

## KETENTUAN TUGAS PENDAHULUAN

- Tugas pendahuluan bertujuan untuk membantu mahasiswa dalam mempersiapkan diri sebelum mengikuti praktikum.
- Tugas pendahuluan bersifat tugas **INDIVIDUAL** dan **WAJIB** dikerjakan sebelum praktikum. Mahasiswa yang tidak mengerjakan akan memperoleh **PENALTI** pengurangan nilai praktikum sebesar 25%.
- Kerjakan soal dan susun jawaban dengan rapi dan jelas.
- Soal teori boleh cari dari internet atau modul dengan **MELAMPIRKAN SUMBER**. Jika sumber dari modul sertakan halaman, jika sumber dari internet sertakan link sumbernya (tidak boleh dari blogspot atau wordpress, harus dari web yang valid seperti codepolitan, w3schools dan lain-lain).
- Jawaban soal teori/non-koding **WAJIB DITULIS TANGAN**, lalu foto/scan jawaban tersebut dan **SCREENSHOT KODE PROGRAM** dan **OUTPUT/HASIL RUNNING PROGRAM** lalu masukan ke dalam file PDF dan **BERIKAN NOMOR DAN KETERANGAN CLASS SESUAI NOMOR JAWABAN DI PDFNYA** dengan format file **TP\_ALPRO\_MOD\_KELAS\_NIM.pdf**
- Jadikan **SEMUA JAWABAN PROGRAM ATAU FILE KODE PROGRAM DALAM 1 FILE ZIP** dengan format nama/ file **TP\_ALPRO\_MOD\_KELAS\_NIM.zip** **WAJIB DALAM ZIP**
- **SELURUH JAWABAN BAIK DI (SCREENSHOOT) PDF MAUPUN DI (PROGRAM/PROJECT) ZIP/RAR HARUS SAMA DAN SESUAI.**
- Upload file PDF dan ZIP ke google form yang dibuka pada waktu yang telah ditentukan
- Google Form akan dibuka sampai hari **KAMIS, 19 MEI 2022, Jam 07.30 PAGI**
- **DILARANG KERAS MELAKUKAN SEGALA TINDAKAN PLAGIARISME**

- (Login dengan SSO, usernameigracias@student.telkomuniversity.ac.id)
- Video tutorial ALPRO dapat diakses di <https://tiny.cc/VideoTutorialALPRO>.

**NB : JIKA MELANGGAR KETENTUAN DI ATAS**

**NILAI TP = 0**

**SEMANGATTT !!!**

## SOAL TUGAS PENDAHULUAN

1. Tuliskan apa saja algoritma pencarian data yang dipelajari pada mata kuliah algoritma dan pemrograman ini! Beri penjelasan serta perbedaannya!
2. Apabila diberikan sebuah array of string dengan jumlah elemen 999, maka buatlah function pencarian secara sekuensialnya terhadap suatu elemen tertentu dengan mengembalikan indeks hasil pencarian!
3. Tuliskan function pencarian secara binari untuk data terurut secara ascending dan descending. Jelaskan di mana letak perbedaannya.
4. Sebuah program yang digunakan untuk mencari sebuah irisan himpunan.

**Masukan** terdiri dari dua baris, di mana setiap barisnya yang berisi sekumpulan bilangan. Masukan disetiap barisnya akan berakhir apabila bilangan yang diberikan sudah pernah diberikan pada baris tersebut (atau duplikat). Catatan: anggota suatu himpunan tidak boleh duplikat.

**Keluaran** adalah sekumpulan bilangan yang menyatakan irisan dari himpunan pada baris pertama dan baris kedua pada masukan.

### Contoh masukan dan keluaran

<b>Masukan</b>	11 28 33 64 95 16 100 15 64 3 11 7 28 33 6 28
<b>Keluaran</b>	11 28 33

Penjelasan: 64 dan 28 duplikat sehingga masukan berhenti dan tidak perlu dimasukkan ke dalam array.

Buatlah program dengan melengkapi potongan berikut ini!

```
package main
import "fmt"

type set [2022]int

func exist(T set, n int, val int) bool {
    /* mengembalikan true apabila bilangan val ada di dalam
    array T yang berisi sejumlah n bilangan bulat */
    ...
}

func inputSet(T *set, n *int){
    /* I.S. data himpunan telah siap pada piranti masukan
    F.S. array T berisi sejumlah n bilangan bulat yang berasal
    dari masukan (masukan berakhir apabila bilangan ada yang
    duplikat, atau array penuh) Catatan: Panggil function exist
    di sini untuk membantu pengecekan */ ...
}

func findIntersection(T1,T2 set, n,m int, T3 *set, h *int){
    /* I.S. terdefinisi himpunan T1 dan T2 yang berisi sejumlah n
    dan m anggota himpunan
    F.S. himpunan T3 berisi sejumlah h bilangan bulat yang
    merupakan irisan dari himpunan T1 dan T2
    Catatan: Panggil function exist di sini untuk
    membantu pengecekan */ ...
}

func printSet(T set, n int){
    /* I.S. terdefinisi sebuah himpunan T yang berisi
    sejumlah n bilangan bulat F.S. menampilkan isi array T
    secara horizontal (dipisahkan oleh spasi) */ ...
}

func main(){
    var s1,s2,s3 set
    var n1,n2,n3 int
    inputSet(&s1,&n1)
    inputSet(&s2,&n2)
    findIntersection(s1,s2,n1,n2,&s3,&n3)
    printSet(s3,n3)
```

}