

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

UJIAN AKHIR



Disusun Oleh :

Haposan Felix Marcel Siregar / 2311102210

S1IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

1. Code Program

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

func identitas() {
    fmt.Println("=====")
    fmt.Println("NIM: 2311102210")
    fmt.Println("Nama:Haposan Felix Marcel Siregar")
    fmt.Println("Kelas: IF-11-06")
    fmt.Println("=====")
}

func main() {
    identitas()
    //Fungsi ini untuk membaca input dari user
    scanner := bufio.NewScanner(os.Stdin)
    fmt.Println("Masukkan elemen baris pertama (pisahkan dengan spasi):")
    scanner.Scan()
    line1 := strings.Fields(scanner.Text())

    fmt.Println("Masukkan elemen baris kedua (pisahkan dengan spasi):")
    scanner.Scan()
    line2 := strings.Fields(scanner.Text())

    // Konversi input ke integer dan buang elemen duplikat
    set1_2311102210 := toUniqueSet(line1)
    set2 := toUniqueSet(line2)

    // Cari irisan dari kedua himpunan
    intersection := findIntersection(set1_2311102210, set2)

    // Tampilkan hasil
    fmt.Println("Irisan kedua himpunan adalah:", intersection)
}

func toUniqueSet(elements []string) map[int]bool {
    set := make(map[int]bool)
    for _, el := range elements {
```

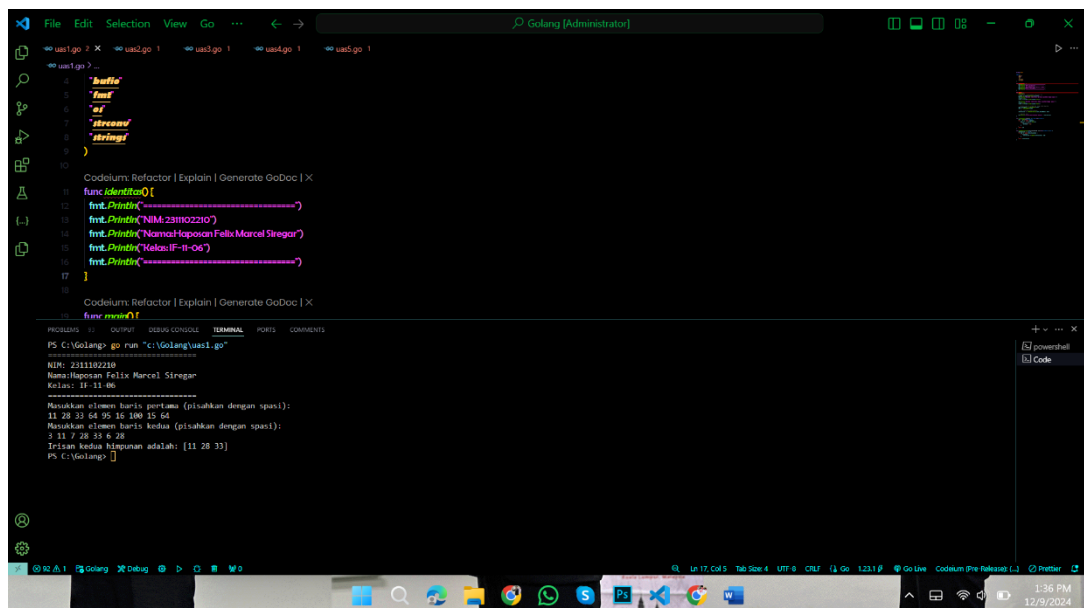
```

        num, err := strconv.Atoi(e1)
        if err == nil {
            set[num] = true
        }
    }
    return set
}

func findIntersection(set1_2311102210, set2 map[int]bool) []int
{
    intersection := []int{}
    for num := range set1_2311102210 {
        if set2[num] {
            intersection = append(intersection, num)
        }
    }
    return intersection
}

```

Output



The screenshot shows a Go IDE with a code editor and a terminal window. The code editor displays a Go program with comments in Indonesian. The terminal window shows the output of the program, which includes the NIM number, name, and class, followed by the intersection of two sets.

```

PS C:\Golang> go run "c:\Golang\uas1.go"
NIM: 2311102210
Nama: Hapson Felix Marcel Siregar
Kelas: IF-11-06

Masukkan elemen baris pertama (pisahkan dengan spasi):
11 28 33 64 99 16 100 15 64
Masukkan elemen baris kedua (pisahkan dengan spasi):
2 11 7 28 33 6 28
Irisan kedua himpunan adalah: [11 28 33]
PS C:\Golang>

```

2. Source Code

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

// Struct Mahasiswa
type Mahasiswa struct {
    NIM_2311102210 string
    Nama string
    Nilai int
}

func identitas() {
    fmt.Println("=====")
    fmt.Println("NIM: 2311102210")
    fmt.Println("Nama:Haposan Felix Marcel Siregar")
    fmt.Println("Kelas: IF-11-06")
    fmt.Println("=====")
}

func main() {
    identitas()
    var mahasiswaList []Mahasiswa
    var n int

    // Input jumlah data mahasiswa
    fmt.Print("Masukkan jumlah data mahasiswa (N): ")
    fmt.Scan(&n)

    // Input data mahasiswa
    scanner := bufio.NewScanner(os.Stdin)
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan data mahasiswa ke-%d (format: NIM_2311102210 Nama Nilai): ", i+1)
        scanner.Scan()
        input := strings.Fields(scanner.Text())

        // Validasi input
        if len(input) != 3 {
            fmt.Println("Error: Format input harus berupa 'NIM_2311102210 Nama Nilai'. Coba lagi.")
            i-- // Ulangi input untuk mahasiswa ini
            continue
        }
    }
}
```

```

        nim := input[0]
        nama := input[1]
        nilai, err := strconv.Atoi(input[2])
        if err != nil {
            fmt.Println("Error: Nilai harus berupa angka. Coba
lagi.")
            i-- // Ulangi input untuk mahasiswa ini
            continue
        }

        mahasiswaList = append(mahasiswaList,
Mahasiswa{NIM_2311102210: nim, Nama: nama, Nilai: nilai})
    }

    // Input NIM_2311102210 yang akan dicari
    fmt.Print("\nMasukkan NIM_2311102210 mahasiswa yang akan
dicari: ")
    scanner.Scan()
    cariNIM := scanner.Text()

    // Pencarian nilai pertama dan nilai terbesar
    nilaiPertama := cariNilaiPertama(mahasiswaList, cariNIM)
    nilaiTerbesar := cariNilaiTerbesar(mahasiswaList, cariNIM)

    // Tampilkan hasil
    if nilaiPertama == -1 {
        fmt.Printf("\nMahasiswa dengan NIM_2311102210 %s tidak
ditemukan.\n", cariNIM)
    } else {
        fmt.Printf("\nNilai pertama mahasiswa dengan
NIM_2311102210 %s: %d\n", cariNIM, nilaiPertama)
        fmt.Printf("Nilai terbesar mahasiswa dengan
NIM_2311102210 %s: %d\n", cariNIM, nilaiTerbesar)
    }
}

// Fungsi mencari nilai pertama berdasarkan NIM_2311102210
func cariNilaiPertama(data []Mahasiswa, nim string) int {
    for _, m := range data {
        if m.NIM_2311102210 == nim {
            return m.Nilai
        }
    }
    return -1 // Jika tidak ditemukan
}

// Fungsi mencari nilai terbesar berdasarkan NIM_2311102210
func cariNilaiTerbesar(data []Mahasiswa, nim string) int {
    maxNilai := -1
    for _, m := range data {

```

Output

```
File Edit Selection View Go ... < -> Golang [Administrator]
--
25 func main() {
26     // Input data mahasiswa
27     scanner := bufio.NewScanner(os.Stdin)
28     for i := 0; i < nProv; i++ {
29         fmt.Printf("Masukkan data mahasiswa ke-%d (format: NIM_231102210 NamaNilai): ", i+1)
30         scanner.Scan()
31         input := strings.Fields(scanner.Text())
32
33         // Validasi input
34         if len(input) != 3 {
35             fmt.Printf("Error: Format input harus berupa 'NIM_231102210 NamaNilai'. Coba lagi.")
36             // Ulangi input untuk mahasiswa ini
37             continue
38         }
39         nim := input[0]
40     }
41 }
42
43 PS C:\Golang> go run "c:\Golang\uas2.go"
44 NIM: 2311102210
45 Nama: Hapusan Felix Marcel Siregar
46 Kelas: IF-11-00
47
48 Masukkan jumlah data mahasiswa (N): 3
49 Masukkan data mahasiswa ke-1 (format: NIM_231102210 NamaNilai): Error: Format input harus berupa 'NIM_231102210 NamaNilai'. Coba lagi.
50 Masukkan data mahasiswa ke-1 (format: NIM_231102210 NamaNilai): 2311102210 Hapusan 100
51 Masukkan data mahasiswa ke-2 (format: NIM_231102210 NamaNilai): 2311102211 Jhon 95
52 Masukkan data mahasiswa ke-3 (format: NIM_231102210 NamaNilai): 2311102212 Dimas 90
53
54 Masukkan NIM_231102210 mahasiswa yang akan dicari: 2311102210
55
56 Nilai pertama mahasiswa dengan NIM_231102210 2311102210: 100
57 Nilai terbesar mahasiswa dengan NIM_231102210 2311102210: 100
58 PS C:\Golang>
```

3. Source Code

```
package main

import (
    "fmt"
    "strings"
)

const nProv = 34

type NamaProv [nProv]string
type PopProv [nProv]int
type TumbuhProv [nProv]float64

// Fungsi untuk input data
func InputData(prov *NamaProv, pop *PopProv, tumbuh *TumbuhProv) {
    for i := 0; i < nProv; i++ {
        fmt.Printf("Masukkan data untuk provinsi %d (format: nama populasi pertumbuhan): ", i+1)
        var nama string
        var populasi int
        var pertumbuhan float64
        fmt.Scan(&nama, &populasi, &pertumbuhan)
        prov[i] = nama
        pop[i] = populasi
        tumbuh[i] = pertumbuhan
    }
}
```

```

// Fungsi untuk mencari provinsi dengan pertumbuhan penduduk
tercepat
func ProvinsiTercepat(tumbuh TumbuhProv) int {
    maxIdx := 0
    for i := 1; i < nProv; i++ {
        if tumbuh[i] > tumbuh[maxIdx] {
            maxIdx = i
        }
    }
    return maxIdx
}

// Fungsi untuk menghitung prediksi jumlah penduduk
func Prediksi(prov NamaProv, pop PopProv, tumbuh TumbuhProv) {
    fmt.Println("Provinsi dengan prediksi populasi tahun depan
(pertumbuhan di atas 2%):")
    for i := 0; i < nProv; i++ {
        if tumbuh[i] > 0.02 {
            prediksiPop := float64(pop[i]) * (1 + tumbuh[i])
            fmt.Printf("%s: %.0f\n", prov[i], prediksiPop)
        }
    }
}

// Fungsi untuk mencari indeks provinsi berdasarkan nama
func IndeksProvinsi(prov NamaProv, nama string) int {
    for i := 0; i < nProv; i++ {
        if strings.EqualFold(prov[i], nama) {
            return i
        }
    }
    return -1
}

func main() {
    var prov_2311102210 NamaProv
    var pop PopProv
    var tumbuh TumbuhProv

    // Input data provinsi
    InputData(&prov_2311102210, &pop, &tumbuh)

    // Menentukan provinsi dengan pertumbuhan tercepat
    tercepatIdx := ProvinsiTercepat(tumbuh)
    fmt.Printf("Provinsi dengan pertumbuhan tercepat: %s\n",
prov_2311102210[tercepatIdx])

    // Input nama provinsi untuk dicari
    var cariNama string
    fmt.Print("Masukkan nama provinsi yang ingin dicari: ")
    fmt.Scan(&cariNama)

    // Mencari indeks provinsi berdasarkan nama
    idxProvinsi := IndeksProvinsi(prov_2311102210, cariNama)
    fmt.Printf("Indeks provinsi %s: %d\n", cariNama, idxProvinsi)
}

```

```
// Menampilkan prediksi populasi provinsi dengan pertumbuhan
di atas 2%
Prediksi(prov_2311102210, pop, tumbuh)
}
```

Output

```
func main() {
    fmt.Println("Indeks provinsi %s: %s", prov_2311102210, cariNama_idxProvinsi())
}

// ... (code for reading data and calculating predictions) ...

// Output:
PS C:\Golang> go run "c:\Golang\ust3.go"
Masukkan data untuk provinsi 1 (format: nama populasi pertumbuhan): JawaTengah 10000000 0.01
Masukkan data untuk provinsi 2 (format: nama populasi pertumbuhan): JawaBarat 7500000 0.2
Masukkan data untuk provinsi 3 (format: nama populasi pertumbuhan): SumateraUtara 2050000 0.50
Masukkan data untuk provinsi 4 (format: nama populasi pertumbuhan): Aceh 3434323243434 0.034
Masukkan data untuk provinsi 5 (format: nama populasi pertumbuhan): SumateraBarat 2300000 0.09
Masukkan data untuk provinsi 6 (format: nama populasi pertumbuhan): SumateraTengah 4500000 0.076
Masukkan data untuk provinsi 7 (format: nama populasi pertumbuhan): Jambi 4576875848574875475 1.2
Masukkan data untuk provinsi 8 (format: nama populasi pertumbuhan): DKI Jakarta 84000000 1.5
Masukkan data untuk provinsi 9 (format: nama populasi pertumbuhan): Riau 4000000 0.001
Masukkan data untuk provinsi 10 (format: nama populasi pertumbuhan): JawaTimur 4454456566 1.9
Masukkan data untuk provinsi 11 (format: nama populasi pertumbuhan): Bali 500000 1.1
Masukkan data untuk provinsi 12 (format: nama populasi pertumbuhan): KalimantanSelatan 67542334 1.7
Masukkan data untuk provinsi 13 (format: nama populasi pertumbuhan): KalimantanTimur 435456566 2.1
Masukkan data untuk provinsi 14 (format: nama populasi pertumbuhan): KalimantanBarat 1234567 8.4
Masukkan data untuk provinsi 15 (format: nama populasi pertumbuhan): KalimantanTengah 4500000 1.6
Masukkan data untuk provinsi 16 (format: nama populasi pertumbuhan): NTB 74378463874574854347 2.2
Masukkan data untuk provinsi 17 (format: nama populasi pertumbuhan): MTI 43407564564786 1.09
Masukkan data untuk provinsi 18 (format: nama populasi pertumbuhan): SulawesiBarat 2345000 0.09
Masukkan data untuk provinsi 19 (format: nama populasi pertumbuhan): SulawesiTengah 43474754564568 0.03
Masukkan data untuk provinsi 20 (format: nama populasi pertumbuhan): SulawesiUtara 37437478478478487 0.88
Masukkan data untuk provinsi 21 (format: nama populasi pertumbuhan): SulawesiSelatan 43474754564568 0.03
Masukkan data untuk provinsi 22 (format: nama populasi pertumbuhan): Makassar 479347384784384 2.1
Masukkan data untuk provinsi 23 (format: nama populasi pertumbuhan): Papua 4337847845346 0.3
Masukkan data untuk provinsi 24 (format: nama populasi pertumbuhan): DITAN 73683784784783 0.45
Masukkan data untuk provinsi 25 (format: nama populasi pertumbuhan): Batuk 8473637847843 0.23
Masukkan data untuk provinsi 26 (format: nama populasi pertumbuhan): PapuaTengah 7845747845343 0.34
Masukkan data untuk provinsi 27 (format: nama populasi pertumbuhan): Negara 47878456 0.45
Masukkan data untuk provinsi 28 (format: nama populasi pertumbuhan): Ayo 7478347554 0.02
Masukkan data untuk provinsi 29 (format: nama populasi pertumbuhan): Thon 482343432 1.78
Masukkan data untuk provinsi 30 (format: nama populasi pertumbuhan): Tenggara 748456578478 1.9
Masukkan data untuk provinsi 31 (format: nama populasi pertumbuhan): Jo 37437463471 2.1
Masukkan data untuk provinsi 32 (format: nama populasi pertumbuhan): Hapusan 73487364874 0.98
Masukkan data untuk provinsi 33 (format: nama populasi pertumbuhan): Abi 4756667876 0.23
Masukkan data untuk provinsi 34 (format: nama populasi pertumbuhan): Marsel 343438743 0.1
Provinsi dengan pertumbuhan tercapat: KalimantanTimur
Masukkan nama provinsi yang ingin dicari: SumateraUtara
Indeks provinsi SumateraUtara: 2
Provinsi dengan prediksi populasi tahun depan (pertumbuhan di atas 2%):
JawaBarat: 9000000
SumateraUtara: 53915148
Aceh: 35099922336171
SumateraBarat: 25070000
SumateraSelatan: 49065000
Jambi: 100477686866728054
DKI Jakarta: 210000000
JawaTimur: 120175940431
Bali: 1200000
KalantanSelatan: 1823643029
KalantanTimur: 1499953554
KalantanBarat: 2925543
KalantanTengah: 117000000
TI: 125678085202102
SulawesiBarat: 25560500
SulawesiUtara: 484324767567128832
SulawesiTengah: 49311029215544
Makassar: 14879768915731598
Papua: 5636862220950
DITAN: 1002916352457935
Batuk: 1042257455238047
PapuaTengah: 105141054695788
Negara: 69423761
Thon: 13489142161
Tenggara: 1776237277586
Jo: 116096136760
Hapusan: 145468982451
Abi: 584001467
Marsel: 377782617
PS C:\Golang>
```

```
func main() {
    fmt.Println("Indeks provinsi %s: %s", prov_2311102210, cariNama_idxProvinsi())
}

// ... (code for reading data and calculating predictions) ...

// Output:
Masukkan data untuk provinsi 30 (format: nama populasi pertumbuhan): Tenggara 748456578478 1.9
Masukkan data untuk provinsi 31 (format: nama populasi pertumbuhan): Jo 37437463471 2.1
Masukkan data untuk provinsi 32 (format: nama populasi pertumbuhan): Hapusan 73487364874 0.98
Masukkan data untuk provinsi 33 (format: nama populasi pertumbuhan): Abi 4756667876 0.23
Masukkan data untuk provinsi 34 (format: nama populasi pertumbuhan): Marsel 343438743 0.1
Provinsi dengan pertumbuhan tercapat: KalimantanTimur
Masukkan nama provinsi yang ingin dicari: SumateraUtara
Indeks provinsi SumateraUtara: 2
Provinsi dengan prediksi populasi tahun depan (pertumbuhan di atas 2%):
JawaBarat: 9000000
SumateraUtara: 53915148
Aceh: 35099922336171
SumateraBarat: 25070000
SumateraSelatan: 49065000
Jambi: 100477686866728054
DKI Jakarta: 210000000
JawaTimur: 120175940431
Bali: 1200000
KalantanSelatan: 1823643029
KalantanTimur: 1499953554
KalantanBarat: 2925543
KalantanTengah: 117000000
TI: 125678085202102
SulawesiBarat: 25560500
SulawesiUtara: 484324767567128832
SulawesiTengah: 49311029215544
Makassar: 14879768915731598
Papua: 5636862220950
DITAN: 1002916352457935
Batuk: 1042257455238047
PapuaTengah: 105141054695788
Negara: 69423761
Thon: 13489142161
Tenggara: 1776237277586
Jo: 116096136760
Hapusan: 145468982451
Abi: 584001467
Marsel: 377782617
PS C:\Golang>
```


4. Source Code

```
package main

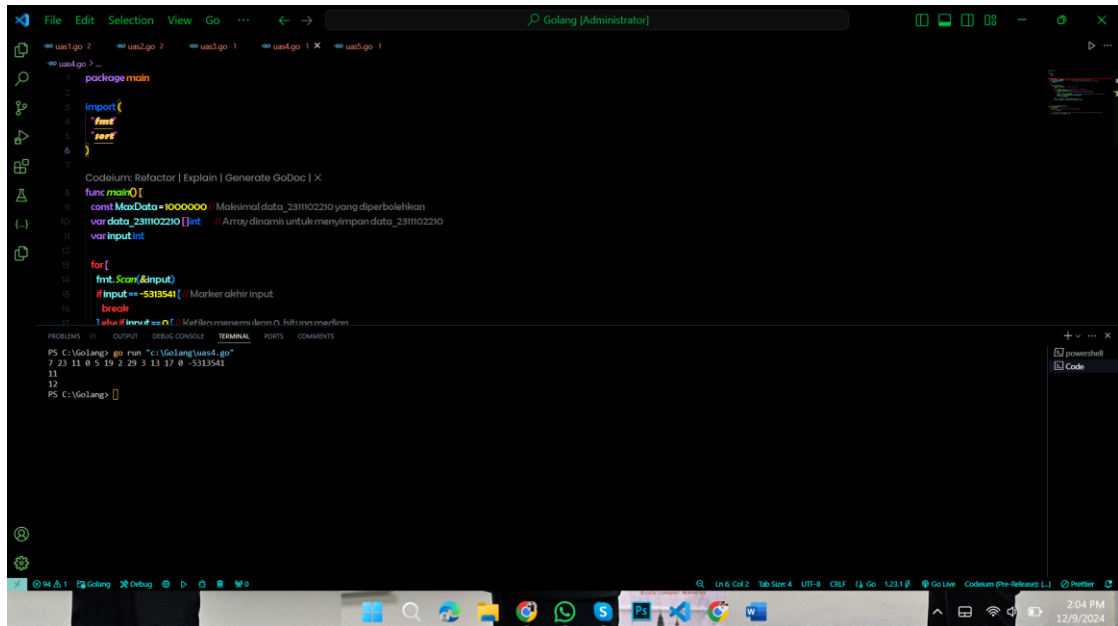
import (
    "fmt"
    "sort"
)

func main() {
    const MaxData = 1000000 // Maksimal data_2311102210 yang
    diperbolehkan
    var data_2311102210 []int // Array dinamis untuk
    menyimpan data_2311102210
    var input int

    for {
        fmt.Scan(&input)
        if input == -5313541 { // Marker akhir input
            break
        } else if input == 0 { // Ketika menemukan 0, hitung
        median
            if len(data_2311102210) == 0 {
                fmt.Println("Tidak ada data_2311102210 untuk
                dihitung median.")
            } else {
                sort.Ints(data_2311102210) // Urutkan
                data_2311102210 menggunakan fungsi bawaan Go
                median := calculateMedian(data_2311102210)
                fmt.Println(median)
            }
        } else { // Tambahkan data_2311102210 ke array
            data_2311102210 = append(data_2311102210, input)
        }
    }
}

// Fungsi untuk menghitung median
func calculateMedian(arr []int) int {
    n := len(arr)
    if n%2 == 1 { // Jika jumlah data_2311102210 ganjil
        return arr[n/2]
    }
    // Jika jumlah data_2311102210 genap, hitung rerata dua
    nilai tengah dan bulatkan ke bawah
    return (arr[n/2-1] + arr[n/2]) / 2
}
```

Output



The screenshot shows a Go IDE with the following source code in `uas4.go`:

```
1 package main
2
3 import {
4     "fmt"
5     "sync"
6 }
7
8 func main() {
9     const MaxData = 1000000 // Maksimal data_231102210 yang diperbolehkan
10    var data_231102210 []int // Array dinamis untuk menyimpan data_231102210
11    var input int
12
13    for {
14        fmt.Scan(&input)
15        if input == -519541 { // Marker akhir input
16            break
17        }
18        data_231102210 = append(data_231102210, input)
19    }
20}
```

The terminal output shows the execution of the program:

```
PS C:\Golang> go run "c:\Golang\uas4.go"
2 23 11 0 5 19 2 29 3 13 17 0 -513541
11
12
PS C:\Golang>
```

5. Source Code

```
package main

import "fmt"

// Konstanta jumlah maksimal partai
const NMAX = 1000000

// Struktur partai
type Partai struct {
    Nama_2311102210 int // Nama_2311102210 partai
    Suara int // Jumlah suara yang diperoleh
}

// Tipe array partai
type tabPartai [NMAX]Partai

func identitas() {
    fmt.Println("=====")
    fmt.Println("NIM: 2311102210")
    fmt.Println("Nama:Haposan Felix Marcel Siregar")
    fmt.Println("Kelas: IF-11-06")
    fmt.Println("=====")
}
```

```

func main() {
    identitas()
    var p tabPartai // Array untuk menyimpan data partai
    var n int        // Jumlah partai unik
    var input int

    // Membaca input suara secara berulang hingga menemukan -1
    for {
        fmt.Scan(&input)
        if input == -1 { // Marker untuk mengakhiri input
            break
        }

        // Mencari posisi partai dalam array
        pos := posisi(p, n, input)
        if pos == -1 { // Jika partai belum ada, tambahkan ke
array
            p[n] = Partai{Nama_2311102210: input, Suara: 1}
            n++
        } else { // Jika partai sudah ada, tambahkan suaranya
            p[pos].Suara++
        }
    }

    // Mengurutkan array p dengan insertion sort secara
descending
    insertionSort(&p, n)

    // Menampilkan hasil
    for i := 0; i < n; i++ {
        fmt.Printf("%d(%d) ", p[i].Nama_2311102210, p[i].Suara)
    }
    fmt.Println()
}

// Fungsi untuk mencari posisi partai dalam array
func posisi(t tabPartai, n int, nama int) int {
    for i := 0; i < n; i++ {
        if t[i].Nama_2311102210 == nama {
            return i // Mengembalikan indeks partai jika
ditemukan
        }
    }
    return -1 // Mengembalikan -1 jika partai tidak ditemukan
}

// Fungsi untuk mengurutkan array partai secara descending
berdasarkan jumlah suara
func insertionSort(t *tabPartai, n int) {
    for i := 1; i < n; i++ {
        temp := t[i]
        j := i - 1

        // Geser elemen yang lebih kecil
        for j >= 0 && (t[j].Suara < temp.Suara || (t[j].Suara
== temp.Suara && t[j].Nama_2311102210 > temp.Nama_2311102210))

```

```

{
    t[j+1] = t[j]
    j--
}

t[j+1] = temp
}
}

```

Output

```

File Edit Selection View Go ... < -> Golang (Administrator)
uns1.go 2 uns2.go 2 uns3.go 1 uns4.go 1 uns5.go 2 X
12 }
13
14 // Tipe array portal
15 type TabuPortal [NMAX]Portal
16
17 Codeium: Refactor | Explain | Generate GoDoc | X
18 func identitas() {
19     fmt.Println("=====")
20     fmt.Println("NIM: 231102210")
21     fmt.Println("Nama: Hapusan Felix Marcel Siregar")
22     fmt.Println("Kelas: IF-11-06")
23     fmt.Println("=====")
24 }
25
26 Codeium: Refactor | Explain | Generate GoDoc | X
27 func main() {
28     identitas()
29     var p TabuPortal // Array untuk menyimpan data portal
30 }
31
32 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
33 PS C:\Golang> go run "C:\Golang\uns5.go"
34
35 NIM: 231102210
36 Nama: Hapusan Felix Marcel Siregar
37 Kelas: IF-11-06
38
39 =====
40 5 1 1 1 1 1 1 3 3 3 2 2 2 5 5 5 5 4 3 2 2 2 -1
41 1(7) 2(6) 5(6) 3(5) 4(1)
42 PS C:\Golang>

```