LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK

MODUL 11

PENCARIAN NILAI ACAK PADA HIMPUNAN DATA

Oleh:

Alden Audy Akbar

231102309

IF-11-07

S1 TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

I. DASAR TEORI

Pencarian data merupakan salah satu operasi dasar dalam algoritma dan pemrograman. Modul ini membahas dua algoritma pencarian utama, yaitu Sequential Search dan Binary Search, serta penerapannya pada berbagai tipe data seperti array biasa dan array bertipe struct.

1. Pengenalan

Pencarian nilai dalam himpunan data merupakan proses penting dalam pengolahan data, yang bertujuan untuk menemukan elemen tertentu dalam struktur data seperti array, list, atau lainnya. Terdapat beberapa algoritma yang digunakan tergantung pada kebutuhan dan sifat data yang digunakan.

2. Algoritma Sequential Search

Sequential search adalah metode pencarian linear yang memeriksa setiap elemen secara berurutan hingga elemen yang dicari ditemukan atau seluruh elemen telah diperiksa. Algoritma ini sederhana dan cocok untuk himpunan data kecil atau tidak terurut.

```
found <- false
i <- 0
while i < n and not found do
   if T[i] == X then
      found <- true
   endif
   i <- i + 1
endwhile
```

3. Algoritma Binary Search

Binary search lebih efisien dibandingkan sequential search, namun memerlukan data yang terurut. Algoritma ini bekerja dengan membagi rentang data menjadi dua bagian, dan hanya memeriksa bagian yang relevan berdasarkan nilai tengah.

```
kr <- 0
kn <- n - 1
found <- false
while kr <= kn and not found do
   med <- (kr + kn) / 2
   if T[med] == X then
      found <- true
   else if T[med] < X then
      kr <- med + 1
   else
      kn <- med - 1
   endif
```

```
endwhile
```

4. Pencarian pada Tipe Data Kompleks

Pada data bertipe struct, pencarian dapat dilakukan berdasarkan field tertentu. Untuk binary search, penting memastikan data terurut berdasarkan field pencarian tersebut.

```
func SeqSearch_3(T arrMhs, n int, X string) int {
    for i := 0; i < n; i++ {
        if T[i].nama == X {
            return i
        }
    }
    return -1
}
```

II. GUIDED

1.

```
package main
import (
	"fmt"
)
// 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)
        }
}
```

```
PS C:\Users\alden> go run "C:\Users\alden\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan jumlah mahasiswa: 3
Masukkan data mahasiswa:
Mahasiswa 1
Nama: Arif
NIM: 2311102300
Kelas: 07
Jurusan: Informatika
IPK: 1.5
Mahasiswa 2
Nama: Arief
NIM: 2311102301
Kelas: 07
Jurusan: Informatika
IPK: 3.0
Mahasiswa 3
Nama: Ariefa
NIM: 2311102302
Kelas: 07
Jurusan: Informatika
IPK: 2.5
Masukkan nama mahasiswa yang ingin dicari: Arif
Mahasiswa ditemukan pada indeks: 0
Masukkan NIM mahasiswa yang ingin dicari: 2311102300
Mahasiswa ditemukan pada indeks: 0
PS C:\Users\alden>
```

Penjelasan: Program meminta pengguna untuk memasukkan data mahasiswa secara manual.Data mahasiswa yang dimasukkan disimpan ke dalam array arrMhs. Pengguna memilih jenis pencarian. Jika memilih pencarian berdasarkan nama, maka fungsi SeqSearch_3 akan dipanggil.Jika memilih pencarian berdasarkan NIM, maka fungsi BinarySearch_3 akan dipanggil. Hasil pencarian (indeks mahasiswa yang ditemukan atau pesan "tidak ditemukan") ditampilkan ke layar.

2.

```
package main

import "fmt"
// 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):")
   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)
    }
}
```

```
PS C:\Users\alden> go run "C:\Users\alden\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan jumlah elemen dalam array: 3
Masukkan elemen array (harus terurut menurun):
Elemen 1: 12
Elemen 2: 16
Elemen 3: 20
Masukkan elemen yang ingin dicari: 16
Elemen ditemukan pada indeks: 1
PS C:\Users\alden>
```

Penjelasan: Pengguna diminta untuk memasukkan jumlah elemen yang ingin diproses.Pengguna kemudian diminta untuk memasukkan nilai-nilai elemen secara berurutan (harus terurut menurun).Pengguna diminta untuk memasukkan nilai yang ingin dicari dalam array.Fungsi BinarySearch_2 dipanggil dengan memberikan array, jumlah elemen, dan nilai yang dicari sebagai parameter.Jika nilai ditemukan, program akan menampilkan indeks di mana nilai tersebut berada.Jika nilai tidak ditemukan, program akan menampilkan pesan "Elemen tidak ditemukan dalam array".

3.

```
package main
import "fmt"
// 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)
    }
}
```

```
PS C:\Users\alden> go run "C:\Users\alden\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan jumlah elemen dalam array: 5
Masukkan elemen array:
Elemen 1: 12
Elemen 2: 14
Elemen 3: 16
Elemen 4: 18
Elemen 5: 20
Masukkan elemen yang ingin dicari: 20
Elemen ditemukan pada indeks: 4
PS C:\Users\alden>
```

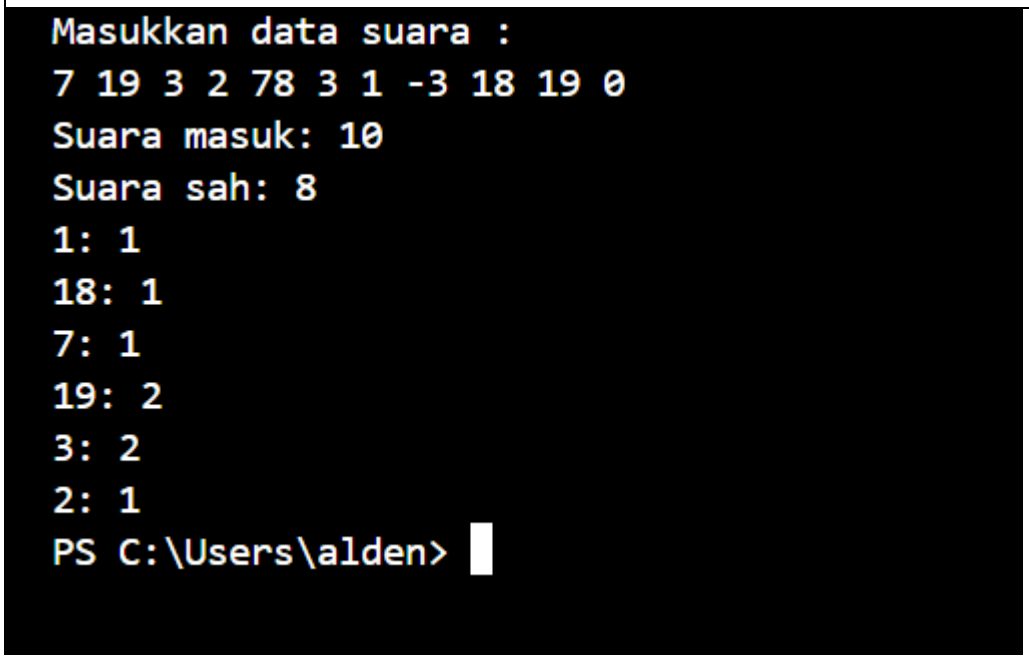
Penjelasan: Pengguna diminta memasukkan jumlah kata yang ingin diproses.Pengguna kemudian diminta memasukkan kata-kata tersebut satu per satu.Pengguna diminta memasukkan kata yang ingin dicari dalam array.Fungsi SeqSearch_1 dipanggil dengan memberikan array, jumlah elemen, dan kata yang dicari sebagai parameter.Jika kata ditemukan, program akan menampilkan indeks di mana kata tersebut berada. Jika kata tidak ditemukan, program akan menampilkan pesan "Elemen tidak ditemukan dalam array".

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.

```
package main
//2311102309
import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)
func main() {
    reader := bufio.NewReader(os.Stdin)
    fmt.Println("Masukkan data suara :")
    input, _ := reader.ReadString('\n')
    input = strings.TrimSpace(input)

    data := strings.Split(input, " ")

    var totalSuara int
    var suaraValid []int
    hitungSuara := make(map[int]int)

    for _, item := range data {
        num, err := strconv.Atoi(item)
        if err != nil {
            continue
        }

        if num == 0 {
            break
        }

        totalSuara++
```

```
if num >= 1 && num <= 20 {
            suaraValid = append(suaraValid, num)
            hitungSuara[num]++
        }
    }

    fmt.Printf("Suara masuk: %d\n", totalSuara)
    fmt.Printf("Suara sah: %d\n", len(suaraValid))

    for calon, jumlah := range hitungSuara {
        fmt.Printf("%d: %d\n", calon, jumlah)
    }
}
```



Penjelasan: Program meminta pengguna untuk memasukkan data suara.Input yang dimasukkan akan disimpan dalam variabel input.Input yang berupa string dipecah menjadi potongan-potongan kata (string) berdasarkan spasi menggunakan fungsi strings.Split. Setiap potongan kata (yang seharusnya berupa angka) disimpan dalam slice data.Setiap elemen dalam slice data diubah menjadi tipe data integer menggunakan fungsi strconv.Atoi. Jika terjadi kesalahan konversi (misalnya, ada karakter selain angka), program akan melanjutkan ke elemen berikutnya.Setiap angka yang berhasil dikonversi akan diperiksa apakah berada dalam rentang 1 hingga 20. Jika valid, angka tersebut akan ditambahkan ke slice suaraValid dan dihitung jumlah kemunculannya dalam map hitungSuara. Program menghitung

total suara yang masuk dan total suara yang valid. Program menampilkan total suara yang masuk, total suara yang valid, dan jumlah suara untuk setiap calon.

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.

```
package main
//2311102309
import (
    "bufio"
    "fmt"
    "os"
    "sort"
    "strconv"
    "strings"
)
func main() {
    reader := bufio.NewReader(os.Stdin)
    fmt.Println("Masukkan data suara :")
    input, _ := reader.ReadString('\n')
    input = strings.TrimSpace(input)
data := strings.Split(input, " ")
var totalSuara int
    var suaraValid []int
    hitungSuara := make(map[int]int)
for _, item := range data {
        num, err := strconv.Atoi(item)
        if err != nil {
            continue
        }

        if num == 0 {
            break
        }
        totalSuara++
       if num >= 1 && num <= 20 {
            suaraValid = append(suaraValid, num)
            hitungSuara[num]++
}
```

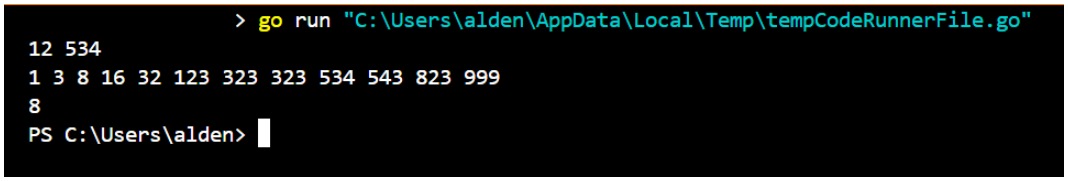
```
    }
type calon struct {
        nomor int
        suara int
    }
    var daftarCalon []calon
    for nomor, suara := range hitungSuara {
        daftarCalon = append(daftarCalon, calon{nomor, suara})
    }
    sort.Slice(daftarCalon, func(i, j int) bool {
        if daftarCalon[i].suara == daftarCalon[j].suara {
            return daftarCalon[i].nomor < daftarCalon[j].nomor
        }
        return daftarCalon[i].suara > daftarCalon[j].suara
    })
ketua := daftarCalon[0].nomor
    wakil := daftarCalon[1].nomor
fmt.Printf("Suara masuk: %d\n", totalSuara)
    fmt.Printf("Suara sah: %d\n", len(suaraValid))
fmt.Printf("Ketua RT: %d\n", ketua)
    fmt.Printf("Wakil ketua: %d\n", wakil)
}
```

```
PS C:\Users\alden> go run "C:\Users\alden\AppData\Local\Temp\tempCodeRunnerFile.go"
Masukkan data suara :
7 19 3 2 78 3 1 -3 18 19 0
Suara masuk: 10
Suara sah: 8
Ketua RT: 3
Wakil ketua: 19
PS C:\Users\alden>
```

Penjelasan: Masukan berupa satu baris data angka dipisahkan oleh spasi.Suara valid dihitung dengan syarat berada dalam rentang 1-20.Angka 0 sebagai tanda akhir masukan.Data suara dihitung dan diurutkan berdasarkan jumlah suara (menurun).Jika ada suara terbanyak sama, nomor peserta terkecil diutamakan.Ketua adalah calon dengan suara terbanyak, dan wakil adalah berikutnya. Total suara masuk, suara sah, serta nomor calon Ketua RT dan Wakil Ketua RT.

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".

```
package main
//2311102309
import "fmt"

const NMAX = 1000000
var data [NMAX]int

func main() {
    var n, k int
fmt.Scan(&n, &k)
isiArray(n)

   index := posisi(n, k)
    if index == -1 {
        fmt.Println("TIDAK ADA")
    } else {
        fmt.Println(index)
    }
}
func isiArray(n int) {
    for i := 0; i < n; i++ {
        fmt.Scan(&data[i])
    }
}
func posisi(n, k int) int {
    left, right := 0, n-1
    for left <= right {
        mid := (left + right) / 2
        if data[mid] == k {
            return mid
        } else if data[mid] < k {
            left = mid + 1
        } else {
            right = mid - 1
        }
    }
    return -1
}
```



Penjelasan: Program membaca input berupa dua angka, yaitu n (jumlah elemen array) dan k (elemen yang dicari). Selanjutnya, n bilangan akan dimasukkan sebagai elemen array.Program mengisi array data dengan n bilangan yang diinputkan pengguna.Program menggunakan pencarian biner untuk mencari elemen k dalam array data yang diasumsikan terurut: Jika elemen k ditemukan, program mengembalikan indeks posisi elemen tersebut. Jika elemen k tidak ditemukan, program mengembalikan nilai -1. Jika elemen ditemukan, program mencetak indeksnya. Jika elemen tidak ditemukan, program mencetak "TIDAK ADA".