Nama : Daffa Aryaputra
Nim : 2311102272
Kelas : 11IF06

Modul 15

Soal 1

Code :

```
package main

import "fmt"
```

```
type set [2022]int
```

```
// Function to check if a value exists in a set
func exist(T set, n int, val int) bool {
    for i := 0; i < n; i++ {
        if T[i] == val {
            return true
        }
    }
    return false
}
```

```
// Function to input values into a set
func inputSet(T *set, n *int) {
    var x int
    for {
        fmt.Scan(&x)
        if exist(*T, *n, x) {
            continue // Skip duplicate values
        }
        if x == -1 { // -1 indicates the end of input
            break
        }
        T[*n] = x
        *n++
    }
}
```

```
// Function to find the intersection of two sets
func findIntersection(T1 set, n1 int, T2 set, n2 int, T3 *set, n3 *int)
{
    for i := 0; i < n1; i++ {
        if exist(T2, n2, T1[i]) {
            (*T3)[*n3] = T1[i]
            *n3++
        }
    }
}
```

```go
// Function to print a set
func printSet(T set, n int) {
    for i := 0; i < n; i++ {
        if i > 0 {
            fmt.Print(" ")
        }
        fmt.Print(T[i])
    }
    fmt.Println()
}
```

```go
func main() {
    var s1_2311102272, s2_Daffa, s3_IF set
    var n1, n2, n3 int
```

```go
    fmt.Println("Input set 1 (end with -1):")
    inputSet(&s1_2311102272, &n1)
```

```go
    fmt.Println("Input set 2 (end with -1):")
    inputSet(&s2_Daffa, &n2)
```

```go
    findIntersection(s1_2311102272, n1, s2_Daffa, n2, &s3_IF, &n3)
```

```go
    fmt.Println("Intersection of sets:")
    printSet(s3_IF, n3)
}
```

```go
package main

import "fmt"

type set [2022]int

// Function to check if a value exists in a set
func exist(T set, n int, val int) bool {
    for i := 0; i < n; i++ {
        if T[i] == val {
            return true
        }
    }
    return false
}

// Function to input values into a set
func inputSet(T *set, n *int) {
    var x int
    for {
        fmt.Scan(&x)
        if exist(*T, *n, x) {
            continue // Skip duplicate values
        }
        if x == -1 { // -1 indicates the end of input
            break
        }
        T[*n] = x
        *n++
    }
}

// Function to find the intersection of two sets
func findIntersection(T1 set, n1 int, T2 set, n2 int, T3 *set, n3 *int) {
    for i := 0; i < n1; i++ {
        if exist(T2, n2, T1[i]) {
            (*T3)[*n3] = T1[i]
            *n3++
        }
    }
}

// Function to print a set
func printSet(T set, n int) {
    for i := 0; i < n; i++ {
        if i > 0 {
            fmt.Print(" ")
        }
        fmt.Print(T[i])
    }
    fmt.Println()
}

func main() {
    var s1_2311102272, s2_Daffa, s3_IF set
    var n1, n2, n3 int

    fmt.Println("Input set 1 (end with -1):")
    inputSet(&s1_2311102272, &n1)

    fmt.Println("Input set 2 (end with -1):")
    inputSet(&s2_Daffa, &n2)

    findIntersection(s1_2311102272, n1, s2_Daffa, n2, &s3_IF, &n3)

    fmt.Println("Intersection of sets:")
    printSet(s3_IF, n3)
}
```

```
PS C:\Users\ACER> go run "e:\golang semester 3\modul15.go"
Input set 1 (end with -1):
1 1 -1
Input set 2 (end with -1):
1 1 -1
Intersection of sets:
1
PS C:\Users\ACER>
```

Soal 2

Code :

```
package main

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

```
// Konstanta untuk batas maksimum mahasiswa
const NMax = 51
```

```
// Struct untuk menyimpan data mahasiswa
type Mahasiswa struct {
    NIM   string
    Nama  string
    Nilai int
}
```

```
// Array untuk menyimpan data mahasiswa
type ArrayMahasiswa [NMax]Mahasiswa
```

```
func main() {
    var mahasiswaList_2311102272 ArrayMahasiswa
    var n int
```

```
    // Memasukkan jumlah data mahasiswa
    fmt.Print("Masukkan jumlah mahasiswa (max ", NMax, "): ")
    fmt.Scan(&n)
```

```
    if n > NMax || n < 1 {
        fmt.Println("Jumlah mahasiswa tidak valid!")
        return
    }
```

```
    // Memasukkan data mahasiswa
    for i := 0; i < n; i++ {
        fmt.Println("Masukkan data mahasiswa ke-", i+1)
        fmt.Print("NIM: ")
        fmt.Scan(&mahasiswaList_2311102272[i].NIM)
        fmt.Print("Nama: ")
        fmt.Scan(&mahasiswaList_2311102272[i].Nama)
        fmt.Print("Nilai: ")
        fmt.Scan(&mahasiswaList_2311102272[i].Nilai)
    }
```

```
    // Menampilkan semua data mahasiswa
    fmt.Println("\nData Mahasiswa:")
    for i := 0; i < n; i++ {
```

```
        fmt.Printf("NIM: %s, Nama: %s, Nilai: %d\n",
mahasiswaList_2311102272[i].NIM, mahasiswaList_2311102272[i].Nama,
mahasiswaList_2311102272[i].Nilai)
    }
```

```
    // Mencari mahasiswa berdasarkan NIM
    fmt.Print("\nMasukkan NIM untuk pencarian: ")
    var cariNIM string
    fmt.Scan(&cariNIM)
```

```
    index := cariMahasiswa(mahasiswaList_2311102272, n, cariNIM)
    if index == -1 {
        fmt.Println("Mahasiswa dengan NIM tersebut tidak ditemukan.")
    } else {
        fmt.Printf("Mahasiswa ditemukan: NIM: %s, Nama: %s,
Nilai: %d\n",
            mahasiswaList_2311102272[index].NIM,
mahasiswaList_2311102272[index].Nama,
mahasiswaList_2311102272[index].Nilai)
```

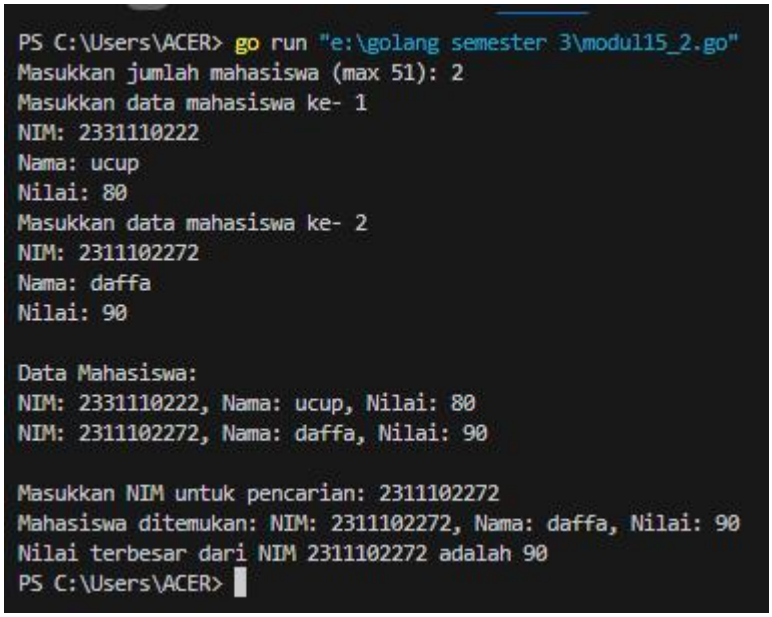
```
        // Mencari nilai terbesar dengan NIM tertentu
        maxNilai := cariNilaiTertinggi(mahasiswaList_2311102272, n,
cariNIM)
        fmt.Printf("Nilai terbesar dari NIM %s adalah %d\n", cariNIM,
maxNilai)
    }
}
```

```
// Fungsi untuk mencari mahasiswa berdasarkan NIM
func cariMahasiswa(arr ArrayMahasiswa, n int, nim string) int {
    for i := 0; i < n; i++ {
        if strings.EqualFold(arr[i].NIM, nim) {
            return i
        }
    }
    return -1
}
```

```
// Fungsi untuk mencari nilai terbesar berdasarkan NIM
func cariNilaiTertinggi(arr ArrayMahasiswa, n int, nim string) int {
    maxNilai := -1
    for i := 0; i < n; i++ {
        if strings.EqualFold(arr[i].NIM, nim) {
            if arr[i].Nilai > maxNilai {
                maxNilai = arr[i].Nilai
            }
        }
    }
    return maxNilai
```

```
}
```


package main
import (
"fmt"
"strings"
)
// Konstanta untuk batas maksimal mahasiswa
const NMax = 51
// Struct untuk menyimpan data mahasiswa
type Mahasiswa struct {
NIM string
Nama string
Nilai int
}
// Array untuk menyimpan data mahasiswa
type ArrayMahasiswa [NMax]Mahasiswa
func main() {
var mahasiswaList_2311302272 ArrayMahasiswa
var n int
// Memasukkan jumlah data mahasiswa
fmt.Print("Masukkan jumlah mahasiswa (max ", NMax, "): ")
fmt.Scan(&n)
if n > NMax || n < 1 {
fmt.Println("Jumlah mahasiswa tidak valid!")
return
}
// Memasukkan data mahasiswa
for i := 0; i < n; i++ {
fmt.Println("Masukkan data mahasiswa ke-", i+1)
fmt.Print("NIM: ")
fmt.Scan(&mahasiswaList_2311302272[i].NIM)
fmt.Print("Nama: ")
fmt.Scan(&mahasiswaList_2311302272[i].Nama)
fmt.Print("Nilai: ")
fmt.Scan(&mahasiswaList_2311302272[i].Nilai)
}
// Menampilkan semua data mahasiswa
fmt.Println("\nData Mahasiswa:")
for i := 0; i < n; i++ {
fmt.Printf("NIM: %s, Nama: %s, Nilai: %d\n", mahasiswaList_2311302272[i].NIM, mahasiswaList_2311302272[i].Nama, mahasiswaList_2311302272[i].Nilai)
}
// Mencari mahasiswa berdasarkan NIM
fmt.Print("\nMasukkan NIM untuk pencarian: ")
var cariNIM string
fmt.Scan(&cariNIM)
indeks := cariMahasiswa(mahasiswaList_2311302272, n, cariNIM)
if indeks == -1 {
fmt.Println("Mahasiswa dengan NIM tersebut tidak ditemukan.")
} else {
fmt.Printf("Mahasiswa ditemukan: NIM: %s, Nama: %s, Nilai: %d\n",
mahasiswaList_2311302272[indeks].NIM, mahasiswaList_2311302272[indeks].Nama, mahasiswaList_2311302272[indeks].Nilai)
// Mencari nilai tertinggi dengan NIM tertentu
maxNilai := cariNilaiTertinggi(mahasiswaList_2311302272, n, cariNIM)
fmt.Printf("Nilai tertinggi dari NIM %s adalah %d\n", cariNIM, maxNilai)
}
}
// Fungsi untuk mencari mahasiswa berdasarkan NIM
func cariMahasiswa(arr ArrayMahasiswa, n int, nim string) int {
for i := 0; i < n; i++ {
if strings.EqualFold(arr[i].NIM, nim) {
return i
}
}
return -1
}
// Fungsi untuk mencari nilai tertinggi berdasarkan NIM
func cariNilaiTertinggi(arr ArrayMahasiswa, n int, nim string) int {
maxNilai := -1
for i := 0; i < n; i++ {
if strings.EqualFold(arr[i].NIM, nim) {
if arr[i].Nilai > maxNilai {
maxNilai = arr[i].Nilai
}
}
}
return maxNilai
}


PS C:\Users\ACER> go run "e:\golang semester 3\modul15_2.go"
Masukkan jumlah mahasiswa (max 51): 2
Masukkan data mahasiswa ke- 1
NIM: 2331110222
Nama: ucup
Nilai: 80
Masukkan data mahasiswa ke- 2
NIM: 2311102272
Nama: daffa
Nilai: 90
Data Mahasiswa:
NIM: 2331110222, Nama: ucup, Nilai: 80
NIM: 2311102272, Nama: daffa, Nilai: 90
Masukkan NIM untuk pencarian: 2311102272
Mahasiswa ditemukan: NIM: 2311102272, Nama: daffa, Nilai: 90
Nilai terbesar dari NIM 2311102272 adalah 90
PS C:\Users\ACER>

Soal 3

Code :

```
package main

import (
    "fmt"
    "math"
)
```

```
const nProv = 34 // Jumlah provinsi yang ditangani program
```

```
// Struct untuk menyimpan data tentang provinsi
type Provinsi struct {
    nama     string  // Nama provinsi
    populasi int     // Populasi provinsi
    tumbuh   float64 // Tingkat pertumbuhan populasi (dalam format
desimal, misalnya 0.03 untuk 3%)
}
```

```
func main() {
    var prov [nProv]Provinsi // Array untuk menyimpan data semua
provinsi
    var namaDicari string    // Nama provinsi yang akan dicari
```

```
    // Input data provinsi
    for i := 0; i < nProv; i++ {
        fmt.Printf("Masukkan data provinsi ke-%d (nama populasi
tingkat_pertumbuhan): ", i+1)
        fmt.Scan(&prov[i].nama, &prov[i].populasi, &prov[i].tumbuh)
    }
    fmt.Print("Masukkan nama provinsi yang ingin dicari: ")
    fmt.Scan(&namaDicari) // Membaca nama provinsi yang dicari
```

```
    // Menampilkan nama provinsi dengan angka pertumbuhan tercepat
    indeksTercepat := provinsiTercepat_2311102272(prov)
    fmt.Printf("Provinsi dengan pertumbuhan tercepat: %s\n",
prov[indeksTercepat].nama)
```

```
    // Menampilkan indeks provinsi yang dicari
    indeksDicari := indeksProvinsi(prov, namaDicari)
    if indeksDicari != -1 {
        fmt.Printf("Indeks provinsi %s: %d\n", namaDicari, indeksDicari)
    } else {
        fmt.Printf("Provinsi %s tidak ditemukan.\n", namaDicari)
    }
```

```
    // Menampilkan prediksi jumlah penduduk untuk provinsi dengan
pertumbuhan di atas 2%
```

```
    fmt.Println("Prediksi populasi untuk provinsi dengan pertumbuhan di
atas 2%:")
    prediksi(prov)
}
```

```
// Fungsi untuk menemukan indeks provinsi dengan tingkat pertumbuhan
tercepat
func provinsiTercepat_2311102272(prov [nProv]Provinsi) int {
    maxIndex := 0 // Asumsikan provinsi pertama memiliki pertumbuhan
tercepat
    for i := 1; i < nProv; i++ {
        if prov[i].tumbuh > prov[maxIndex].tumbuh {
            maxIndex = i
        }
    }
    return maxIndex
}
```

```
// Fungsi untuk mencari indeks provinsi berdasarkan nama
func indeksProvinsi(prov [nProv]Provinsi, nama string) int {
    for i := 0; i < nProv; i++ {
        if prov[i].nama == nama {
            return i // Mengembalikan indeks jika ditemukan
        }
    }
    return -1 // Jika tidak ditemukan
}
```

```
// Fungsi untuk menampilkan prediksi populasi untuk provinsi dengan
tingkat pertumbuhan di atas 2%
func prediksi(prov [nProv]Provinsi) {
    for i := 0; i < nProv; i++ {
        if prov[i].tumbuh > 0.02 { // Cek jika tingkat pertumbuhan
lebih dari 2%
            prediksiPopulasi := int(math.Round(float64(prov[i].populasi)
* (1 + prov[i].tumbuh)))
            fmt.Printf("%s: %d\n", prov[i].nama, prediksiPopulasi)
        }
    }
}
```

```go
package main

import (
    "fmt"
    "math"
)

const nProv = 34 // Jumlah provinsi yang ditangani program

// Struct untuk menyimpan data tentang provinsi
type Provinsi struct {
    nama     string  // Nama provinsi
    populasi int     // Populasi provinsi
    tumbuh   float64 // Tingkat pertumbuhan populasi (dalam format desimal, misalnya 0.03 untuk 3%)
}

func main() {
    var prov [nProv]Provinsi // Array untuk menyimpan data semua provinsi
    var namaDicari string    // Nama provinsi yang akan dicari

    // Input data provinsi
    for i := 0; i < nProv; i++ {
        fmt.Printf("Masukkan data provinsi ke-%d (nama populasi tingkat_pertumbuhan): ", i+1)
        fmt.Scan(&prov[i].nama, &prov[i].populasi, &prov[i].tumbuh)
    }
    fmt.Print("Masukkan nama provinsi yang ingin dicari: ")
    fmt.Scan(&namaDicari) // Membaca nama provinsi yang dicari

    // Menampilkan nama provinsi dengan angka pertumbuhan tersepat
    indeksTercepat := provinsiTercepat_2311102272(prov)
    fmt.Printf("Provinsi dengan pertumbuhan tersepat: %s\n", prov[indeksTercepat].nama)

    // Menampilkan indeks provinsi yang dicari
    indeksDicari := indeksProvinsi(prov, namaDicari)
    if indeksDicari != -1 {
        fmt.Printf("Indeks provinsi %s: %d\n", namaDicari, indeksDicari)
    } else {
        fmt.Printf("Provinsi %s tidak ditemukan.\n", namaDicari)
    }

    // Menampilkan prediksi jumlah penduduk untuk provinsi dengan pertumbuhan di atas 2%
    fmt.Println("Prediksi populasi untuk provinsi dengan pertumbuhan di atas 2%:")
    prediksi(prov)
}

// Fungsi untuk menemukan indeks provinsi dengan tingkat pertumbuhan tersepat
func provinsiTercepat_2311102272(prov [nProv]Provinsi) int {
    maxIndex := 0 // Asumsikan provinsi pertama memiliki pertumbuhan tersepat
    for i := 1; i < nProv; i++ {
        if prov[i].tumbuh > prov[maxIndex].tumbuh {
            maxIndex = i
        }
    }
    return maxIndex
}

// Fungsi untuk mencari indeks provinsi berdasarkan nama
func indeksProvinsi(prov [nProv]Provinsi, nama string) int {
    for i := 0; i < nProv; i++ {
        if prov[i].nama == nama {
            return i // Mengembalikan indeks jika ditemukan
        }
    }
    return -1 // Jika tidak ditemukan
}

// Fungsi untuk menampilkan prediksi populasi untuk provinsi dengan tingkat pertumbuhan di atas 2%
func prediksi(prov [nProv]Provinsi) {
    for i := 0; i < nProv; i++ {
        if prov[i].tumbuh > 0.02 { // Cek jika tingkat pertumbuhan lebih dari 2%
            prediksiPopulasi := int(math.Round(float64(prov[i].populasi) * (1 + prov[i].tumbuh)))
            fmt.Printf("%s: %d\n", prov[i].nama, prediksiPopulasi)
        }
    }
}
```

```
PROBLEMS 131   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\ACER> go run "e:\golang semester 3\modul15_3.go"
Masukkan data provinsi ke-1 (nama populasi tingkat_pertumbuhan): Jakarta 10000 0,03
Masukkan data provinsi ke-2 (nama populasi tingkat_pertumbuhan): Bandung 20000 0,02
```

Soal 4

Code :

```
package main

import (
    "fmt"
    "sort"
)
```

```
func main() {
    var input_2311102272 int
    var data []int
```

```
    fmt.Println("Masukkan bilangan (akhiri dengan -5313541):")
    for {
        fmt.Scan(&input_2311102272)
```

```
        if input_2311102272 == -5313541 {
            break
        }
```

```
        if input_2311102272 == 0 {
            if len(data) > 0 {
                sort.Ints(data)
                median := calculateMedian(data)
                fmt.Printf("Median: %d\n", median)
            }
        } else {
            data = append(data, input_2311102272)
        }
    }
}
```

```
func calculateMedian(data []int) int {
    n := len(data)
    if n%2 == 1 {
        return data[n/2]
    }
    return (data[(n/2)-1] + data[n/2]) / 2
}
```

: > golang semester 3 > modul15_4.go > main

```go
package main

import (
    "fmt"
    "sort"
)

func main() {
    var input_2311102272 int
    var data []int

    fmt.Println("Masukkan bilangan (akhiri dengan -5313541):")
    for {
        fmt.Scan(&input_2311102272)

        if input_2311102272 == -5313541 {
            break
        }

        if input_2311102272 == 0 {
            if len(data) > 0 {
                sort.Ints(data)
                median := calculateMedian(data)
                fmt.Printf("Median: %d\n", median)
            }
        } else {
            data = append(data, input_2311102272)
        }
    }
}

func calculateMedian(data []int) int {
    n := len(data)
    if n%2 == 1 {
        return data[n/2]
    }
    return (data[(n/2)-1] + data[n/2]) / 2
}
```

```
PS C:\Users\ACER> go run "e:\golang semester 3\modul15_4.go"
Masukkan bilangan (akhiri dengan -5313541):
7 23 11 0 5 19 2 29 3 13 17 0 -5313541
Median: 11
Median: 12
PS C:\Users\ACER>
```

Soal 5

Code :

```
package main

import (
    "fmt"
    "sort"
)
```

```
const NMAX = 1000000
```

```
// Struct untuk menyimpan data partai
type Partai struct {
    ID    int
    Suara int
}
```

```
// Tipe untuk array partai
type TabPartai []Partai
```

```
func main() {
    var tabPartai_2311102272 TabPartai
    var suara int
    partaiMap := make(map[int]int)
```

```
    fmt.Println("Masukkan perolehan suara untuk setiap partai (akhiri
dengan -1):")
```

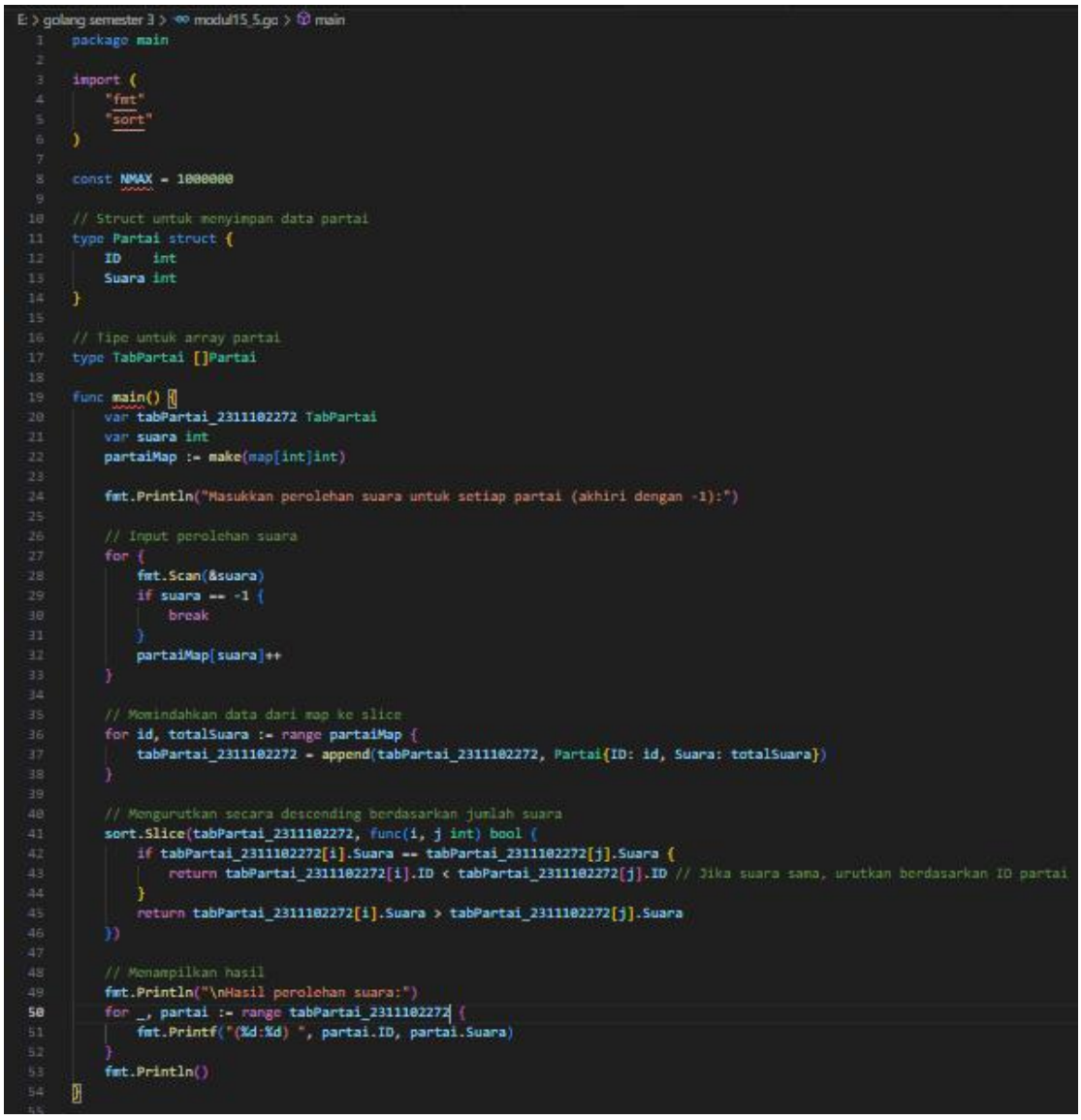
```
    // Input perolehan suara
    for {
        fmt.Scan(&suara)
        if suara == -1 {
            break
        }
        partaiMap[suara]++
    }
```

```
    // Memindahkan data dari map ke slice
    for id, totalSuara := range partaiMap {
        tabPartai_2311102272 = append(tabPartai_2311102272, Partai{ID:
id, Suara: totalSuara})
    }
```

```
    // Mengurutkan secara descending berdasarkan jumlah suara
    sort.Slice(tabPartai_2311102272, func(i, j int) bool {
        if tabPartai_2311102272[i].Suara ==
tabPartai_2311102272[j].Suara {
```

```
            return tabPartai_2311102272[i].ID <
tabPartai_2311102272[j].ID // Jika suara sama, urutkan berdasarkan ID
partai
        }
        return tabPartai_2311102272[i].Suara >
tabPartai_2311102272[j].Suara
    })
```

```
    // Menampilkan hasil
    fmt.Println("\nHasil perolehan suara:")
    for _, partai := range tabPartai_2311102272 {
        fmt.Printf("(%d:%d) ", partai.ID, partai.Suara)
    }
    fmt.Println()
}
```



```
PS C:\Users\ACER> go run "e:\golang semester 3\modul15_5.go"
Masukkan perolehan suara untuk setiap partai (akhiri dengan -1):
8 8 8 8 8 8 8 8 8 8 8 8 8 8 -1

Hasil perolehan suara:
(8:15)
PS C:\Users\ACER>
```