

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 8

PENCARIAN NILAI ACAK PADA HIMPUNAN DATA



Oleh:

ARIEL AHNAF KUSUMA

103112400050

12-IF-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. DASAR TEORI

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.

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")

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

GUIDED 1

SOURCE CODE

```
package main

import (
    "fmt"
    "sort"
)

func sequentialSearch(arr []float64, target float64) (int, int) {
    iterations := 0
    for i, val := range arr {
        iterations++
        fmt.Printf("Sequential Step %d: cek arr[%d] = %.1f\n", iterations, i,
val)
        if val == target {
            return i, iterations
        }
    }
    return -1, iterations
}

func binarySearch(arr []float64, target float64) (int, int) {
    iterations := 0
    low := 0
    high := len(arr) - 1

    for low <= high {
        iterations++
        mid := (low + high) / 2
        fmt.Printf("Binary Step %d cek arr[%d] = %.1f\n", iterations, mid,
arr[mid])

        if arr[mid] == target {
            return mid, iterations
        } else if target < arr[mid] {
            high = mid - 1
        } else {
            low = mid + 1
        }
    }
    return -1, iterations
}
```

```

func main() {
    data := []float64{2, 7, 9, 1, 5, 6, 18, 13, 25, 20}
    target := 13.0

    fmt.Println("Squential Search (data tidak perluurut)")
    idxSeq, iterSeq := sequentialSearch(data, target)
    if idxSeq != 1 {
        fmt.Printf("Hasil: Ditemukan di indeks %d dalam %d langkah\n\n",
idxSeq, iterSeq)
    } else {
        fmt.Printf("Hasil: Tidak ditemukan setelah %d langkah\n\n",
iterSeq)
    }

    sort.Float64s(data)
    fmt.Println("Binary Search (setelah data diurutkan):")
    fmt.Println("Data terurut:", data)

    idxBin, iterBin := binarySearch(data, target)
    if idxBin != -1 {
        fmt.Printf("Hasil: Ditemukan di indeks %d dalam %d langkah\n",
idxBin, iterBin)
    } else {
        fmt.Printf("Hasil: Tidak ditemukan setelah %d langkah\n", iterBin)
    }
}

```

OUTPUT CODE

```

PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 8> go run "c:\ARIEL AHNAF KUSUMA\103112400050_Modul 8\seqsearch1.go"
Squential Search (data tidak perluurut)
Squential Step 1: cek arr[0] = 2.0
Squential Step 2: cek arr[1] = 7.0
Squential Step 3: cek arr[2] = 9.0
Squential Step 4: cek arr[3] = 1.0
Squential Step 5: cek arr[4] = 5.0
Squential Step 6: cek arr[5] = 6.0
Squential Step 7: cek arr[6] = 18.0
Squential Step 8: cek arr[7] = 13.0
Hasil: Ditemukan di indeks 7 dalam 8 langkah

Binary Search (setelah data diurutkan):
Data terurut: [1 2 5 6 7 9 13 18 20 25]
Binary Step 1 cek arr[4] = 7.0
Binary Step 2 cek arr[7] = 18.0
Binary Step 3 cek arr[5] = 9.0
Binary Step 4 cek arr[6] = 13.0
Hasil: Ditemukan di indeks 6 dalam 4 langkah
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 8>

```

DESKRIPSI CODE

Program code di atas berfungsi untuk menghitung total gaji karyawan menggunakan algoritma sequential search dan binary search

GUIDED 2

SOURCE CODE

```
package main

import (
    "fmt"
    "sort"
)

type mahasiswa struct {
    nama, nim, kelas, jurusan string
    ipk                        float64
}

type arrMhs [2023]mahasiswa

// Sequential Search berdasarkan nama
func SeqSearch_3(T arrMhs, n int, X string) (int, int) {
    var found int = -1
    var j int = 0
    var iterasi int = 0

    for j < n && found == -1 {
        iterasi++
        if T[j].nama == X {
            found = j
        }
        j++
    }
    return found, iterasi
}

// Binary Search berdasarkan NIM (data harus sudah terurut
// berdasarkan nim)
func BinarySearch_3(T arrMhs, n int, X string) (int, int) {
    var found int = -1
    var med int
    var kr int = 0
    var kn int = n - 1
```

```

var iterasi int = 0

for kr <= kn && found == -1 {
    iterasi++
    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, iterasi
}

func main() {
    var data arrMhs
    n := 10

    // Mengisi data secara manual
    data = arrMhs{
        {nama: "Ari", nim: "2201", kelas: "A", jurusan: "Informatika", ipk:
3.4},
        {nama: "Budi", nim: "2203", kelas: "A", jurusan: "Informatika",
ipk: 3.6},
        {nama: "Cici", nim: "2202", kelas: "B", jurusan: "Sistem
Informasi", ipk: 3.5},
        {nama: "Dina", nim: "2205", kelas: "A", jurusan: "Informatika",
ipk: 3.3},
        {nama: "Eko", nim: "2204", kelas: "B", jurusan: "Sistem
Informasi", ipk: 3.7},
        {nama: "Fajar", nim: "2206", kelas: "C", jurusan: "Informatika",
ipk: 3.1},
        {nama: "Gita", nim: "2209", kelas: "C", jurusan: "Informatika",
ipk: 3.8},
        {nama: "Hana", nim: "2208", kelas: "B", jurusan: "Sistem
Informasi", ipk: 3.2},
        {nama: "Iwan", nim: "2207", kelas: "C", jurusan: "Informatika",
ipk: 3.0},
        {nama: "Joko", nim: "2210", kelas: "A", jurusan: "Informatika",
ipk: 3.9},
    }

    // Pencarian Sequential Search berdasarkan nama
    namaDicari := "Fajar"

```

```

idxSeq, iterSeq := SeqSearch_3(data, n, namaDicari)

fmt.Printf("Sequential Search - Cari nama '%s'\n", namaDicari)
if idxSeq != -1 {
    fmt.Printf("Ditemukan di indeks: %d, Iterasi: %d\n", idxSeq,
iterSeq)
} else {
    fmt.Printf("Tidak ditemukan, Iterasi: %d\n", iterSeq)
}

// Urutkan data berdasarkan NIM untuk binary search
sort.Slice(data[:n], func(i, j int) bool {
    return data[i].nim < data[j].nim
}))

// Pencarian Binary Search berdasarkan NIM
nimDicari := "2206"
idxBin, iterBin := BinarySearch_3(data, n, nimDicari)

fmt.Printf("\nBinary Search - Cari NIM '%s'\n", nimDicari)
if idxBin != -1 {
    fmt.Printf("Ditemukan di indeks: %d, Iterasi: %d\n", idxBin,
iterBin)
} else {
    fmt.Printf("Tidak ditemukan, Iterasi: %d\n", iterBin)
}
}

```

OUTPUT CODE

```

PS C:\VARIABLE AHNAF KUSUMA\103112400050_Modul 8> go run "c:\VARIABLE AHNAF KUSUMA\103112400050_Modul 8\tempCodeRunnerFile.go"
Sequential Search - Cari nama 'Fajar'
Ditemukan di indeks: 5, Iterasi: 6

Binary Search - Cari NIM '2206'
Ditemukan di indeks: 5, Iterasi: 3
PS C:\VARIABLE AHNAF KUSUMA\103112400050_Modul 8>

```

DESKRIPSI CODE

Program diatas berfungsi untuk mengelola data data mahasiswa lalu mengelolanya

III. UNGUIDED

UNGUIDED 1

SOURCE CODE

```
1  package main
2
3  //103112400050
4
5  import "fmt"
6
7  func main() {
8      var nomorPilihan int
9      suaraMasuk := 0
10     suaraSah := 0
11     hitungSuaraCalon := make([]int, 21)
12     for {
13         _, err := fmt.Scan(&nomorPilihan)
14         if err != nil || nomorPilihan == 0 {
15             break
16         }
17         suaraMasuk++
18         if nomorPilihan >= 1 && nomorPilihan <= 20 {
19             suaraSah++
20             hitungSuaraCalon[nomorPilihan]++
21         }
22     }
23     fmt.Printf("Suara masuk: %d\n", suaraMasuk)
24     fmt.Printf("Suara sah: %d\n", suaraSah)
25     for i := 1; i <= 20; i++ {
26         if hitungSuaraCalon[i] > 0 {
27             fmt.Printf("%d: %d\n", i, hitungSuaraCalon[i])
28         }
29     }
30 }
31
```

OUTPUT CODE


```
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 8> go run "c
EL AHNAF KUSUMA\103112400050_Modul 8\unguided1.go"
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
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 8> █
```

DESKRIPSI CODE

Program di atas berfungsi untuk menghitung suara pemilihan ketua rt, suara masuk, suara sah diikuti dengan diikuti berapa banyak suara yang valid.

UNGUIDED 2

SOURCE CODE

```

1  package main
2
3  //103112400050
4  import (
5      "fmt"
6  )
7
8  func main() {
9      const MaxCalon = 20
10     var suara [MaxCalon + 1]int
11     var input, total, sah int
12
13     for {
14         fmt.Scan(&input)
15         if input == 0 {
16             break
17         }
18         total++
19         if input >= 1 && input <= MaxCalon {
20             suara[input]++
21             sah++
22         }
23     }
24
25     fmt.Printf("Suara masuk: %d\n", total)
26     fmt.Printf("Suara sah: %d\n", sah)
27
28     var ketua, wakil int
29     for i := 1; i <= MaxCalon; i++ {
30         if suara[i] > suara[ketua] || (suara[i] == suara[ketua] && i < ketua) {
31             wakil = ketua
32             ketua = i
33         } else if (suara[i] > suara[wakil] || (suara[i] == suara[wakil] && i < wakil)) && i != ketua {
34             wakil = i
35         }
36     }
37
38     fmt.Printf("Ketua RT: %d\n", ketua)
39     fmt.Printf("Wakil ketua: %d\n", wakil)
40 }
41

```

OUTPUT CODE

```

PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 8> go run "c:\ARIEL AHNAF KUSUMA\103112400050_Modul 8\umguided2.go"
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:\ARIEL AHNAF KUSUMA\103112400050_Modul 8>

```

DESKRIPSI CODE

Program diatas berfungsi untuk menghitung total suara dan siapa yang memenangkan pemilihan ketua rt dan wakil ketua rt

UNGGUIDED 3

SOURCE CODE

```

1  package main
2
3  //103112400050
4  import "fmt"
5
6  const NMAX = 1000000
7
8  var data [NMAX]int
9
10 func main() {
11     var n, k int
12     fmt.Scan(&n, &k)
13     isiArray(n)
14
15     pos := posisi(n, k)
16     if pos == -1 {
17         fmt.Println("TIDAK ADA")
18     } else {
19         fmt.Println(pos)
20     }
21 }
22
23 func isiArray(n int) {
24     for i := 0; i < n; i++ {
25         fmt.Scan(&data[i])
26     }
27 }
28
29 func posisi(n, k int) int {
30     kr := 0
31     kn := n - 1
32
33     for kr <= kn {
34         mid := (kr + kn) / 2
35         if data[mid] == k {
36             return mid
37         } else if data[mid] < k {
38             kr = mid + 1
39         } else {
40             kn = mid - 1
41         }
42     }
43     return -1
44 }

```

OUTPUT CODE

```
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 8> go run "c:\ARIEL AHNAF KUSUMA\103112400050_Modul 8\unguided3.go"
12 534
1 3 8 16 32 123 323 323 534 543 823 999
8
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 8> go run "c:\ARIEL AHNAF KUSUMA\103112400050_Modul 8\unguided3.go"
12 535
1 3 8 16 32 123 323 323 534 543 823 999
TIDAK ADA
PS C:\ARIEL AHNAF KUSUMA\103112400050_Modul 8> █
```

DESKRIPSI CODE

Program diatas berfungsi untuk mencari posisi dari suatu bilangan k dalam sekumpulan data yang telah terurut membesar, program tersebut menggunakan algoritma binary search

IV. KESIMPULAN

V. REFERENSI

**ALGORITMA PEMROGRAMMAN 2 MODUL 8 |
TELKOM UNIVERSITY**