan kedalam ui/ux kedepannya untuk dijadikan sebuah website atau aplikasi android/IOS.
2. Pembuatan algoritma dapat lebih rapih, dan terstruktur agar mendapat kecepatan yang optimal.

Tautan Source Kode

1. Link OneDrive : [Kelompok-17 Tubes-04](#)
2. Link Drive :
<https://drive.google.com/drive/folders/1wLCUJvJU9Bc5o4h2O214vXqPIYLk1A2j?usp=sharing>

Referensi

<https://dasarpemrogramangolang.novalagung.com/>
<https://www.youtube.com/@ProgrammerZamanNow>

Modul Praktikum Semester 1 & 2