

PERATURAN PRAKTIKUM

1. Praktikum diampu oleh dosen kelas dan dibantu oleh asisten laboratorium dan asisten praktikum.
2. Praktikan wajib memakai pakaian berkerah dan sesuai aturan institusi selama kegiatan praktikum.
3. Praktikan wajib hadir 10 menit sebelum jadwal praktikum yang ditentukan.
4. Praktikan wajib hadir minimal 75% dari seluruh pertemuan praktikum (sesuai ketentuan institusi).
5. Praktikum dilaksanakan di Gedung Telkom University Lanmark Tower (TULT) Lantai 6 dan Lantai 7 sesuai jadwal yang ditentukan.
6. Durasi kegiatan praktikum S-1 = 2 jam (100 menit).
7. Jumlah pertemuan praktikum sebanyak 16 kali.
8. Dosen berhak melarang praktikan masuk ataupun mengeluarkan praktikan yang tidak mematuhi aturan praktikum.
9. Praktikan yang datang terlambat :
 - ≤ 30 menit : diperbolehkan mengikuti praktikum tanpa tambahan waktu pengerjaan praktikum.
 - >30 menit : tidak diperbolehkan mengikuti praktikum.
10. Saat praktikum berlangsung, asisten praktikum dan praktikan:
 - Wajib menggunakan seragam sesuai aturan institusi.
 - Wajib mematikan/ mengkondisikan semua alat komunikasi.
 - Dilarang membuka aplikasi yang tidak berhubungan dengan praktikum yang berlangsung.
 - Dilarang mengubah pengaturan *software* maupun *hardware* komputer tanpa ijin.
 - Dilarang meninggalkan tempat selama tidak diizinkan oleh asisten.
 - Dilarang bekerjasama atau kecurangan lainnya selama praktikum.
 - Dilarang membawa makanan maupun minuman di ruang praktikum.
 - Dilarang memberikan jawaban ke praktikan lain.
 - Dilarang menyebarkan soal praktikum.
 - Dilarang membuang sampah di ruangan praktikum.
 - Wajib meletakkan alas kaki dengan rapi pada tempat yang telah disediakan.
 - Wajib menjaga sopan santun kepada sesama praktikan maupun asisten.
11. Setiap praktikan dapat mengikuti praktikum susulan maksimal dua modul untuk satu mata kuliah praktikum.
 - 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.)
 - Persyaratan untuk praktikum susulan diserahkan sesegera mungkin kepada asisten laboratorium untuk keperluan administrasi.
 - Praktikan yang diijinkan menjadi peserta praktikum susulan ditetapkan oleh Asman Lab dan Bengkel Informatika dan tidak dapat diganggu gugat.

SELAMAT MENGERJAKAN!

KONSEKUENSI PELANGGARAN ATURAN PRAKTIKUM

1. Keterlambatan menghadiri kelas praktikum (max 30 menit)
 - Tidak diperkenankan mengikuti kegiatan praktikum
2. Ketidakhadiran pada kelas praktikum
 - Absensi dibawah 75% = nilai '0' pada assessment akhir/tubes
3. Meminta, mendapatkan, dan menyebarluaskan soal atau kunci jawaban praktikum
 - Penyebar soal dan kunci jawaban : Pengajuan sanksi kepada Komisi Disiplin Fakultas
 - Penerima soal dan kunci jawaban : Nilai '0' pada (seluruh assessment) praktikum
4. Lupa menghapus file praktikum
 - Pengurangan nilai assessment

SELAMAT MENGERJAKAN!

JURNAL MODUL 4

1. MEMBUAT PROJECT CONSOLE/TANPA GUI

Buka IDE misalnya dengan Visual Studio

A. Pilih “Create a new project” kemudian pilih “Console App”. Pada IDE lain pada umumnya hanya perlu membuat project baru saja.

B. Masukkan project baru dengan nama modul5_NIM.

2. MELAKUKAN GIT COMMIT PADA PROJECT YANG DIBUAT

Task atau langkah-langkah yang perlu dikerjakan adalah sebagai berikut:

A. Buatlah github public repository kosong (pastikan bagian “*Initialize this repository with*” tidak ada yang dicentang pada saat membuat repository baru) melalui <https://github.com/>

B. Melakukan inisialisasi dengan “git init” di folder project yang dibuat atau menggunakan IDE.

C. Membuat commit untuk versi pertama dari project yang dibuat dengan pesan bebas.

D. Melakukan git push ke github repository.

3. MEMBUAT BRANCH BARU PADA GIT PROJECT

Pada folder project yang sedang aktif, langkah-langkah yang perlu dikerjakan:

A. Membuat git branch baru dengan nama “implementasi-generic-method”.

B. Berpindah ke branch yang sudah dibuat sebelumnya yaitu “implementasi-generic-method”.

4. MENAMBAHKAN METHOD DENGAN GENERIC

Tanpa membuat file baru (gunakan file yang dibuat saat membuat project):

A. Buatlah sebuah class bernama “Penjumlahan”.

B. Pada class tersebut, tambahkan sebuah method dengan nama “JumlahTigaAngka” yang memiliki tiga parameter generic yang sama yaitu “T”

C. Method tersebut dapat melakukan penjumlahan dari tiga input/argument yang diberikan pada method tersebut.

D. **Hint:** gunakan variable sementara dengan tipe data **dynamic** untuk memungkinkan operasi matematis misalnya penjumlahan.

E. Panggil method tersebut pada fungsi/method utama dengan tiga input angka yaitu 2-digit dari NIM. Misalnya NIM 12345678, maka tiga input angka yaitu “12”, “34” dan “56” dengan tipe data sebagai berikut:

SELAMAT MENGERJAKAN!

- i. NIM berakhiran 1 atau 2: tipe data input float
- ii. NIM berakhiran 3, 4 atau 5: tipe data input double
- iii. NIM berakhiran 6, 7 atau 8: tipe data input int
- iv. NIM berakhiran 9 atau 0: tipe data input long

5. MELAKUKAN COMMIT, PUSH DAN PINDAH BRANCH

Pada branch yang sedang aktif saat ini (branch “implementasi-generic-method”):

- A. Lakukan commit dengan pesan “menambahkan class Penjumlahan dengan method JumlahTigaAngka”.
- B. Lakukan push ke github pada branch “implementasi-generic-method” di remote/github repo.
- C. Setelah proses push berhasil, ganti branch yang aktif ke master/main branch.
- D. Pada branch master/main, buat branch baru dengan nama “implementasi-generic-class”.
- E. Setelah itu, lakukan pindah branch ke branch “implementasi-generic-class” yang sudah dibuat.

6. MENAMBAHKAN METHOD DENGAN GENERIC

Tanpa membuat file baru (gunakan file yang dibuat saat membuat project dan pastikan branch aktif adalah pada branch implementasi-generic-class):

- A. Buatlah sebuah class bernama “SimpleDataBase” dengan mengikuti class model yang ditunjukkan pada gambar/tabel di bawah ini.

SimpleDataBase
- storedData: List<T> - inputDates: List<Date>
+ SimpleDataBase() + AddNewData(T) + PrintAllData(): void

SELAMAT MENGERJAKAN!

B. Class tersebut memiliki dua property yaitu:

- i. Property “storedData” yang merupakan suatu List (struktur data bawaan/default) yang berisi data bertipe generic “T”.
- ii. Property “inputDates” yang bertipe List<Date> (atau tipe data List<DateTime> di C#) yang akan list dari waktu input.

C. Class tersebut juga memiliki beberapa method yaitu:

- i. Konstruktor SimpleDataBase() yang akan membuat property “storedData” berisi List kosong.
- ii. Method AddNewData(T) yang akan menambahkan data baru bertipe T ke dalam list “storedData” dan waktu saat itu (Now) ke dalam list “inputDates”.
- iii. Method PrintAllData() yang akan memberikan output console berupa teks yang menampilkan seluruh data yang tersimpan pada “storedData” dan “inputDates”, contohnya:
 - Data 1 berisi: 12, yang disimpan pada waktu UTC: 3/10/2022 5:32:01 AM
 - Data 2 berisi: 34, yang disimpan pada waktu UTC: 3/10/2022 5:32:02 AM
 - Data 2 berisi: 56, yang disimpan pada waktu UTC: 3/10/2022 5:32:02 AM

D. Panggil method PrintAllData() pada fungsi/method utama setelah menambahkan tiga data yang berisi dan bertipe dua-digit NIM seperti pada bagian 4E.

7. MELAKUKAN COMMIT, PUSH DAN PINDAH BRANCH BAGIAN 2

Pada branch yang sedang aktif saat ini (branch “implementasi-generic-class”):

- A. Lakukan commit dengan pesan “menambahkan class SimpleDataBase”.
- B. Lakukan push ke github pada branch “implementasi-generic-class” di remote/github repo.
- C. Setelah proses push berhasil, ganti branch yang aktif ke master/main branch.

8. MELAKUKAN GIT MERGE DARI KEDUA BRANCH BARU

Pastikan branch aktif adalah branch master/main:

- A. Lakukan git merge branch “implementasi-generic-method” ke branch master/main.
- B. Lakukan git merge branch “implementasi-generic-class” ke branch master/main, dan jika terjadi merge conflict, pastikan semua baris yang conflict sudah diperbaiki.
- C. Lakukan git push untuk branch master/main ke github repository.

SELAMAT MENGERJAKAN!

9. PENGUMPULAN FILE/TUGAS JURNAL

Sebelum pengumpulan, praktikan wajib menunjukkan hasil run via share screen ke asprak.
Kumpulkan semua file berikut dalam bentuk file zip/rar/7zip:

- A. Source code dari project yang dibuat
- B. File docx/pdf yang berisi:
 - i. Link github repository
 - ii. Screenshot hasil run (baik dari table-driven maupun state-based)
 - iii. Penjelasan singkat dari kode implementasi yang dibuat (beserta screenshot dari potongan source code yang dijelaskan).

10. KOMPONEN PENILAIAN

- A. Integrasi dengan Github [10 pts]
- B. Penggunaan git [10 pts]
- C. Implementasi generic method [30 pts]
- D. Implementasi generic class [35 pts]
- E. Laporan jurnal [15 pts]

SELAMAT MENGERJAKAN!