

MODUL 1. RUNNING MODUL

1.1 Peraturan Praktikum Laboratorium Informatika

- 1) Praktikum diampu oleh dosen kelas dan dibantu oleh asisten laboratorium dan asisten praktikum.
- 2) Praktikum dilaksanakan di Gedung Telkom University Landmark Tower (TULT) Lantai 6 dan Lantai 7 sesuai jadwal yang ditentukan.
- 3) Praktikan wajib membawa modul praktikum, kartu praktikum, dan alat tulis.
- 4) Praktikan wajib mengecek kehadiran di igracias dan sheet yang dibagikan asisten.
- 5) Durasi kegiatan praktikum S-1 = 2 jam (100 menit).
- 6) Jumlah pertemuan praktikum:
 - 16 kali pertemuan
- 7) Praktikan wajib hadir minimal 75% dari seluruh pertemuan praktikum di lab.
- 8) Praktikan yang datang terlambat :
 - a. ≤ 5 menit : diperbolehkan mengikuti praktikum tanpa tambahan waktu praktikum.
 - b. ≤ 30 menit : tidak diperbolehkan mengikuti praktikum.
- 9) Saat praktikum berlangsung, asisten praktikum dan praktikan:
 - a. Wajib menggunakan seragam sesuai aturan institusi.
 - b. Wajib mematikan/ mengkondisikan semua alat komunikasi.
 - c. Dilarang membuka aplikasi yang tidak berhubungan dengan praktikum yang berlangsung.
 - d. Dilarang mengubah pengaturan *software* maupun *hardware* komputer tanpa izin.
 - e. Dilarang membawa makanan maupun minuman di ruang praktikum.
 - f. Dilarang memberikan jawaban ke praktikan lain.
 - g. Dilarang menyebarkan soal praktikum.
 - h. Dilarang membuang sampah di ruangan praktikum.
 - i. Wajib meletakkan alas kaki dengan rapi pada tempat yang telah disediakan.

10) Setiap praktikan dapat mengikuti praktikum susulan maksimal dua modul untuk satu mata kuliah praktikum.

- a. Praktikan yang dapat mengikuti praktikum susulan hanyalah praktikan yang memenuhi syarat sesuai ketentuan institusi, yaitu: sakit (dibuktikan dengan surat keterangan medis), tugas dari institusi (dibuktikan dengan surat dinas atau dispensasi dari institusi), atau mendapat musibah atau keduakaan (menunjukkan surat keterangan dari orangtua/wali mahasiswa.)
- b. Persyaratan untuk praktikum susulan diserahkan sesegera mungkin kepada asisten laboratorium untuk keperluan administrasi.
- c. Praktikan yang diizinkan menjadi peserta praktikum susulan ditetapkan oleh Asman Lab dan Bengkel Informatika dan tidak dapat diganggu gugat.

11) a. Ketidakhadiran pada kelas praktikum

- Nilai modul = 0

b. Meminta, mendapatkan, dan menyebarkan soal dan atau kunci jawaban praktikum:

- Penyebar soal dan kunci jawaban: Pengajuan sanksi kepada Komisi Disiplin Fakultas
- Penerima soal dan kunci jawaban: Nilai '0' pada (seluruh asesmen) praktikum

c. Lupa menghapus file praktikum

- Pengurangan nilai modul 20%

d. Memicu kegaduhan, sehingga membuat situasi tidak kondusif (jalan-jalan, mengganggu teman, mengobrol, dll), asisten praktikum diwajibkan menegur sebanyak 3x

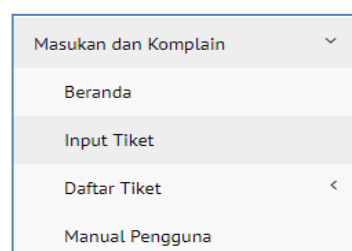
- Pengurangan nilai modul 50%

e. MENYALAHGUNAKAN FITUR LMS

- Siap menerima sanksi dari lab

1.2 Penyampaian Keluhan Praktikum IFLAB Melalui iGracias

- 1) Login iGracias
- 2) Pilih menu **Masukan dan Komplain**, pilih **Input Tiket**
- 3) Pilih Fakultas/Bagian: **Bidang Akademik (FIF)**
- 4) Pilih Program Studi/Urusan: **Urusan Laboratorium/Bengkel/Studio (FIF)**
- 5) Pilih Layanan: **Praktikum**

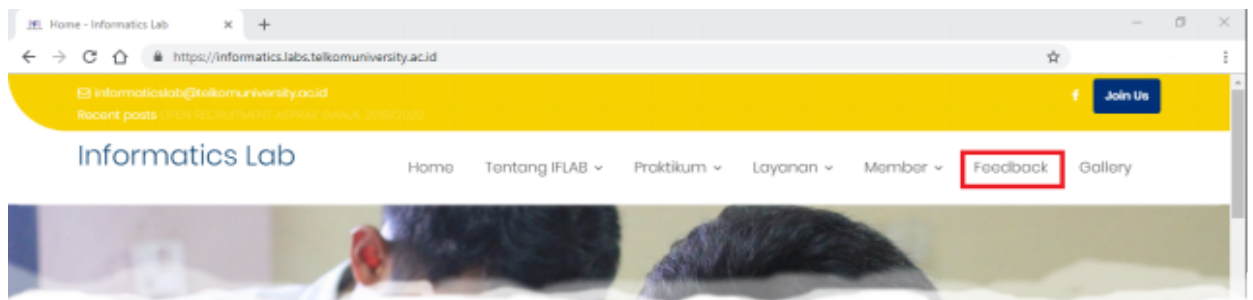


- 6) Pilih Kategori: **Pelaksanaan Praktikum**, lalu pilih **Sub Kategori**.
- 7) Isi **Deskripsi** sesuai keluhan yang ingin disampaikan.

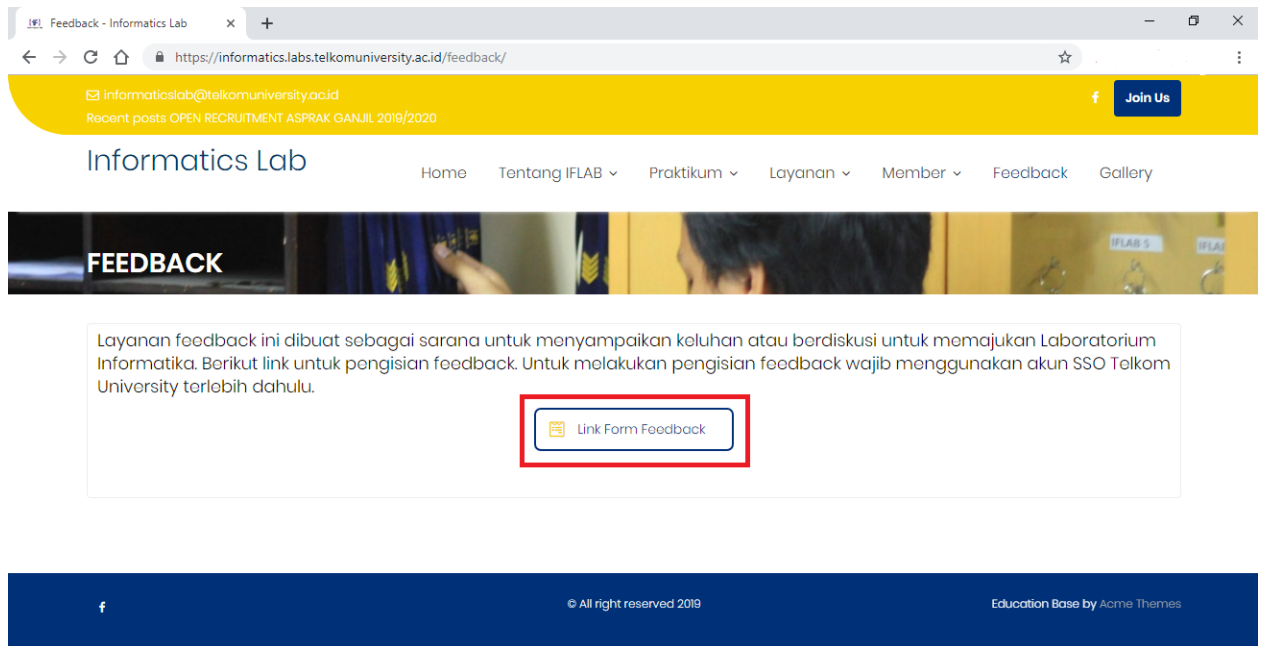
- 8) Lampirkan file jika perlu. Lalu klik Kirim.

1.3 Penyampaian Keluhan Praktikum IFLAB Melalui Situs Web

- 1) Gunakan browser untuk membuka situs <https://informatics.labs.telkomuniversity.ac.id/>
- 2) Pilih menu **Feedback** pada navigation bar website



- 3) Pilih tombol Link Form Feedback.



- 4) Lakukan login menggunakan akun SSO Telkom University untuk mengakses form feedback.
- 5) Isi form sesuai dengan feedback yang ingin diberikan.

1.4 Aturan Penulisan dan Submisi Tugas

- 1) Setiap program harus disimpan dalam file dengan ekstensi/akhiran **.go**.
- 2) Program harus dibuat rapi dan terstruktur.
 - a. Buat indentasi yang benar (Instruksi pada level yang sama dimulai dari kolom yang sama)

Instruksi di dalam suatu blok, misal di dalam struktur while-loop, harus masuk 4 spasi ke dalam dibandingkan dengan struktur kontrolnya.

Gunakan perintah **go fmt** sebelum submisi untuk memastikan program Anda mempunyai struktur yang konsisten. Pada contoh di bawah, spasi digambarkan dengan simbol `█` agar lebih jelas.

Gunakan perintah **go clean** sebelum submisi untuk membuang semua file yang tidak diperlukan di dalam folder, seperti executable code, sehingga hanya menyisakan file program sumbernya/source code saja.

```
i := 1
for i < 100 {
    █ if i%10 == 0 {
    █ █ █ █ █ fmt.Println( i*3 )
    █ █ █ █ █ }
}
```

```
|||| i = i + 1
}
fmt.Println( "Selesai" )
```

- b. Setiap baris hanya boleh berisi satu instruksi saja, walaupun instruksi tersebut sangat pendek
- 3) Pemilihan nama variabel harus sesuai dengan kebutuhan.
- JANGAN** menggunakan kata kunci sebagai bagian dari nama variabel, seperti: `_true`, `_false`, `_for`, `forloop`, ...
 - Pilih nama variabel yang berhubungan dengan nilai yang disimpan dalam variabel tersebut, seperti: `jumlah`, `rerata`, `terbesar`, `ganjil`, `ketemu`, `mhs`, `skor`, `min`, `max`, ...
 - Gunakan variabel satu huruf atau akronim untuk manfaat variabel yang sudah umum untuk huruf/akronim tersebut, seperti:
 - `i`, `j`, `k`, ... untuk indeks, iterator loop, dsb.
 - `t`, `t1`, `temp`, ... untuk variabel temporer
 - `p`, `q`, `r`, ... untuk pointer
 - `n`, `m`, ... untuk jumlah data, ukuran array, dsb.
 - `u`, `v`, `w`, `x`, `y`, `z`, ... untuk nilai dari suatu koleksi data
 - Gunakan penulisan nama variabel yang konsisten.
 - CamelCase**: pada nama berbentuk kata majemuk, setiap kata dimulai dengan huruf besar, kecuali kata pertama. Kata pertama (atau hanya satu kata) maka tetap dimulai dengan huruf kecil. Dalam bahasa Go, nama dimulai dengan huruf besar merupakan nama yang diekspor oleh paket.
Contoh: `nilaiTerkecil`, `mhsBaru`, `mhsLama`, ...
 - Dengan **`_underscore`**: pada nama berbentuk kata majemuk, kata-kata digabungkan dengan `"_"`.
Contoh `nilai_terkecil`, `mhs_baru`, `mhs_lama`, ...
 - HURUF BESAR/KAPITAL** digunakan untuk konstanta simbolik, yaitu nilainya tidak pernah, tidak mungkin, dan tidak boleh diubah dalam program.
Contoh: `MAX_MHS`, `NMAX`, ...

- Pada tugas pemrograman AP **HINDARI** pemilihan nama yang dimulai dengan underscore.

Contoh yang dilarang: `_mhs`, `_temp`, ...

- 4) Setiap algoritma dan program sumbernya selalu berlaku prinsip satu pintu masuk satu pintu keluar.
 - a. Sekuens instruksi dimulai dari instruksi pertama, eksekusi secara berurutan, dan berakhir pada satu instruksi terakhir.
 - b. Bentuk perulangan dimulai dari instruksi pertama, masuk ke dalam dan mengulang iterasi badan loop, dan kemudian keluar dari **SATU** lokasi saja, yaitu dari kondisi iterasi tersebut.
 - Jika bentuk struktur **while-loop**, maka loop berakhir dan pintu keluar dari instruksi **while** (bukan dari instruksi **endwhile**).
 - Jika bentuk struktur **repeat-until**, maka loop berakhir dan pintu keluar dari instruksi **until** (bukan dari instruksi **repeat**).
 - Bentuk lain juga menggunakan prinsip yang sama, satu pintu keluar saja, sehingga tidak diperkenankan bentuk **while** dicampur dengan adanya instruksi **break**, atau menggunakan beberapa perintah **break** dalam satu loop, atau menggunakan perintah **continue** di dalam suatu loop.
 - c. Bentuk percabangan dimulai dari instruksi pertama, kemudian mengeksekusi blok instruksi sesuai dengan kondisinya, dan berakhir pada instruksi penutup (**endif** atau **endcase**).
 - Tidak diperkenankan merenteng beberapa blok instruksi, misalkan menggunakan instruksi **fallthrough**.
 - Tidak diperkenankan keluar dari tengah blok **if** atau **switch** menggunakan perintah **break**, kecuali untuk kebutuhan keluar dari loop yang memenuhi syarat bentuk perulangan di atas.
- 5) Kelas ini mempelajari konsep pemrograman konvensional. Dalam tugas yang diberikan, tidak diperlukan instruksi yang bersifat konkurensi dan/atau paralel. Ini termasuk untuk **JANGAN** menggunakan instruksi assignment paralel.
- 6) Algoritma dan program sumbernya harus jelas dan menggunakan tipe data yang sesuai peruntukannya. Beberapa **contoh penyalahgunaan tipe data**:

- a. Tipe **integer**, tetapi nilai yang dipakai hanya 0 dan 1 (atau sejenisnya), yang manfaatnya adalah untuk menyatakan suatu keadaan (ada/tidak ada, ketemu/tidak ketemu), dsb.. Tipe yang tepat yang sebaiknya digunakan sudah jelas adalah tipe **boolean**.
 - b. Tipe **real**, tetapi untuk semua operasi yang diterapkan pada variabel tersebut tidak pernah berkaitan dengan pecahan. Tipe yang sebaiknya digunakan adalah tipe **integer**.
- 7) Algoritma dan program sumber implementasinya harus selalu bersifat efektif, tidak menggunakan instruksi yang pada dasarnya tidak memberikan efek neto yang penting. Beberapa contoh bentuk ekspresi yang berlebihan:

Bentuk yang TIDAK TEPAT	Bentuk yang SEHARUSNYA
if found == true then .. if found != false then ..	if found then ..
if found == false then .. if found != true then ..	if not found then ..
if suatu_kondisi then abc ← true else abc ← false endif	abc ← suatu_kondisi
abc ← def * 1	abc ← def
abc ← def - 0	abc ← def
abc ← def * -1	abc ← -def

- 8) Proses debugging adalah bagian dari kegiatan untuk membuat program sumber yang berjalan dengan baik dan sesuai dengan algoritma yang dirancang. Dalam proses debugging mungkin ditambahkan perintah yang input dan output untuk mencari kesalahan yang ada dalam program.
- a. Perintah seperti **fmt.Scanln()**, misalnya sebagai instruksi terakhir dalam program, digunakan untuk menghentikan eksekusi agar keluaran dapat diverifikasi.
 - b. Perintah **fmt.Println("...")** digunakan di berbagai lokasi program untuk mencetak isi variabel dan/atau ekspresi tertentu untuk memastikan kebenaran proses yang sedang terjadi.
 - c. **Akan tetapi pada saat submisi**, semua perintah tambahan tersebut harus dihapus dari program yang akan diserahkan!
 - d. Pastikan semua perintah **fmt.Print*** sudah memberikan **keluaran yang sesuai dalam format, bentuk, dan hasil seperti yang diminta** dalam tugas.

- e. Pastikan semua perintah **fmt.Scan*** memang membaca yang harus dibaca, tidak kurang dan tidak lebih.
- 9) Patuhi etika pendidikan.

Jika diperintahkan sebagai tugas mandiri, maka selayaknya untuk tidak mengutip, sebagian atau pun seluruhnya, apa adanya atau dengan modifikasi, hasil karya orang lain, rekan lain, atau tim lain.

1.5 Instalasi Program Penunjang Praktikum

Kode program dengan bahasa pemrograman Go tidak bisa langsung dijalankan oleh komputer, perlu diterjemahkan terlebih dahulu menjadi sebuah program yang dimengerti oleh komputer. Proses penerjemahan ini dikenal dengan istilah kompilasi, yang pada proses ini kode program dicek struktur atau tata bahasanya, kemudian diterjemahkan menjadi suatu program dengan bahasa yang dimengerti oleh komputer yaitu bahasa mesin. Program yang digunakan untuk menerjemahkan ini disebut Compiler, artinya setiap bahasa pemrograman memiliki compiler-nya masing-masing.

Selain compiler diperlukan juga sebuah program **text editor** khusus atau biasa disebut **Integrated Development Environment (IDE)**. Program ini digunakan untuk menuliskan kode program, dan biasanya sudah dilengkapi fitur-fitur yang membantu programmer dalam membuat suatu program.

(1) Instalasi Go Language (GoLang)

Compiler dari program Go dapat diunduh pada tautan berikut <https://go.dev/dl/>, dan proses instalasi dapat dilakukan dengan mengikuti petunjuk yang terdapat pada tautan berikut <https://go.dev/doc/install>. Kemudian pilih sesuai dengan sistem operasi yang digunakan oleh komputer.

Video tutorial instalasi Go: <https://youtu.be/k46ZjkT5-Rc>

(2) Instalasi Integrated Development Environment (IDE)

Pada pemrograman terdapat banyak sekali IDE yang bisa digunakan, berikut ini adalah beberapa IDE yang umum digunakan untuk pemula.

- a. Notepad++ (<https://notepad-plus-plus.org/downloads/>)
- b. Visual Studio Code (<https://code.visualstudio.com/>)
- c. Go Land (<https://www.jetbrains.com/go/>)

Ikuti setiap petunjuk instalasi yang diberikan oleh masing-masing IDE tersebut.