

```
package main
```

```
import (  
    "fmt"  
    "strings"  
    "time"  
)
```

```
// struktur dari tanggal, kategori, nama, durasi, kalori latihan (opsi 1-bagian 2)
```

```
type Workout struct {  
    tanggalWorkout time.Time  
    kategoriWorkout string  
    namaLatihan    string  
    durasiLatihan  int  
    kalori         float64  
}
```

```
// struktur dari jenis kategori dan kalori yang dibakar per detik (opsi 1-bagian 1)
```

```
type kategoriWorkout struct {  
    jenisLatihan string  
    kaloriPerMenit float64  
}
```

```
// Opsi jenis latihan dimana kalori didapat dalam hitungan 'kalori per menit'
```

```
var jenisLatihan_kategori = map[string][]kategoriWorkout{  
    "Bahu-dan-Punggung": {  
        {"pull-up", 13.8},  
        {"bent-over row", 15.0},  
        {"face-pull", 12.0},  
        {"shrug", 10.8},  
        {"superman-hold", 9.6},  
    },  
}
```

```
},
"Lengan": {
  {"bicep-curl", 10.2},
  {"tricep-dip", 12.0},
  {"hammer-curl", 10.8},
  {"overhead-tricep-extension", 11.4},
  {"concentration-curl", 9.6},
},
"Dada": {
  {"push-up", 12.6},
  {"chest-fly", 13.2},
  {"bench-press", 15.0},
  {"incline-push-up", 12.0},
  {"cable-crossover", 14.4},
},
"Perut": {
  {"plank", 15.0},
  {"sit-up", 10.2},
  {"leg-raise", 12.0},
  {"russian-twist", 13.2},
  {"mountain-climber", 15.6},
},
"Kaki": {
  {"squat", 13.8},
  {"lunges", 13.2},
  {"jumping-squat", 16.2},
  {"calf-raise", 10.8},
  {"wall-sit", 11.4},
},
}
```

```
// membuat array (slice) global untuk menampung daftar latihan
```

```
var daftarLatihan []Workout
```

```
// Data Dummy (Fake Data) untuk riwayat workout 2 hari sebelumnya dengan manual
```

```
func dataDummy() {
```

```
    tanggal1 := time.Now().AddDate(0, 0, -2)
```

```
    tanggal2 := time.Now().AddDate(0, 0, -1)
```

```
    dummy1 := []Workout{
```

```
        {tanggal1, "Dada", "push-up", 6, 12.6 * 6},
```

```
        {tanggal1, "Lengan", "hammer-curl", 7, 10.8 * 7},
```

```
        {tanggal1, "Bahu-dan-Punggung", "shrug", 7, 10.8 * 7},
```

```
        {tanggal1, "Dada", "cable-crossover", 5, 14.4 * 5},
```

```
        {tanggal1, "Lengan", "tricep-dip", 6, 12.0 * 6},
```

```
        {tanggal1, "Bahu-dan-Punggung", "face-pull", 6, 12.0 * 6},
```

```
        {tanggal1, "Dada", "push-up", 6, 12.6 * 6},
```

```
    }
```

```
    dummy2 := []Workout{
```

```
        {tanggal2, "Perut", "plank", 8, 15.0 * 8},
```

```
        {tanggal2, "Kaki", "squat", 10, 13.8 * 10},
```

```
        {tanggal2, "Perut", "russian-twist", 6, 13.2 * 6},
```

```
        {tanggal2, "Kaki", "jumping-squat", 6, 16.2 * 6},
```

```
        {tanggal2, "Perut", "mountain-climber", 5, 15.6 * 5},
```

```
    }
```

```
    daftarLatihan = append(daftarLatihan, dummy1...)
```

```
    daftarLatihan = append(daftarLatihan, dummy2...)
```

```
}
```

```
// tampilkan data latihan
```

```

func tampilkanLatihan() {
    if len(daftarLatihan) == 0 {
        fmt.Println("Belum ada data latihan yang tersimpan.")
        return
    }

    fmt.Println("-----")
    fmt.Printf("| %-3s | %-10s | %-20s | %-20s | %-10s | %-10s |\n", "No", "Tanggal", "Jenis Latihan",
    "Kategori", "Durasi", "Kalori")
    fmt.Println("-----")

    for i := 0; i < len(daftarLatihan); i++ {
        latihan := daftarLatihan[i]
        fmt.Printf("| %-3d | %-10s | %-20s | %-20s | %-4d menit | %-10.2f |\n",
            i+1,
            latihan.tanggalWorkout.Format("02-01-2006"),
            latihan.namaLatihan,
            latihan.kategoriWorkout,
            latihan.durasiLatihan,
            latihan.kalori)
    }

    fmt.Println("----- \n")
}

// Fitur menu 1 -> menambah data latihan
func tambahLatihan() {
    var kategori = []string{"Bahu-dan-Punggung", "Lengan", "Dada", "Perut", "Kaki"}

    // User menginput kategori workout
    var pilihKategori int

```

```

fmt.Println("Kategori Workout: ")
for i := 0; i < len(kategori); i++ {
    fmt.Printf("%d. %s \n", i+1, kategori[i])
}

fmt.Print("Pilih Kategori Workout (1 - 5): ")
fmt.Scan(&pilihKategori)

jenisKategori := kategori[pilihKategori-1] //untuk program tahu index kategori mana
yang di pilih

listLatihan_kategori := jenisLatihan_kategori[jenisKategori] //menampilkan list latihan yang
ada pada kategori yang dipilih

// user menginput nama latihan yang tersedia di tiap kategori
var pilihLatihan int
for i := 0; i < len(listLatihan_kategori); i++ {
    fmt.Printf("%d. %s \n", i+1, listLatihan_kategori[i].jenisLatihan)
}

fmt.Print("Pilih jenis latihan yang ingin anda lakukan: ")
fmt.Scan(&pilihLatihan)

namaLatihan := listLatihan_kategori[pilihLatihan-1].jenisLatihan
kalori_menit := listLatihan_kategori[pilihLatihan-1].kaloriPerMenit

// user menginput , durasi dan program menghitung kalori
tanggal := time.Now()
var durasi int

fmt.Printf("Berapa lama anda latihan \"%s\" (dalam menit): ", namaLatihan)
fmt.Scan(&durasi)

kaloriTerbakar := kalori_menit * float64(durasi)

dataLatihan := Workout{

```

```

    tanggalWorkout: tanggal,
    kategoriWorkout: jenisKategori,
    namaLatihan: namaLatihan,
    durasiLatihan: durasi,
    kalori: kaloriTerbakar,
}

daftarLatihan = append(daftarLatihan, dataLatihan)
}

// Fitur menu 2 -> mengubah data latihan
func ubahLatihan() {
    tampilkanLatihan()

    var pilihNomor int
    fmt.Print("Pilih nomor latihan yang ingin diubah: ")
    fmt.Scan(&pilihNomor)

    fmt.Println("Masukan data baru :")
    var kategori = []string{"Bahu-dan-Punggung", "Lengan", "Dada", "Perut", "Kaki"}
    var pilihKategori int
    fmt.Println("Kategori Workout: ")
    for i := 0; i < len(kategori); i++ {
        fmt.Printf("%d. %s \n", i+1, kategori[i])
    }
    fmt.Print("Pilih Kategori Workout (1-5): ")
    fmt.Scan(&pilihKategori)

    jenisKategori := kategori[pilihKategori-1]
    listLatihan_kategori := jenisLatihan_kategori[jenisKategori]

```

```

var pilihLatihan int

fmt.Println("\nJenis Latihan:")

for i := 0; i < len(listLatihan_kategori); i++ {
    fmt.Printf("%d. %s \n", i+1, listLatihan_kategori[i].jenisLatihan)
}

fmt.Print("Pilih jenis latihan (1-", len(listLatihan_kategori), "): ")

fmt.Scan(&pilihLatihan)

namaLatihan := listLatihan_kategori[pilihLatihan-1].jenisLatihan
kalori_menit := listLatihan_kategori[pilihLatihan-1].kaloriPerMenit

var durasi int

fmt.Printf("\nBerapa lama anda latihan \"%s\" (dalam menit): ", namaLatihan)

fmt.Scan(&durasi)

kaloriTerbakar := kalori_menit * float64(durasi)

daftarLatihan[pilihNomor-1] = Workout{
    tanggalWorkout: time.Now(), // Tanggal diupdate ke hari ini
    kategoriWorkout: jenisKategori,
    namaLatihan:    namaLatihan,
    durasiLatihan:  durasi,
    kalori:         kaloriTerbakar,
}

fmt.Println("\nData latihan berhasil diubah!")
}

// Fitur menu 3 -> hapus data latihan
func hapusLatihan() {
    if len(daftarLatihan) == 0 {

```

```

    fmt.Println("Tidak data latihan.")
    return
}

tampilkanLatihan()

var nomorLatihan int
fmt.Print("Masukkan nomor latihan yang ingin dihapus: ")
fmt.Scan(&nomorLatihan)
if nomorLatihan < 1 || nomorLatihan > len(daftarLatihan) {
    fmt.Println("Nomor yang dimasukkan tidak ada.")
    return
}

for i := nomorLatihan - 1; i < len(daftarLatihan)-1; i++ {
    daftarLatihan[i] = daftarLatihan[i+1]
}
daftarLatihan = daftarLatihan[:len(daftarLatihan)-1]
fmt.Println("\nData latihan berhasil dihapus. \n")
}

// < --- START SERACHING ---> //

// Fungsi untuk mencari latihan berdasarkan jenis menggunakan sequential search
func Sequential(data []Workout, target string) int { // fungsi sequential search (Digunakan
dengan memeriksa satu per satu dari awal hingga akhir)
    for i := 0; i < len(data); i++ {
        if data[i].namaLatihan == target {
            return i
        }
    }
}

```



```
    return -1
}
```

// Fungsi untuk mencari latihan berdasarkan jenis menggunakan binary search

```
func Binary(data []Workout, target string) int { // fungsi binary search (Digunakan dengan memeriksa data yang berada ditengah)
```

```
    awal := 0
```

```
    akhir := len(data) - 1
```

```
    for awal <= akhir {
```

```
        pembagi := (awal + akhir) / 2
```

```
        if strings.ToLower(data[pembagi].namaLatihan) == target {
```

```
            return pembagi
```

```
        } else if strings.ToLower(data[pembagi].namaLatihan) < target {
```

```
            awal = pembagi + 1
```

```
        } else {
```

```
            akhir = pembagi - 1
```

```
        }
```

```
    }
```

```
    return -1
```

```
}
```

```
func UrutkanNama(data []Workout) { // Fungsi mengurutkan nama latihan
```

```
    n := len(data)
```

```
    for i := 0; i < n-1; i++ {
```

```
        for j := 0; j < n-1-i; j++ {
```

```
            if strings.ToLower(data[j].namaLatihan) > strings.ToLower(data[j+1].namaLatihan) {
```

```
                data[j], data[j+1] = data[j+1], data[j]
```

```
            }
```

```
        }
```

```
}  
}
```

```
// Fitur menu 4 -> cari data latihan
```

```
func LatihanYangDicari() {
```

```
    if len(daftarLatihan) == 0 {
```

```
        fmt.Println("Belum ada data latihan untuk dicari.")
```

```
        return
```

```
    }
```

```
    fmt.Print("Masukkan nama latihan: ")
```

```
    var Dicari string
```

```
    fmt.Scan(&Dicari)
```

```
    Dicari = strings.ToLower(Dicari)
```

```
    fmt.Println("1. Metode Sequential")
```

```
    fmt.Println("2. Metode Binary")
```

```
    fmt.Print("Pilih metode pencarian: ")
```

```
    var metode int
```

```
    fmt.Scan(&metode)
```

```
    var hasil []Workout
```

```
    if metode == 1 {
```

```
        for i := 0; i < len(daftarLatihan); i++ {
```

```
            if strings.ToLower(daftarLatihan[i].namaLatihan) == Dicari {
```

```
                hasil = append(hasil, daftarLatihan[i])
```

```
            }
```

```
        }
```

```
    } else if metode == 2 {
```

```
        salinan := make([]Workout, len(daftarLatihan))
```

```

copy(salinan, daftarLatihan)

UrutkanNama(salinan)

index := Binary(salinan, Dicari)

if index != -1 {
    hasil = append(hasil, salinan[index])
}
} else {
    fmt.Println("Metode pencarian tidak valid.")
    return
}

// Tampilkan hasil
if len(hasil) > 0 {
    fmt.Println("Hasil Pencarian:")
    fmt.Println("-----")

    fmt.Printf("| %-3s | %-10s | %-20s | %-20s | %-10s | %-10s |\n", "No", "Tanggal", "Jenis
Latihan", "Kategori", "Durasi", "Kalori")
    fmt.Println("-----")

    for i := 0; i < len(hasil); i++ {
        fmt.Printf("| %-3d | %-10s | %-20s | %-20s | %-4d menit | %-10.2f |\n",
            i+1,
            hasil[i].tanggalWorkout.Format("02-01-2006"),
            hasil[i].namaLatihan,
            hasil[i].kategoriWorkout,
            hasil[i].durasiLatihan,
            hasil[i].kalori)
    }

    fmt.Println("-----")
} else {
    fmt.Println("Latihan dengan nama tersebut tidak ditemukan.")
}

```

```
}  
}
```

```
// < --- END SEARCHING ---> //
```

```
// < --- START SORTING ---> //
```

```
var daftarLatihanBackup []Workout // backup data asli sebelum di sorting
```

```
// Selection sort "Durasi"
```

```
func selectionSort(data []Workout, ascending bool) {  
    n := len(data)  
    for i := 0; i < n-1; i++ {  
        idx := i  
        for j := i + 1; j < n; j++ {  
            if ascending {  
                if data[j].durasiLatihan < data[idx].durasiLatihan {  
                    idx = j  
                }  
            } else {  
                if data[j].durasiLatihan > data[idx].durasiLatihan {  
                    idx = j  
                }  
            }  
        }  
        data[i], data[idx] = data[idx], data[i]  
    }  
}
```

```
// Selection sort "Kalori"
```

```
func insertionSortKalori(data []Workout, ascending bool) {
```

```

n := len(data)
for i := 1; i < n; i++ {
    temp := data[i]
    j := i - 1

    if ascending {
        for j >= 0 && data[j].kalori > temp.kalori {
            data[j+1] = data[j]
            j--
        }
    } else {
        for j >= 0 && data[j].kalori < temp.kalori {
            data[j+1] = data[j]
            j--
        }
    }

    data[j+1] = temp
}

```

// Fitur menu 5 -> sorting kalori dan durasi

```

func sorting() {

    if len(daftarLatihanBackup) == 0 {
        daftarLatihanBackup = make([]Workout, len(daftarLatihan))
        copy(daftarLatihanBackup, daftarLatihan)
    }

    var pilihJenis int
    var urutan int

```

```

fmt.Println("Pilih jenis sorting:")
fmt.Println("1. Durasi (Selection Sort)")
fmt.Println("2. Kalori (Insertion Sort)")
fmt.Println("3. Kembali ke data awal")
fmt.Print("Pilih menu di atas: ")
fmt.Scan(&pilihJenis)

if pilihJenis == 3 {
    if len(daftarLatihanBackup) > 0 {
        daftarLatihan = make([]Workout, len(daftarLatihanBackup))
        copy(daftarLatihan, daftarLatihanBackup)
    }
    return
}

fmt.Println("Urutan:")
fmt.Println("1. Dari yang terkecil")
fmt.Println("2. Dari yang terbesar")
fmt.Print("Masukkan pilihan urutan: ")
fmt.Scan(&urutan)

ascending := (urutan == 1)

salinanData := make([]Workout, len(daftarLatihan))
copy(salinanData, daftarLatihan)

if pilihJenis == 1 {
    selectionSort(salinanData, ascending)
} else if pilihJenis == 2 {
    insertionSortKalori(salinanData, ascending)
}

```

```

daftarLatihan = salinanData
}

// < --- END SORTING ---> //

// Fitur menu 6 -> rekomendasi latihan
func rekomendasi() {
    tanggalTerakhir := daftarLatihan[len(daftarLatihan)-1].tanggalWorkout.Format("02-01-2006")

    var bodyAtas, bodyBawah int
    for _, data := range daftarLatihan {
        if data.tanggalWorkout.Format("02-01-2006") == tanggalTerakhir {
            kategori := strings.ToLower(data.kategoriWorkout)
            if kategori == "bahu-dan-punggung" || kategori == "dada" || kategori == "lengan" {
                bodyAtas++
            } else if kategori == "perut" || kategori == "kaki" {
                bodyBawah++
            }
        }
    }

    var targetKategoris []string
    if bodyAtas > bodyBawah {
        targetKategoris = []string{"Bahu-dan-Punggung", "Dada", "Lengan"}
        fmt.Println("Hari ini dominan latihan upper body, rekomendasi dari kategori: ",
            targetKategoris)
    } else {
        targetKategoris = []string{"Perut", "Kaki"}
        fmt.Println("Hari ini dominan latihan lower body, rekomendasi dari kategori: ",
            targetKategoris)
    }
}

```

```

//Gabungkan seluruh jenis latihan
var semuaLatihan []kategoriWorkout
for _, kategori := range targetKategoris {
    semuaLatihan = append(semuaLatihan, jenisLatihan_kategori[kategori]...)
}

// Urutan berdasarkan kalori per menit tertinggi pada data kategorie
for i := 0; i < len(semuaLatihan)-1; i++ {
    maxIdx := i
    for j := i + 1; j < len(semuaLatihan); j++ {
        if semuaLatihan[j].kaloriPerMenit > semuaLatihan[maxIdx].kaloriPerMenit {
            maxIdx = j
        }
    }
    semuaLatihan[i], semuaLatihan[maxIdx] = semuaLatihan[maxIdx], semuaLatihan[i]
}

// Rekomendasi hingga 500 kalori
var totalKalori float64
var rekomendasi []Workout

for _, item := range semuaLatihan {
    if totalKalori >= 500 {
        break
    }
    sisaKalori := 500 - totalKalori

    durasi := int(sisaKalori / item.kaloriPerMenit)
    if durasi < 3 {
        durasi = 3
    }
}

```



```

    } else if durasi > 15 {
        durasi = 15
    }

    kalori := item.kaloriPerMenit * float64(durasi)

// Cari kategori latihan
var kategoriLatihan string
for _, kategori := range targetKategoris {
    for _, latihan := range jenisLatihan_kategori[kategori] {
        if latihan.jenisLatihan == item.jenisLatihan {
            kategoriLatihan = kategori
        }
    }
}

rekomendasi = append(rekomendasi, Workout{
    namaLatihan: item.jenisLatihan,
    kategoriWorkout: kategoriLatihan,
    durasiLatihan: durasi,
    kalori: kalori,
})
totalKalori += kalori
}

// Output
fmt.Printf("\n 📋 Rekomendasi Latihan (Target ~500 Kalori)\n")
fmt.Println("-----")
fmt.Printf("| %-20s | %-20s | %-10s | %-10s |\n", "Jenis Latihan", "Kategori", "Durasi", "Kalori")
fmt.Println("-----")
for _, item := range rekomendasi {
    fmt.Printf("| %-20s | %-20s | %-4d menit | %-10.2f |\n",

```

```

        item.namaLatihan,
        item.kategoriWorkout,
        item.durasiLatihan,
        item.kalori)
    }
    fmt.Println("-----")
    fmt.Printf("Total estimasi kalori: %.2f kalori\n\n", totalKalori)
}

// Fitur menu 8 -> tampilkan statistik
func statistik() {
    fmt.Println("Statistik akan menampilkan 10 aktivitas terakhir dan juga data latihan per hari")
    fmt.Println("dan anda dapat melihat total kalori dalam periode tertentu \n")

    var tanggalAwal, tanggalAkhir string
    fmt.Print("\nMasukkan tanggal mulai (DD-MM-YYYY): ")
    fmt.Scan(&tanggalAwal)
    fmt.Print("Masukkan tanggal akhir (DD-MM-YYYY): ")
    fmt.Scan(&tanggalAkhir)

    tMulai, err1 := time.Parse("02-01-2006", tanggalAwal)
    tAkhir, err2 := time.Parse("02-01-2006", tanggalAkhir)

    if err1 != nil || err2 != nil {
        fmt.Println("Format tanggal salah. Harus DD-MM-YYYY.")
        return
    }

    // --- Tampilkan 10 Aktivitas Terakhir ---
    fmt.Println("10 Aktivitas Terakhir:")
    fmt.Println("-----")

```

```
fmt.Printf("| %-3s | %-10s | %-20s | %-20s | %-10s | %-10s |\n", "No", "Tanggal", "Jenis Latihan",  
"Kategori", "Durasi", "Kalori")
```

```
fmt.Println("-----")
```

```
sumberData := daftarLatihanBackup
```

```
if len(sumberData) == 0 {
```

```
    sumberData = daftarLatihan
```

```
}
```

```
jumlahData := len(sumberData)
```

```
start := jumlahData - 10
```

```
if start < 0 {
```

```
    start = 0
```

```
}
```

```
no := 1
```

```
for i := start; i < jumlahData; i++ {
```

```
    data := sumberData[i]
```

```
    fmt.Printf("| %-3d | %-10s | %-20s | %-20s | %-4d menit | %-10.2f |\n",
```

```
        no,
```

```
        data.tanggalWorkout.Format("02-01-2006"),
```

```
        data.namaLatihan,
```

```
        data.kategoriWorkout,
```

```
        data.durasiLatihan,
```

```
        data.kalori)
```

```
}
```

```
fmt.Println("-----")
```

```
var totalKalori float64
```

```
for _, latihan := range daftarLatihan {
```

```
    if !latihan.tanggalWorkout.Before(tMulai) && !latihan.tanggalWorkout.After(tAkhir) {
```

```

        totalKalori += latihan.kalori
    }
}

fmt.Printf("\nTotal Kalori dalam periode %s sampai %s: %.2f kalori\n\n",
    tMulai.Format("02-01-2006"), tAkhir.Format("02-01-2006"), totalKalori)

// --- Statistik per Hari ---
statistikPerHari := make(map[string]struct {
    jumlahLatihan int
    totalKalori   float64
    totalDurasi   int
})

for _, data := range daftarLatihan {
    keyTanggal := data.tanggalWorkout.Format("02-01-2006")
    stat := statistikPerHari[keyTanggal]
    stat.jumlahLatihan++
    stat.totalDurasi += data.durasiLatihan
    stat.totalKalori += data.kalori
    statistikPerHari[keyTanggal] = stat
}

fmt.Println("\nStatistik Per Hari:")
fmt.Println("-----")
for tanggal, data := range statistikPerHari {
    fmt.Println("Tanggal      :", tanggal)
    fmt.Println("Jumlah Latihan :", data.jumlahLatihan)
    fmt.Printf("Total Durasi   : %d menit\n", data.totalDurasi)
    fmt.Printf("Jumlah Kalori  : %.2f kalori\n", data.totalKalori)
    fmt.Println()
}

```

```

        no++
    }

    fmt.Println("-----")
}

```

```

func main() {
    var pilihMenu int

    dataDummy()

    for {
        tampilkanLatihan()

        fmt.Println("===== Aplikasi Workout =====")
        fmt.Println("1. Tambah Data Latihan")
        fmt.Println("2. Ubah Data Latihan")
        fmt.Println("3. Hapus Data Latihan")
        fmt.Println("4. Cari Jenis Latihan") // ojan yg service
        fmt.Println("5. Urutkan Data Latihan")
        fmt.Println("6. Rekomendasi Latihan")
        fmt.Println("7. Tampilkan Statistik")
        fmt.Println("8. Keluar")

        fmt.Print("Silahkan pilih menu yang ingin anda gunakan: ")
        fmt.Scan(&pilihMenu)
        fmt.Println()

        switch pilihMenu {
        case 1:

```

```
        tambahLatihan()
    case 2:
        ubahLatihan()
    case 3:
        hapusLatihan()
    case 4:
        LatihanYangDicari()
    case 5:
        sorting()
    case 6:
        rekomendasi()
    case 7:
        statistik()
    case 8:
        fmt.Print("Terima kasih sudah menggunakan aplikasi ini :) ")
        return
    default:
        fmt.Println("Pilihan tidak valid, coba lagi")
    }

}

}
```