

1.go > ...

```
1  package main
2  ✓ type v23 struct {
3      v2, v3 int
4  }
5  ✓ type Trec struct {
6      v1 int
7      vx v23
8      v4 int
9  }
10 func main() {}
11 func banyakNilai(A *Trec) {}
12 type tabInt [9999]int
13 ✓ func minimum(tab tabInt, n int) int {
14     var min int
15     var k int
16     var idx int
17     min = tab[0]
18     k = 1
19     ✓ for k < n {
20     ✓         if min > tab[k] {
21             min = tab[k]
22             idx = k
23         }
24         k++
25     }
26     return idx
27 }
28 type arr [9999]int
29 ✓ func total(tab arr, n int) int {
30     var sum int
31     ✓ for i := 0; i < n; i++ {
32         sum = sum + tab[i]
33     }
34     return sum
35 }
36 ✓ func mean(tab arr, n int) int {
37     return total(tab, n) / n
```

```
}  
type arr [9999]int  
func total(tab arr, n int) int {  
    var sum int  
    for i := 0; i < n; i++ {  
        sum = sum + tab[i]  
    }  
    return sum  
}  
func mean(tab arr, n int) int {  
    return total(tab, n) / n  
}  
func maximum(tab tabInt, n int) int  
{  
    var max int  
    var k int  
    var idx int  
    max = tab[0]  
    k = 1  
    for k < n {  
        if max < tab[k] {  
            max = tab[k]  
            idx = k  
        }  
        k++  
    }  
    return idx  
}
```

2.go > tambahData

```
1  package main
2
3  import "fmt"
4
5  type tabInt [100000]int
6
7  func tambahData(arr *tabInt, n *int) {
8      var input int
9      i := 0
10     for true {
11         fmt.Scan(&input)
12         if input == 9999 {
13             break
14         }
15         arr[i] = input
16         i++
17     }
18     *n = i
19 }
20
```

3.go > CariSekuensial


```
1  package main
2
3  type tabInt [100000]int
4
5  func CariSekuensial(t tabInt, n int) int {
6      i := 0
7      for i := 0; i < len(t); i++ {
8          if t[i] == n {
9              break
10         }
11         i++
12     }
13     if i == len(t) {
14         return -1
15     } else {
16         return i
17     }
18 }
19
```

4.go > NilaiMinimum


```
1  package main
2
3  type tabInt [9999]int
4
5  func NilaiMinimum(tab tabInt, n int) int {
6      var min int
7      var k int
8      var idx int
9      min = tab[0]
10     k = 1
11     for k < n {
12         if min > tab[k] {
13             min = tab[k]
14             idx = k
15         }
16         k++
17     }
18     return idx
19 }
20
21
```

5.go > ...

```
1  package main
2
3  type arr [9999]int
4
5  func NilaiRerata(tab arr, n int) int {
6      var sum int
7      for i := 0; i < n; i++ {
8          sum = sum + tab[i]
9      }
10     return sum / n
11 }
12
```

6.go >  terurutA

```
1  package main
2
3  type arr [10000]int
4
5  func terurutA(A *arr, n int) {
6      var pass, idx, i, temp int
7      pass = 0
8      for pass < n-1 {
9          idx = pass
10         i = idx + 1
11         for i < n {
12             if (*A)[i] < (*A)[idx] {
13                 idx = i
14             }
15             i++
16         }
17         temp = (*A)[pass]
18         (*A)[pass] = (*A)[idx]
19         (*A)[idx] = temp
20         pass = pass + 1
21     }
22 }
23
```

GO 7.go >  terurutB

```
1  package main
2
3  type arr [10000]int
4
5  func terurutB(a *arr, n int) {
6      var pass, i, temp int
7      pass = 1
8      for pass < n {
9          i = pass
10         temp = a[pass]
11         for i > 0 && temp > a[i-1] {
12             a[i] = a[i-1]
13             i--
14         }
15         a[i] = temp
16         pass++
17     }
18
19 }
20
```


8.go > CariCepat

```
1  package main
2
3  type arr [10000]int
4
5  func CariCepat(tab arr, n, x int) bool {
6      var left, right, mid int
7      left = 1
8      right = n
9      mid = (left + right) / 2
10     for left <= right && tab[mid] != x {
11         if x < tab[mid] {
12             right = mid - 1
13         } else {
14             left = mid + 1
15         }
16         mid = (left + right) / 2
17     }
18     return mid > 0 && tab[mid] == x
19 }
20
```

No.:

Tugas Pendahuluan

Date:

10.9 |

Prosedur Snaggy akan melibatkan pengalokasian dan alokasi array array parameter. Untuk menentukan variabel array yang memiliki atribut $V_1 = V_x$, $V_2 = V_y$, $V_3 = V_z$, $\text{found} = \text{true}$.

If true, diprint kelima di dalam string