



## INSTITUT TEKNOLOGI DEL

### UJIAN TENGAH SEMESTER

Semester Ganjil 2024/2025

Hari/Tanggal Ujian	Kamis / 17 Oktober 2024
Kode/ Nama Mata Kuliah	4143101 / Algoritma Lanjut
Jenis Ujian	PRAKTIKUM
Sifat Ujian	Open Source Code (Hanya Pada Source Code Latihan Kelas, Dilarang Browsing kecuali 30 menit terakhir)
Waktu Pengerjaan	180 menit (150 <b>menit</b> : waktu pengerjaan soal dan 30 <b>menit</b> : waktu pengumpulan dengan ketentuan yang sudah diberikan)
Pengajar	TAP Indah

#### **PETUNJUK**

**Sebelum mengerjakan soal ujian, Anda harus membaca petunjuk pengerjaan berikut:**

1. Bekerja sendiri, tidak diperkenankan bekerja sama dengan cara apa pun. Kegagalan memenuhi aturan ini akan memperoleh sanksi akademik.
2. Anda diizinkan membuka segala sumber dari laptop/PC anda sendiri dan tidak diperkenankan untuk mengakses jaringan selama ujian berlangsung.
3. Bacalah setiap soal dengan rinci sebelum mulai menjawab. Kesempatan bertanya diperbolehkan pada 15 menit pertama ujian.
4. Anda diwajibkan membuat video berupa demo pekerjaan Anda [tanpa suara tidak apa] kemudian kompress keseluruhan *file* anda baik video dan kode program dengan penamaan sebagai berikut **NIM\_uts\_D3TI.zip** cth. 11323323\_uts\_D3TI.zip kemudian upload di google drive masing-masing dan **WAJIB** beri akses ke email [arifintegar12@gmail.com](mailto:arifintegar12@gmail.com) dan link nya di kumpulkan pada google form pada link berikut: [https://bit.ly/Pengumpulan\\_UTS\\_ALU\\_D4TRPL\\_del\\_2024](https://bit.ly/Pengumpulan_UTS_ALU_D4TRPL_del_2024)
5. Pada ecourse kumpulkan versi .pdf , jadi anda di **WAJIB** kan membuat laporan atau summary kode program yang anda buat beserta penjelasannya [**WAJIB** dijelaskan] serta **WAJIB** ditunjukkan/Screenshot Outputnya Juga dan buat di word save ekstensi .pdf dengan filename **NIM\_Nama\_UTS\_ALU** dan kumpulkan pada link berikut: <https://ecourse.del.ac.id/mod/assign/view.php?id=49316>
6. Jumlah halaman 3 termasuk sampul.

Soal-soal berikut dibebaskan dikerjakan menggunakan bahasa pemrograman apa saja namun disarankan menggunakan python atau C#. Soal-soal berikut juga dibebaskan input dan outputnya namun diberikan contoh input dan outputnya.

1. Kompleksitas Waktu dan Rekursi: Buatlah dua fungsi untuk menghitung bilangan Fibonacci ke- $n$ :
  - Fungsi pertama menggunakan pendekatan rekursif sederhana.
  - Fungsi kedua menggunakan teknik optimisasi (misalnya dynamic programming) untuk memperbaiki kompleksitas waktu.

Jelaskan kompleksitas waktu dari kedua metode yang kamu buat.

Contoh hasil:

Input: Bilangan bulat  $n$  ( $0 \leq n \leq 50$ )

Output: Bilangan Fibonacci ke- $n$

2. Divide and Conquer: Diberikan sebuah array  $A$  berisi  $n$  bilangan bulat, temukan subarray dengan jumlah elemen terbesar menggunakan algoritma divide and conquer.

Contoh hasil:

Input: Array  $A$  dengan  $n$  elemen ( $-1000 \leq A[i] \leq 1000, 1 \leq n \leq 1000$ )

Output: Jumlah maksimum dari subarray terbesar

3. Algoritma Greedy: Aktivitas Terbanyak Diberikan sejumlah aktivitas dengan waktu mulai dan waktu selesai, kamu diminta memilih aktivitas yang bisa dilakukan sebanyak mungkin, tanpa tumpang tindih waktu.

Contoh hasil:

Input: Array berisi pasangan waktu mulai dan selesai dari  $n$  aktivitas ( $1 \leq n \leq 100$ )

Output: Jumlah maksimum aktivitas yang bisa dilakukan

4. Algoritma Backtracking: Permutasi yang Unik Diberikan sebuah array berisi angka-angka (mungkin terdapat angka yang sama), buatlah semua permutasi yang unik menggunakan pendekatan backtracking.

Contoh hasil:

Input: Array berisi  $n$  bilangan bulat ( $1 \leq n \leq 8$ )

Output: Semua permutasi unik yang mungkin

5. Dynamic Programming: Diberikan sebuah ransel dengan kapasitas tertentu dan beberapa barang dengan berat dan keuntungan masing-masing, tentukan keuntungan maksimum yang bisa diperoleh tanpa melebihi kapasitas ransel menggunakan dynamic programming.

Contoh hasil:

Input:

Kapasitas ransel  $W (1 \leq W \leq 1000)$

Berat dan keuntungan dari  $n$  barang ( $1 \leq n \leq 100, 1 \leq \text{berat}[i], \text{keuntungan}[i] \leq 1000$ )

Output: Keuntungan maksimum yang dapat diperoleh

Buatlah web sederhana yang mengintegrasikan jawaban dari soal-soal di atas. Web ini harus memenuhi kriteria berikut:

- Memiliki halaman input untuk setiap soal (dengan form input sederhana).
- Menampilkan hasil dari setiap soal di halaman hasil setelah proses dijalankan.
- Web ini boleh dibuat menggunakan framework sederhana seperti Flask atau Node.js atau yang lain sesuai kemampuan Anda.
- Deploy web di platform gratis seperti Heroku atau Vercel [Bonus jika sempat di 30 menit terakhir bisa pakai internet]

Catatan:

- Solusi setiap soal dapat dijalankan di server-side dan hasilnya dikirimkan kembali ke client-side.
- Untuk setiap soal, tampilkan waktu eksekusi (dalam milidetik) di halaman hasil untuk menunjukkan kompleksitas waktu.

[Bonus Besar] Jika memiliki waktu, tambahkan fitur untuk menghitung performa dan visualisasi perbandingan waktu eksekusi dari beberapa algoritma yang digunakan, misalnya perbandingan antara rekursi Fibonacci dan Fibonacci dengan dynamic programming.