LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2

MODUL 11

PENCARIAN NILAI ACAK PADA HIMPUNAN DATA

Oleh:

HANAH NUR AZIZAH

2311102312

IF-11-07

PROGRAM STUDY S1 INFORMATIKA

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM PURWOKERTO

2024

I. DASAR TEORI

11.1 Sequential Search

Pencarian secara sekuensial ini adalah pencarian yang dilakukan dari data pertama, kedua hingga terakhir secara satu persatu dan berurutan. Ciri khas dari pencarian ini adalah proses pencarian akan berhenti ketika data yang dicari ditemukan, walaupun masih terdapat data yang belum dicek nilainya. Algoritma ini dikenal dengan nama **Sequential Search**, karena prosesnya melakukan pengecekan setiap elemen array secara satu persatu dan sekuensial dari data pertama hingga ditemukan atau data terakhir.

1) Asumsi terdapat suatu array of integer T dengan indeks dari 0 hingga N-1, dan suatu nilai yang dicari pada array T, yaitu X.

2) Status pencarian digunakan untuk menandakan data yang dicari ditemukan atau tidak, misalnya found dengan tipe boolean.

3) Pencarian dilakukan dari T[0] sampai ke T[N-1], setiap kali perbandingan dengan X, update nilai found.

4) Perulangan harus dihentikan apabila status pencarian found bernilai true (data ditemukan) atau T[N-1] telah dicek.

11.2 Binary Search

Ide algoritma adalah: (dengan asumsi data terurut dari kecil membesar (ascending), dan data dengan indeks kecil ada di "kiri" dan indeks besar ada di "kanan")

1) Ambil salah satu data dalam rentang data yang ada, algoritma di bawah menggunakan rentang dari ***kr*** s.d. kanan ***kn***. Untuk kemudahan dan alasan lainnya, biasanya yang diambil adalah paling tengah dalam rentang tersebut.

2) Jika data terambil tersebut terlalu kecil, maka ubah/geser rentang data ke sebelah kanan posisi data tersebut. Ini karena jika data terambil terlalu kecil, maka semua data sebalah kirinya juga akan terlalu kecil dari yang ingin dicari.

3) Begitu juga sebaliknya jika data terambil terlalu besar.

11.3 Pencarian pada Array Bertipe Data Struct

Algoritma pencarian pada array bertipe data struct tidak berbeda jauh dengan tipe data dasar, kita hanya perlu menambahkan field kategori dari pencarian yang akan dilakukan. Khusus untuk algoritma binary search, maka keterurutan array harus sama dengan field kategori dari pencariannya. Misalnya apabila pencarian berdasarkan nim mahasiswa, maka algoritma binary search hanya akan bisa digunakan apabila array terurut berdasarkan nim. Algoritma binary search tidak akan berjalan apabila array terurut berdasarkan nim mahasiswa, tetapi pencarian yang dilakukan berdasarkan filed nama atau kategori yang lain selain nim. Solusinya adalah array harus diurutkan berdasarkan nim terlebih dahulu, atau gunakan algoritma sequential search.

II. GUIDED

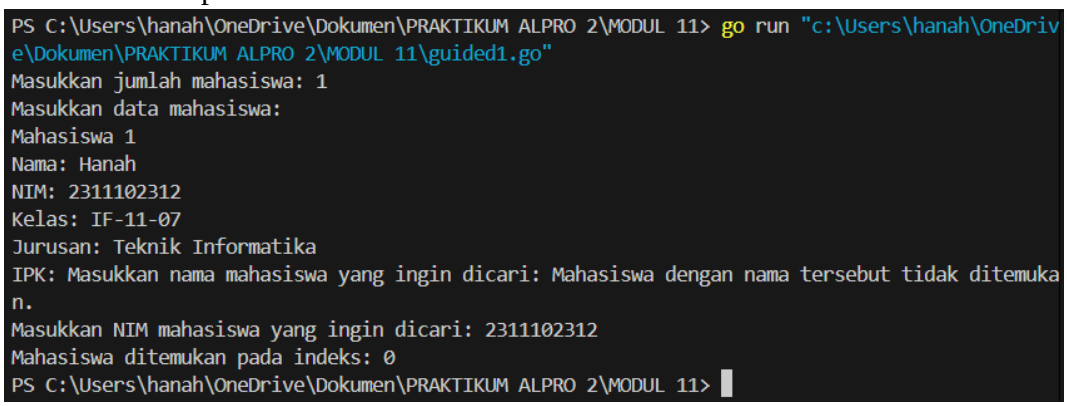
1. Source Code

```go
package main
import (
    "fmt"
)
//Hanah Nur Azizah 2311102312
// Definisi struct mahasiswa
type mahasiswa struct {
    nama    string
    nim     string
    kelas   string
    jurusan string
    ipk     float64
}

// Definisi array bertipe struct mahasiswa
type arrMhs [2023]mahasiswa

// Sequential Search berdasarkan nama
func SeqSearch_3(T arrMhs, n int, X string) int {
    /* Mengembalikan indeks mahasiswa dengan nama X,
       atau -1 apabila tidak ditemukan pada array T
       yang berisi n data mahasiswa */
    var found int = -1
    var j int = 0
    for j < n && found == -1 {
        if T[j].nama == X {
            found = j
        }
        j = j + 1
    }
    return found
}

// Binary Search berdasarkan nim
func BinarySearch_3(T arrMhs, n int, X string) int {
    /* Mengembalikan indeks mahasiswa dengan nim X,
       atau -1 apabila tidak ditemukan pada array T
       yang berisi n data mahasiswa dan terurut membesar berdasarkan
nim */
    var found int = -1
    var med int
    var kr int = 0
```

```
    var kn int = n - 1

    for kr <= kn && found == -1 {
        med = (kr + kn) / 2
        if X < T[med].nim {
            kn = med - 1
        } else if X > T[med].nim {
            kr = med + 1
        } else {
            found = med
        }
    }
    return found
}

// Fungsi utama untuk demonstrasi
func main() {
    var data arrMhs
    var n int

    // Input jumlah mahasiswa
    fmt.Print("Masukkan jumlah mahasiswa: ")
    fmt.Scanln(&n)

    // Input data mahasiswa
    fmt.Println("Masukkan data mahasiswa:")
    for i := 0; i < n; i++ {
        fmt.Printf("Mahasiswa %d\n", i+1)
        fmt.Print("Nama: ")
        fmt.Scanln(&data[i].nama)
        fmt.Print("NIM: ")
        fmt.Scanln(&data[i].nim)
        fmt.Print("Kelas: ")
        fmt.Scanln(&data[i].kelas)
        fmt.Print("Jurusan: ")
        fmt.Scanln(&data[i].jurusan)
        fmt.Print("IPK: ")
        fmt.Scanln(&data[i].ipk)
    }

    // Pencarian berdasarkan nama
    var cariNama string
    fmt.Print("Masukkan nama mahasiswa yang ingin dicari: ")
    fmt.Scanln(&cariNama)
    posNama := SeqSearch_3(data, n, cariNama)
    if posNama == -1 {
        fmt.Println("Mahasiswa dengan nama tersebut tidak
ditemukan.")
    } else {
        fmt.Printf("Mahasiswa ditemukan pada indeks: %d\n", posNama)
```

```
    }

    // Pencarian berdasarkan nim
    var cariNIM string
    fmt.Print("Masukkan NIM mahasiswa yang ingin dicari: ")
    fmt.Scanln(&cariNIM)
    posNIM := BinarySearch_3(data, n, cariNIM)
    if posNIM == -1 {
        fmt.Println("Mahasiswa dengan NIM tersebut tidak
ditemukan.")
    } else {
        fmt.Printf("Mahasiswa ditemukan pada indeks: %d\n", posNIM)
    }
}
```

Screenshot Output



Penjelasan

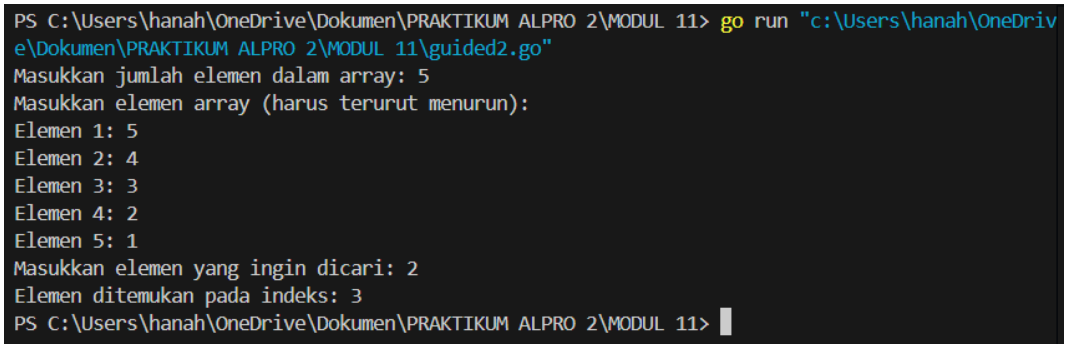
Program di atas adalah program Go yang mendemonstrasikan pencarian data mahasiswa menggunakan dua metode pencarian, yaitu Sequential Search berdasarkan nama dan Binary Search berdasarkan NIM. Program dimulai dengan meminta pengguna memasukkan jumlah mahasiswa, lalu data mahasiswa (nama, NIM, kelas, jurusan, dan IPK) disimpan dalam array bertipe mahasiswa. Pencarian nama dilakukan dengan Sequential Search, yang memeriksa elemen satu per satu hingga menemukan nama yang cocok atau mencapai akhir array. Pencarian NIM dilakukan dengan Binary Search, yang lebih efisien tetapi membutuhkan data yang sudah terurut berdasarkan NIM. Program mengembalikan indeks data mahasiswa yang ditemukan, atau pesan jika data tidak ditemukan.

2. Source Code

```go
package main

import "fmt"
//Hanah Nur Azizah 2311102312
// Define arrInt as an array of integers with a fixed size
type arrInt [4321]int
// Binary search function for descending sorted array
func BinarySearch_2(T arrInt, n int, X int) int {
    /* Mengembalikan indeks dari X apabila X ditemukan di dalam
array T
       yang berisi n buah bilangan bulat terurut secara
descending/menciut,
       atau -1 apabila X tidak ditemukan */
    var found int = -1
    var med int
    var kr int = 0
    var kn int = n - 1

    // Iterasi mencari elemen dengan binary search
    for kr <= kn && found == -1 {
        med = (kr + kn) / 2
        if X > T[med] { // Karena descending, jika X lebih besar,
bergerak ke kiri
            kn = med - 1
        } else if X < T[med] { // Jika X lebih kecil, bergerak ke
kanan
            kr = med + 1
        } else { // Jika ditemukan
            found = med
        }
    }
    return found
}

func main() {
    // Contoh penggunaan
    var data arrInt
    var n int
    // Input jumlah elemen
    fmt.Print("Masukkan jumlah elemen dalam array: ")
    fmt.Scanln(&n)
    // Input elemen array (harus terurut secara descending)
    fmt.Println("Masukkan elemen array (harus terurut menurun):")
```

```go
    for i := 0; i < n; i++ {
        fmt.Printf("Elemen %d: ", i+1)
        fmt.Scanln(&data[i])
    }

    // Input elemen yang dicari
    var search int
    fmt.Print("Masukkan elemen yang ingin dicari: ")
    fmt.Scanln(&search)

    // Panggil fungsi binary search
    result := BinarySearch_2(data, n, search)

    // Cetak hasil
    if result == -1 {
        fmt.Println("Elemen tidak ditemukan dalam array.")
    } else {
        fmt.Printf("Elemen ditemukan pada indeks: %d\n", result)
    }
}
```

Screenshot Output



Penjelasan

Program di atas adalah implementasi pencarian elemen dalam array bilangan bulat yang terurut secara menurun (descending) menggunakan algoritma Binary Search dalam bahasa Go. Program dimulai dengan meminta pengguna memasukkan jumlah elemen dalam array dan elemen-elemennya, dengan catatan bahwa array harus diinput dalam urutan menurun. Selanjutnya, pengguna diminta memasukkan nilai yang ingin dicari dalam array. Algoritma Binary Search digunakan untuk mencari nilai tersebut dengan membagi array secara berulang hingga ditemukan elemen yang cocok atau memastikan elemen tidak ada dalam array. Program mengembalikan indeks elemen jika ditemukan, atau pesan bahwa elemen tidak ditemukan.

3. Source Code

```go
package main
import "fmt"
//Hanah Nur Azizah 2311102312
// Define arrStr as an array of strings with a fixed size
type arrStr [1234]string
// Sequential search function
func SeqSearch_1(T arrStr, n int, X string) int {
    /* Mengembalikan indeks dari X apabila X ditemukan di dalam
array T
       yang berisi n buah teks, atau -1 apabila X tidak ditemukan */
    var found int = -1
    var j int = 0

    // Iterasi mencari elemen yang cocok
    for j < n && found == -1 {
        if T[j] == X {
            found = j
        }
        j = j + 1
    }
    return found
}
func main() {
    // Contoh penggunaan
    var data arrStr
    var n int

    // Input jumlah elemen
    fmt.Print("Masukkan jumlah elemen dalam array: ")
    fmt.Scanln(&n)

    // Input elemen array
    fmt.Println("Masukkan elemen array:")
    for i := 0; i < n; i++ {
        fmt.Printf("Elemen %d: ", i+1)
        fmt.Scanln(&data[i])
    }
    // Input elemen yang dicari
    var search string
    fmt.Print("Masukkan elemen yang ingin dicari: ")
    fmt.Scanln(&search)
    // Panggil fungsi sequential search
    result := SeqSearch_1(data, n, search)
    // Cetak hasil
```

```
    if result == -1 {
        fmt.Println("Elemen tidak ditemukan dalam array.")
    } else {
        fmt.Printf("Elemen ditemukan pada indeks: %d\n", result)
    }
}
```

Screenshot Output

```
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 11> go run "c:\Users\hanah\OneDriv
e\Dokumen\PRAKTIKUM ALPRO 2\MODUL 11\guided3.go"
Masukkan jumlah elemen dalam array: 4
Masukkan elemen array:
Elemen 1: 1
Elemen 2: 2
Elemen 3: 3
Elemen 4: 4
Masukkan elemen yang ingin dicari: 2
Elemen ditemukan pada indeks: 1
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 11>
```

Penjelasan

Program di atas merupakan implementasi pencarian teks dalam array string menggunakan algoritma Sequential Search dalam bahasa Go. Pengguna diminta untuk memasukkan jumlah elemen dalam array dan elemen-elemennya berupa string. Setelah itu, pengguna memasukkan teks yang ingin dicari. Algoritma Sequential Search akan memeriksa setiap elemen array secara berurutan hingga menemukan teks yang cocok atau mencapai akhir array. Jika teks ditemukan, program mengembalikan indeksnya; jika tidak, program mencetak pesan bahwa elemen tidak ditemukan.

III. UNGUIDED

1. Pada pemilihan ketua RT yang baru saja berlangsung, terdapat 20 calon ketua yang bertanding memperebutkan suara warga. Perhitungan suara dapat segera dilakukan karena warga cukup mengisi formulir dengan nomor dari calon ketua RT yang dipilihnya. Seperti biasa, selalu ada pengisian yang tidak tepat atau dengan nomor pilihan di luar yang tersedia, sehingga data juga harus divalidasi. Tugas Anda untuk membuat program mencari siapa yang memenangkan pemilihan ketua RT.

   Buatlah program **pilkart** yang akan membaca, memvalidasi, dan menghitung suara yang diberikan dalam pemilihan ketua RT tersebut.
   Masukan hanya satu baris data saja, berisi bilangan bulat valid yang kadang tersisipi dengan data tidak valid. Data valid adalah integer dengan nilai di antara 1 s.d. 20 (inklusif). Data berakhir jika ditemukan sebuah bilangan dengan nilai 0.
   Keluaran dimulai dengan baris berisi jumlah data suara yang terbaca, diikuti baris yang berisi berapa banyak suara yang valid. Kemudian sejumlah baris yang mencetak data para calon apa saja yang mendapatkan suara.

No	Masukan	Keluaran
1	7 19 3 2 78 3 1 -3 18 19 0	Suara masuk: 10 Suara sah: 8 1: 1 2: 1 3: 2 7: 1 18: 1 19: 2

Source Code

```
package main
import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)
//Hanah Nur Azizah 2311102312
func main() {
    reader := bufio.NewReader(os.Stdin)
    fmt.Println("Masukkan data suara (pisahkan dengan spasi):")
    input, _ := reader.ReadString('\n')
```

```
    input = strings.TrimSpace(input)
    votes := strings.Split(input, " ")
    var validVotes []int
    invalidCount := 0

    for _, voteStr := range votes {
        vote, err := strconv.Atoi(voteStr)
        if err != nil || vote < 0 {
            invalidCount++
            continue
        }
        if vote == 0 {
            break
        }
        if vote >= 1 && vote <= 20 {
            validVotes = append(validVotes, vote)
        } else {
            invalidCount++
        }
    }

    voteCount := make([]int, 20)
    for _, vote := range validVotes {
        voteCount[vote-1]++
    }

    fmt.Printf("Suara masuk: %d\n", len(votes))
    fmt.Printf("Suara Sah: %d\n", len(validVotes))
    for i, count := range voteCount {
        if count > 0 {
            fmt.Printf("%d: %d\n", i+1, count)
        }
    }

}
```

Screenshot Output

```
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 11> go run "c:\Users\hanah\OneDriv
e\Dokumen\PRAKTIKUM ALPRO 2\MODUL 11\unguided1.go"
Masukkan data suara (pisahkan dengan spasi):
7 19 3 2 78 3 1 -3 18 19 0
Suara masuk: 11
Suara Sah: 8
1: 1
2: 1
3: 2
7: 1
18: 1
19: 2
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 11>
```

Penjelasan
Program di atas adalah program Go untuk menghitung dan menganalisis hasil pemilihan ketua RT berdasarkan input suara. Program dimulai dengan meminta pengguna memasukkan data suara dalam satu baris, di mana setiap suara dipisahkan dengan spasi. Data suara ini kemudian divalidasi: hanya angka antara 1 hingga 20 yang dianggap sebagai suara sah, sementara angka di luar rentang tersebut atau input non-angka dihitung sebagai suara tidak sah. Angka 0 digunakan sebagai penanda akhir input, sehingga suara setelahnya diabaikan. Suara sah dihitung dan diakumulasi untuk masing-masing kandidat, disimpan dalam array voteCount. Program kemudian mencetak jumlah total suara yang masuk, jumlah suara sah, serta distribusi suara yang diterima oleh masing-masing kandidat. Program ini membantu menampilkan informasi penting tentang hasil pemilihan secara terstruktur.

2. Berdasarkan program sebelumnya, buat program **pilkart** yang mencari siapa pemenang pemilihan ketua RT. Sekaligus juga ditentukan bahwa wakil ketua RT adalah calon yang mendapatkan suara terbanyak kedua. Jika beberapa calon mendapatkan suara terbanyak yang sama, ketua terpilih adalah dengan nomor peserta yang paling kecil dan wakilnya dengan nomor peserta terkecil berikutnya.

   Masukan hanya satu baris data saja, berisi bilangan bulat valid yang kadang tersisipi dengan data tidak valid. Data valid adalah bilangan bulat dengan nilai di antara 1 s.d. 20 (inklusif). Data berakhir jika ditemukan sebuah bilangan dengan nilai 0.

   Keluaran dimulai dengan baris berisi jumlah data suara yang terbaca, diikuti baris yang berisi berapa banyak suara yang valid. Kemudian tercetak calon nomor berapa saja yang menjadi pasangan ketua RT dan wakil ketua RT yang baru.

No	Masukan	Keluaran
1	7 19 3 2 78 3 1 -3 18 19 0	Suara masuk: 10 Suara sah: 8 Ketua RT: 3 Wakil ketua: 19

Source Code

```go
package main
import (
    "bufio"
    "fmt"
    "os"
    "sort"
    "strconv"
    "strings"
)

// Hanah Nur Azizah 2311102312
func main() {
    reader := bufio.NewReader(os.Stdin)
    fmt.Println("Masukan:")
    input, _ := reader.ReadString('\n')
    input = strings.TrimSpace(input)
    votes := strings.Split(input, " ")
    var validVotes []int
    invalidCount := 0

    for _, voteStr := range votes {
        vote, err := strconv.Atoi(voteStr)
        if err != nil || vote < 0 {
            invalidCount++
            continue
        }
        if vote == 0 {
            break
        }
        if vote >= 1 && vote <= 20 {
            validVotes = append(validVotes, vote)
        } else {
            invalidCount++
        }
    }

    voteCount := make([]int, 20)
    for _, vote := range validVotes {
        voteCount[vote-1]++
    }

    fmt.Printf("Suara masuk: %d\n", len(votes))
    fmt.Printf("Suara Sah: %d\n", len(validVotes))
```

```
    type candidate struct {
        number int
        votes  int
    }
    var candidates []candidate
    for i, count := range voteCount {
        if count > 0 {
            candidates = append(candidates, candidate{number: i + 1,
votes: count})
        }
    }

    sort.Slice(candidates, func(i, j int) bool {
        if candidates[i].votes == candidates[j].votes {
            return candidates[i].number < candidates[j].number
        }
        return candidates[i].votes > candidates[j].votes
    })

    if len(candidates) == 0 {
        fmt.Println("Tidak ada suara valid. Tidak ada ketua atau
wakil terpilih.")
    } else if len(candidates) == 1 {
        fmt.Printf("Ketua RT: %d\n", candidates[0].number)
        fmt.Println("Wakil Ketua: Tidak ada wakil terpilih karena
hanya satu kandidat dengan suara.")
    } else {
        fmt.Printf("Ketua RT: %d\n", candidates[0].number)
        fmt.Printf("Wakil Ketua: %d\n", candidates[1].number)
    }
}
```

Screenshot Output

```
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 11> go run "c:\Users\hanah\OneDriv
e\Dokumen\PRAKTIKUM ALPRO 2\MODUL 11\unguided2.go"
Masukan:
7 19 3 2 78 3 1 -3 18 19
Suara masuk: 10
Suara Sah: 8
Ketua RT: 3
Wakil Ketua: 19
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 11>
```

Penjelasan

Program di atas adalah program GO yang bertujuan untuk menghitung dan menentukan pemenang dalam pemilihan ketua RT berdasarkan data suara yang dimasukkan oleh pengguna. Program ini dimulai

dengan membaca input suara dalam bentuk angka yang dipisahkan oleh spasi. Suara yang valid adalah angka antara 1 hingga 20, dan data yang invalid atau tidak sesuai akan diabaikan. Jika ditemukan angka 0, program akan berhenti memproses suara. Setelah itu, program menghitung jumlah suara untuk setiap calon ketua RT, mengurutkan calon berdasarkan jumlah suara terbanyak, dan menampilkan ketua serta wakil ketua berdasarkan urutan suara. Jika hanya ada satu calon yang valid, maka hanya ketua yang diumumkan tanpa wakil ketua. Program ini juga mencetak jumlah suara yang masuk dan jumlah suara yang sah.

3. Diberikan n data integer positif dalam keadaan terurut membesar dan sebuah integer lain k, apakah bilangan k tersebut ada dalam daftar bilangan yang diberikan? Jika ya, berikan indeksnya, jika tidak sebutkan "TIDAK ADA".

   Masukan terdiri dari dua baris. Baris pertama berisi dua buah integer positif, yaitu n dan k. n menyatakan banyaknya data, dimana 1 < n <= 1000000. k adalah bilangan yang ingin dicari. Baris kedua berisi n buah data integer positif yang sudah terurut membesar.

   Keluaran terdiri dari satu baris saja, yaitu sebuah bilangan yang menyatakan posisi data yang dicari (k) dalam kumpulan data yang diberikan. Posisi data dihitung dimulai dari angka 0. Atau memberikan keluaran "TIDAK ADA" jika data k tersebut tidak ditemukan dalam kumpulan.

   Program yang dibangun harus menggunakan subprogram dengan mengikuti kerangka yang sudah diberikan berikut ini.

```
package main
import "fmt"

const NMAX = 1000000
var data [NMAX]int

func main(){
/* buatlah kode utama yang membaca baris pertama (n dan k). kemudian data
diisi
   oleh prosedur isiArray(n), dan pencarian oleh fungsi posisi(n,k), dan
setelah
   itu output dicetak. */
}
```

```
func isiArray(n int){
/* I.S. terdefinisi integer n, dan sejumlah n data sudah siap pada piranti
masukan.
   F.S. Array data berisi n (<=NMAX) bilangan */
}

func posisi(n, k int) int {
/* mengembalikan posisi k dalam array data dengan n elemen. Posisi dimulai
dari
   posisi 0. Jika tidak ada kembalikan -1 */
}
```

Contoh masukan dan keluaran:

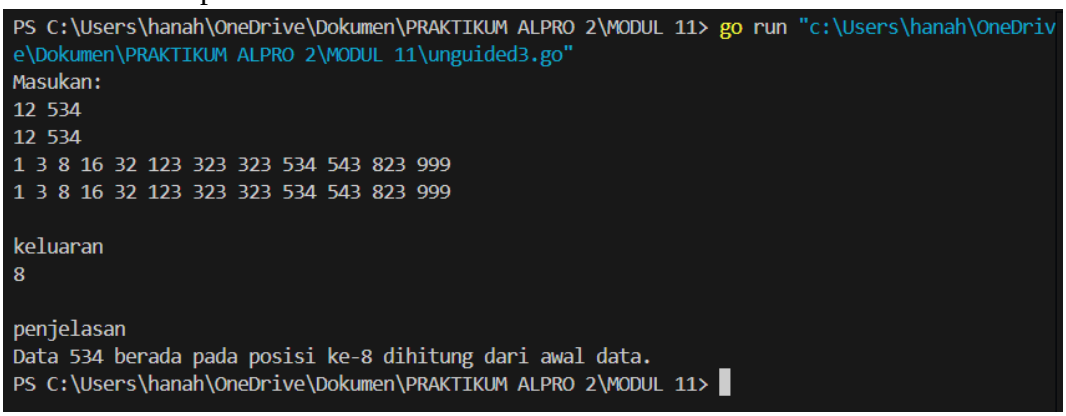
No	Masukan	Keluaran	Penjelasan
1	12 534 1 3 8 16 32 123 323 323 534 543 823 999	8	Data 534 berada pada posisi ke-8 dihitung dari awal data.
2	12 535 1 3 8 16 32 123 323 323 534 543 823 999	TIDAK ADA	

Source Code

```go
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)
// Hanah Nur Azizah 2311102312
const NMAX = 1000000

var data [NMAX]int

func main() {
    reader := bufio.NewReader(os.Stdin)

    fmt.Println("Masukan:")
    line1, _ := reader.ReadString('\n')
    line1 = strings.TrimSpace(line1)
    fmt.Println(line1)
    nk := strings.Split(line1, " ")
    n, _ := strconv.Atoi(nk[0])
    k, _ := strconv.Atoi(nk[1])

    line2, _ := reader.ReadString('\n')
    line2 = strings.TrimSpace(line2)
    fmt.Println(line2)
```

```
    numbers := strings.Split(line2, " ")
    for i := 0; i < n; i++ {
        data[i], _ = strconv.Atoi(numbers[i])
    }

    pos := posisi(n, k)

    fmt.Println("\nkeluaran")
    if pos == -1 {
        fmt.Println("TIDAK ADA")
    } else {
        fmt.Println(pos)
    }

    fmt.Println("\npenjelasan")
    if pos == -1 {
        fmt.Printf("Data %d tidak ditemukan dalam kumpulan data.\n",
k)
    } else {
        fmt.Printf("Data %d berada pada posisi ke-%d dihitung dari
awal data.\n", k, pos)
    }
}

func posisi(n, k int) int {
    low := 0
    high := n - 1

    for low <= high {
        mid := (low + high) / 2
        if data[mid] == k {
            return mid
        } else if data[mid] < k {
            low = mid + 1
        } else {
            high = mid - 1
        }
    }

    return -1
}
```

Screenshot Output



Penjelasan

Program di atas adalah implementasi pencarian bilangan dalam array yang sudah terurut menggunakan metode binary search. Program membaca input dari pengguna yang terdiri dari dua baris: baris pertama berisi jumlah elemen array n dan bilangan yang akan dicari k, sementara baris kedua berisi n bilangan yang terurut membesar. Array diisi melalui iterasi, dan pencarian indeks bilangan k dilakukan menggunakan fungsi posisi dengan algoritma binary search untuk efisiensi. Jika bilangan ditemukan, program mencetak indeksnya; jika tidak, program mencetak "TIDAK ADA". Program juga memberikan penjelasan hasil pencarian, apakah bilangan ditemukan beserta posisinya atau tidak ditemukan sama sekali.

DAFTAR PUSTAKA

[1] Modul 11: Pencarian Nilai Acak Pada Himpunan Data