



Institusi : Universitas Pendidikan Ganesha
Fakultas : Teknik dan Kejuruan
Prodi : Ilmu Komputer

Rencana Tugas Mahasiswa (RTM)

MATA KULIAH	Desain dan Analisis Algoritma	SKS: 3	SEMESTER: IV
KODE MATA KULIAH	KOMS120403		
DOSEN PENGAMPU	Ni Luh Dewi Sintuari, Ph.D.		
TUGAS KE	1	MINGGU KE:	1
BENTUK TUGAS	Tugas Mandiri (Kelompok 2 orang)		

JUDUL TUGAS

Penulisan Makalah Ilmiah tentang Penerapan Strategi Algoritma untuk Penyelesaian Masalah Dunia Nyata

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Sub-CPMK1 sampai dengan Sub-CPMK10

DESKRIPSI TUGAS

Tugas dikerjakan secara mandiri, atau berkelompok 2 orang. Penulisan makalah mengikuti format dan aturan penulisan yang diberikan (deskripsi tertulis pada lembar soal, dan format penulisan dalam file .doc).

Selanjutnya, setiap individu/kelompok mengumpulkan video presentasi berdurasi sekitar 10 menit, yang diunggah melalui platform youtube. Video menjelaskan isi makalah secara keseluruhan.

1 Aturan pengerjaan tugas

1. Anda dipersilahkan untuk mengerjakan tugas ini secara individual maupun berkelompok (yang beranggotakan 2 orang, boleh lintas kelas). Jika memilih untuk mengerjakan secara berkelompok, hendaknya pembagian tugas dapat dilakukan secara seimbang. Makalah yang dihasilkan diharapkan dapat lebih menarik (misal: topik makalah yang menarik disertai metode/pembahasan yang lebih jelas, didukung dengan eksperimen menggunakan program komputer).
2. Tugas dikumpulkan dalam format pdf, dengan format penamaan tugas: **DAA02_Kelas_Nama Lengkap_NIM**. Pengumpulan tugas melalui e-learning Undiksha.
Contoh: **DAA02_6A_Gede Ganesha (1610101001)** atau **DAA02_6A_Gede Ganesha (1610101001)_Luh Sukreni (1610101002)** jika berkelompok.
3. Makalah yang dibuat tidak boleh berupa saduran, terjemahan, maupun plagiasi hasil karya orang lain. Di samping studi literatur, makalah juga harus menyertakan kontribusi orisinil Anda, misalnya berupa ide untuk menyelesaikan permasalahan tersebut dengan metode baru, proposisi algoritma baru atau pengembangan algoritma yang sudah ada, hasil eksperimen atau pemrograman komputer yang berkaitan dengan permasalahan yang diangkat.
4. Rujukan yang dijadikan dasar studi literatur atau referensi harus merupakan karya tulis atau artikel ilmiah, surat kabar atau media resmi, dan bukan dari media/website yang tidak dapat dibuktikan kredibilitasnya. Wikipedia dapat dijadikan sumber bacaan, namun hendaknya tidak menjadi daftar referensi (silahkan merujuk pada artikel ilmiah yang biasanya dicantumkan pada halaman wikipedia).

5. Mencantumkan gambar, tabel, diagram, dan sejenisnya yang dikutip dari karya orang lain harus menyertakan sumber aslinya. Pengutipan definisi secara keseluruhan dari karya orang lain harus menyertakan sitasi.
6. Kriteria penilaian mengikuti rincian berikut: kesesuaian format penulisan; penggunaan bahasa ilmiah; kesesuaian dengan deskripsi tugas yang diberikan; orisinalitas ide dan penulisan; kesesuaian metode dengan permasalahan yang diajukan; pembahasan (pemodelan, implementasi program komputer, analisis hasil); kejelasan tulisan pada setiap bab (pendahuluan, landasan teori, pembahasan, kesimpulan, daftar pustaka).

Dengan ini, Anda menyatakan bahwa Anda siap menerima segala konsekuensi jika nantinya ditemukan adanya kecurangan dalam pengerjaan tugas ini.

2 Deskripsi tugas

Makalah yang ditulis adalah makalah ilmiah yang berkaitan dengan aplikasi salah satu metode/algorithm yang dibahas dalam perkuliahan, dengan tema: "Aplikasi Desain dan Analisis Algoritma untuk Menyelesaikan *Real-World Problems*". Pilihlah salah satu topik berikut:

- Algoritma Brute Force/Exhaustive Search
- Metode Rekursif
- Algoritma Divide and Conquer
- Algoritma Decrease and Conquer
- Algoritma DFS dan BFS
- Algoritma Backtracking (Runut balik)
- Algoritma Branch and Bound
- Algoritma Greedy
- Permasalahan Graf (Shortest Path, Minimum Spanning Tree, Coloring, dsb.)
- Teori P, NP, dan NP-Complete

Makalah ditulis dalam Bahasa Indonesia dengan menggunakan bahasa ilmiah sesuai aturan KBBI. Jumlah halaman adalah minimal 6 halaman dan maksimal 10 halaman. Kertas yang digunakan berukuran A4. Penyertaan gambar harus dibuat seminimal mungkin dan disesuaikan dengan artikel (tidak terlalu besar), dan *bukan dengan tujuan untuk menambah jumlah halaman*.

Format penulisan sesuai dengan template yang diberikan (dapat diunduh di [link](#) berikut, dengan komponen sebagai berikut: judul, nama penulis, NIM, instansi, abstrak, kata kunci, isi makalah (Pendahuluan, Kajian Pustaka/Landasan Teori dan Metodologi, Implementasi dan Pembahasan, Kesimpulan), apendiks (jika ada), dan daftar pustaka. Anda diizinkan untuk menambahkan maksimal 2 halaman lampiran (spesifikasi program komputer atau pembuktian teorema, jika Anda membuktikan sendiri dan bukan merupakan kutipan dari sumber lain). Lebih lengkap, struktur makalah dapat dirinci sebagai berikut:

- *Abstrak*: ringkasan mengenai isi makalah ilmiah. Abstrak harus ditulis secara jelas dan padat, maksimum 200 kata.
- *Pendahuluan*: berisi latar belakang, rumusan dan batasan masalah, dan tujuan.
- *Landasan Teori*: berisi seperangkat definisi, konsep, dan teori yang dituliskan secara sistematis.
- *Pembahasan*: berisi diskusi terkait dengan jawaban permasalahan yang diajukan, berdasarkan pada hasil eksperimen, atau implementasi pemrograman komputer. Detail implementasi algoritma dalam bahasa pemrograman tidak perlu dituliskan dalam bab ini. Jika dibutuhkan, silahkan jelaskan pada bagian Apendiks.
- *Kesimpulan*: berisi penjelasan mengenai keseluruhan hasil penelitian atau jawaban permasalahan, beserta saran untuk pengembangan.

3 Contoh draft makalah

Berikut adalah contoh draft makalah, yang dapat dijadikan gambaran/acuan untuk penulisan makalah. Makalah yang Anda tulis tidak harus secara persis mengikuti draft berikut, dan dapat disesuaikan dalam beberapa aspek sesuai dengan kebutuhan Anda.

Judul: Penerapan algoritma Greedy untuk pengaturan jadwal kuliah

Latar belakang/rumusan masalah:

- Deskripsi/penjelasan tentang masalah penjadwalan mata kuliah;
- Batasan-batasan apa saja yang terkait dengan penyusunan jadwal (misal: jadwal mata kuliah, daftar mahasiswa yang mengambil mata kuliah tertentu, jadwal dosen, ketersediaan ruang kuliah, batas jam pembelajaran);
- Kendala apa yang dialami jika jadwal disusun dengan metode konvensional;
- Bagaimana keterkaitan desain algoritma untuk mengatasi permasalahan tersebut (algoritma apa yang dapat digunakan sebagai pendekatan untuk pemecahan masalah tersebut, dan bagaimana pendekatannya);
- Review/ulasan singkat mengenai beberapa contoh masalah penjadwalan beserta metode/pendekatan untuk menjawab permasalahan tersebut, disertai dengan sitasi/rujukan pada artikel ilmiah. Sebagai contoh, dalam hal ini algoritma Greedy dapat menjadi salah satu metode untuk pengaturan jadwal mata kuliah. Maka dapat dicari contoh artikel yang membahas permasalahan serupa dengan algoritma Greedy.

Landasan Teori/Metode:

- Definisi/ulasan/deskripsi terkait dengan “Masalah Penjadwalan”.
- Definisi/ulasan/deskripsi terkait dengan “Algoritma Greedy”, serta bagaimana algoritma ini dapat diaplikasikan dalam masalah penjadwalan tersebut.
- Pemodelan matematis yang mendeskripsikan masalah pengaturan jadwal, disertai dengan deskripsi bagaimana *constraints* (batasan) masalah (seperti jadwal mata kuliah, daftar mahasiswa, jadwal dosen, dsb.) dapat dimodelkan secara matematis
- Perancangan algoritma Greedy untuk menyelesaikan masalah pengaturan jadwal sesuai dengan pemodelan yang dirancang. Implementasi algoritma dalam bahasa pemrograman.
- Persiapan dan pemilihan sampel data. Jika tidak dimungkinkan untuk mengambil data riil (misal dengan menggunakan data mata kuliah/data mahasiswa/data dosen di Prodi Ilkom), maka dapat digunakan data fiktif yang sesuai dengan pemodelan yang dirancang.
- Hal-hal lain yang terkait dapat ditambahkan.

Pembahasan

- Analisis data, pengujian secara matematis, implementasi algoritma dalam bahasa pemrograman, hasil eksperimen, analisis hasil eksperimen, dan diskusi mengenai analisis matematis dan hasil pengujian dengan program (disertai tabel, gambar, dsb. jika dibutuhkan).

Kesimpulan dan Saran: berisi kesimpulan tentang jawaban dari permasalahan penjadwalan tersebut, serta saran-saran terkait.

TAHAPAN Pengerjaan Tugas

1. Mengikuti perkuliahan Strategi Algoritma sesuai dengan topik yang ingin dibahas.
2. Mengajukan topik makalah kepada dosen pengampu.
3. Mengumpulkan data-data melalui buku dan internet, dan menyusun makalah secara mandiri atau berkelompok, dengan menyertakan hasil eksperimen sesuai dengan topik yang dibahas
4. Mengumpulkan makalah dalam format pdf.
5. Membuat video presentasi makalah dan mengunggah di platform youtube.

BENTUK DAN FORMAT LUARAN	
File makalah dalam format pdf, dan link video presentasi.	
INDIKATOR, KRITERIA DAN BOBOT PENILAIAN	
Penilaian dengan angka skor (0-100) Kriteria penilaian: 1. Orisinalitas ide, penyelesaian dan eksperimen, penulisan: 60% 2. Presentasi: 40% 3. Kontribusi dan ketepatan pengumpulan tugas: penambahan/pengurangan skor	
JADWAL PELAKSANAAN	WAKTU
Pemberian tugas dan materi terkait	Pertemuan 1
Pengumpulan tugas	Pertemuan 12
LAIN-LAIN	
Kriteria penilaian sesuai dengan deskripsi pada lembar penugasan. Penilaian secara umum meliputi kebenaran dan kejelasan penyelesaian tugas, integritas dalam pengerjaan tugas, ketepatan waktu penyelesaian tugas, dan kontribusi mahasiswa dalam penyelesaian tugas.	
DAFTAR RUJUKAN	
<ul style="list-style-type: none"> • Materi perkuliahan seluruh pertemuan • Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc • Sumber internet sesuai dengan permasalahan/topik yang dibahas 	



Institusi : Universitas Pendidikan Ganesha
Fakultas : Teknik dan Kejuruan
Prodi : Ilmu Komputer

Rencana Tugas Mahasiswa (RTM)

MATA KULIAH	Desain dan Analisis Algoritma	SKS:	3	SEMESTER:	VI
KODE MATA KULIAH	KOMS119601				
DOSEN PENGAMPU	Ni Luh Dewi Sintuari, Ph.D.				
TUGAS KE	2	MINGGU KE:		2	
BENTUK TUGAS	Tugas Mandiri				

JUDUL TUGAS

Tugas tertulis tentang Analisis Kompleksitas Waktu

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu menghitung kompleksitas waktu algoritma (worst-case, best-case, average-case), menggunakan notasi Big-O, Big-Omega, dan Big-Theta, dan mengklasifikasikan algoritma berdasarkan kompleksitas waktunya dengan benar

DESKRIPSI TUGAS

Mengerjakan soal-soal berikut dengan baik dan benar. Tugas mandiri, mahasiswa diizinkan untuk mendiskusikan metode penyelesaian. Namun, mahasiswa harus mengerjakan dan menuliskan jawaban secara mandiri, dengan menggunakan Bahasa sendiri.

1. (Computing time complexity, 15 point)

1. What is the time complexity of the “naive gcd algorithm” and the “Euclidean algorithm” explained in the lecture? Justify your answer!

Algorithm 1 Naive gcd algorithm of two integers

```
1: procedure GCD( $m, n$ )
2:    $r = 1$ 
3:    $x = \min(m, n)$ 
4:   for  $i = 1$  to  $x$  do
5:     if  $a \text{ MOD } i == 0$  and  $b \text{ MOD } i == 0$  then  $r = i$ 
6:   end if
7: end for
8: end procedure
```

Algorithm 2 Euclidean algorithm

```
1: procedure EUCLIDGCD( $m, n$ )
2:   while  $b \neq 0$  do
3:      $r = a \text{ MOD } b$ 
4:      $a = b$ 
5:      $b = r$ 
6:   end while
7:   return  $a$ 
8: end procedure
```

2. Compute the best-case, worst-case, and average-case complexities of the following algorithm!

Algorithm 3 Sequential search

```

1: procedure SEQSEARCH( $A[1..n], x$ )
2:   found  $\leftarrow$  False
3:    $i \leftarrow 1$ 
4:   while (not found) and ( $i \leq N$ ) do
5:     if ( $A[i] = x$ ) then found  $\leftarrow$  True
6:     else  $i \leftarrow i + 1$ 
7:   end if
8: end while
9: if (found) then index  $\leftarrow i$ 
10: else index  $\leftarrow 0$ 
11: end if
12: end procedure

```

2. (Dominant terms in a function, 20 point)

Select the dominant term(s) and specify the lowest \mathcal{O} -complexity of each algorithm! Give a short proof or an explanation for each function to justify your answer!

	Expression	Dominant term(s)	$\mathcal{O}(\cdot)$
1.	$5 + 0.001n^3 + 0.025n$		
2.	$500n + 100n^{1.5} + 50n \log_{10} n$		
3.	$0.3n + 5n^{1.5} + 2.5 \cdot n^{1.75}$		
4.	$n^2 \log_2 n + n(\log_2 n)^2$		
5.	$n \log_3 n + n \log_2 n$		
6.	$3 \log_8 n + \log_2 \log_2 \log_2 n$		
7.	$2n + n^{0.5} + 0.5n^{1.25}$		
8.	$0.01n \log_2 n + n(\log_2 n)^2$		
9.	$100n \log_3 n + n^3 + 100n$		
10.	$0.003 \log_4 n + \log_2 \log_2 n$		

3. (Comparing time complexities of two algorithms, 20 point)

- Suatu algoritma memiliki kompleksitas waktu $\mathcal{O}(f(n))$ dan *running time* $T(n) = cf(n)$. Jika algoritma tersebut membutuhkan waktu 10 detik untuk memproses 1000 data, berapakah waktu yang dibutuhkan untuk memproses 100,000 data jika $f(n) = n$? Bagaimana halnya jika $f(n) = n^3$?
- Algoritma **A** dan **B** masing-masing memiliki kompleksitas waktu $T_A(n) = 5n \log_{10} n$ and $T_B(n) = 25n$ microdetik, jika diberikan input dengan ukuran n . Dalam aturan Big-O, algoritma manakah yang bekerja lebih baik? Berikan batasan nilai n agar algoritma **B** memiliki kompleksitas waktu yang lebih baik!
- Suatu perangkat lunak **A** dan **B** masing-masing memiliki kompleksitas waktu $\mathcal{O}(n \log n)$ dan $\mathcal{O}(n^2)$. Keduanya memiliki *running time* $T_A(n) = c_A n \log_{10} n$ dan $T_B(n) = c_B n$ milidetik untuk memproses data dengan ukuran n . Selama proses pengujian, waktu rata-rata pemrosesan $n = 10^4$ data dengan perangkat **A** dan **B** masing-masing adalah 100 milidetik dan 500 detik. Berikan batasan nilai n agar perangkat **A** bekerja lebih cepat dibandingkan perangkat **B**, dan berikan batasan nilai n untuk kondisi sebaliknya. Perangkat mana yang Anda rekomendasikan jika kita harus memproses data dengan ukuran tidak lebih dari $n = 10^9$?

4. (Formal proof, 20 point)

- (a) Using the formal definition of Big-O notation, prove the following:

$$T(n) = a_0 + a_1n + a_2n^2 + a_3n^3 \in \mathcal{O}(n^3)$$

Hint: temukan konstanta c dan batasan nilai n_0 sedemikian sehingga $cn^3 \geq T(n)$ untuk $n \geq n_0$.

- (b) Let $T_1(n) \in \mathcal{O}(f(n))$ and $T_2(n) \in \mathcal{O}(g(n))$. Prove the following:

- (a) $T_1(n) + T_2(n) \in \mathcal{O}(f(n)) + \mathcal{O}(g(n)) \in \mathcal{O}(\max(f(n), g(n)))$
- (b) $T_1(n) \cdot T_2(n) \in \mathcal{O}(f(n)) \cdot \mathcal{O}(g(n)) \in \mathcal{O}(f(n) \cdot g(n))$
- (c) $\mathcal{O}(cf(n)) \in \mathcal{O}(f(n))$, where c is a constant
- (d) $f(n) \in \mathcal{O}(f(n))$

5. (Big-O, Big-Ω, and Big-Θ, 5 point / question, total 40 point)

Let $f, g : \mathbb{N}^+ \rightarrow \mathbb{N}^+$ be two functions. Prove or invalidate each of the following:

1. If $f(n) \in \mathcal{O}(g(n))$ then $g(n) \in \mathcal{O}(f(n))$
2. $f(n) + g(n) \in \Theta(\min(f(n), g(n)))$ (assuming that it exists)
3. If $f(n) \in \mathcal{O}(g(n))$, then $\log(f(n)) \in \mathcal{O}(\log(g(n)))$
4. If $f(n) \in \mathcal{O}(g(n))$ then $2^{f(n)} \in \mathcal{O}(2^{g(n)})$
5. $f(n) \in \mathcal{O}(f^2(n))$
6. If $f(n) \in \mathcal{O}(g(n))$ then $g(n) \in \Omega(f(n))$
7. $f(n) \in \Theta(f(n/2))$
8. If $g(n) \in \mathcal{O}(f(n))$ then $f(n) + g(n) \in \Theta(f(n))$

TAHAPAN Pengerjaan Tugas

1. Mengikuti perkuliahan materi *Analisis Kompleksitas Waktu*
2. Berdiskusi, mencari sumber pembelajaran di internet, dan mengerjakan tugas

Bentuk dan Format Luaran

Jawaban dikumpulkan dalam format pdf

Indikator, Kriteria dan Bobot Penilaian

Penilaian dengan angka skor (0-100)

Kriteria penilaian:

1. Ketepatan jawaban (30%)
2. Keruntutan langkah penyelesaian soal (70%)
3. Kejujuran/integritas dan ketepatan waktu: penambahan/pengurangan skor

Jadwal Pelaksanaan

Waktu

Pemberian tugas dan materi terkait

Pertemuan 2

Pengumpulan tugas

Pertemuan 4

Lain-lain

Kebenaran dan kejelasan penyelesaian tugas, integritas dalam pengerjaan tugas, ketepatan waktu penyelesaian tugas

Daftar Rujukan

- Materi perkuliahan pertemuan 2
- Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc



Institusi : Universitas Pendidikan Ganesha
Fakultas : Teknik dan Kejuruan
Prodi : Ilmu Komputer

Rencana Tugas Mahasiswa (RTM)

MATA KULIAH	Desain dan Analisis Algoritma	SKS: 3	SEMESTER: VI
KODE MATA KULIAH	KOMS119601		
DOSEN PENGAMPU	Ni Luh Dewi Sintiar, Ph.D.		
TUGAS KE	3	MINGGU KE:	5
BENTUK TUGAS	Tugas Mandiri		

JUDUL TUGAS

Tugas teoritis dan pemrograman tentang Strategi Brute-Force dan Strategi Rekursif

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

- Mahasiswa mampu menjelaskan tentang konsep strategi brute-force/exhaustive search dan teknik heuristik dengan baik, menganalisis kebenaran dan kompleksitas waktu algoritma brute-force, serta mengaplikasikan strategi tersebut dalam pemecahan masalah dengan baik dan benar.
- Mahasiswa mampu menjelaskan konsep algoritma rekursif, menuliskan pseudocode, menganalisis kebenaran, memformulasikan bentuk rekursif dari fungsi kompleksitas waktunya dan menghitung rumus eksplisit fungsi tersebut, , serta mengaplikasikan metode rekursif dalam pemecahan masalah dan mengimplementasikannya dalam program komputer dengan baik dan benar

DESKRIPSI TUGAS

Aturan pengerjaan tugas:

1. Kerjakan **semua** soal yang ada. Naskah ini terdiri dari 4 soal wajib dan 1 soal bonus.
2. Tugas boleh diketik/ditulis tangan (pastikan bisa dibaca), boleh menggunakan Bahasa Indonesia/Inggris. Hindari menggunakan tinta merah. Soal teori disimpan dalam format pdf. Jika menggunakan tulis tangan, harap discan (tidak difoto), kemudian dikompresi untuk memperkecil ukuran file.
3. Untuk tugas pemrograman (soal nomor 2 (a-g), 3 b, dan 4 (c-e)), buatlah dengan jupyter notebook. Harap mengurutkan program sesuai dengan urutan pada soal, dan berikan keterangan nomor soal pada setiap program. Penjelasan dapat ditulis dengan “markdown” yang tersedia pada jupyter. Untuk setiap program, tambahkan “**annotation**” atau “**comment**” pada bagian yang Anda rasa perlu, untuk dokumentasi/memudahkan pembacaan program Anda.
4. Saya menganjurkan Anda menulis program dalam python dengan jupyter notebook. Namun apabila Anda memilih menggunakan bahasa pemrograman lain (C, C++, atau java), mohon disampaikan ke saya. File pemrograman disimpan dalam 1 folder yang kemudian dikompres menjadi ekstensi zip. File terdiri dari (1) source code; (2) readme file yang menjelaskan bagaimana program Anda dapat dijalankan melalui terminal, serta rangkuman tentang apa yang Anda kerjakan dalam program tersebut; (3) file tambahan yang dibutuhkan untuk mengetes program Anda; (4) hasil eksperimen yang Anda kerjakan (format pdf atau png). Penamaan setiap file harus rapi.

5. Tugas teori dikumpulkan dalam format **pdf**, dan file program disimpan dalam 1 folder yang kemudian dikompres menjadi **zip**, dengan format penamaan tugas: **NamaLengkap_Kelas_NIM.extension**. Contoh: **GedeGanesha_6A_1610101001.pdf** atau **GedeGanesha_6A_1610101001.zip**. Pengumpulan tugas melalui e-learning Undiksha.
6. Anda diizinkan untuk berdiskusi dengan rekan Anda. Namun Anda harus menuliskan/menjelaskan jawaban Anda sendiri, dan paham dengan baik apa yang Anda tulis. Terutama untuk bagian pemrograman, Anda dilarang keras melakukan copy-paste program dari rekan Anda maupun dari internet! Anda siap bertanggung jawab terhadap hasil pekerjaan Anda. Hasil pekerjaan yang memiliki kemiripan yang tinggi dengan pekerjaan mahasiswa lain mempengaruhi poin penilaian.
7. Tugas dinilai berdasarkan kerapian penulisan tugas teori, kerapian dan kejelasan program komputer, dan kejelasan serta kesesuaian jawaban/penjelasan dengan pertanyaan yang diajukan. Total nilai maksimum tugas ini adalah 110. Keterlambatan dalam pengumpulan tugas mengurangi poin penilaian.

Problems

1. (Brute force, 15 poin)

Suatu perusahaan sedang melakukan uji coba untuk menentukan lantai tertinggi dari kantor pusatnya yang berlantai n , yang memungkinkan sebuah gadget dapat jatuh tanpa mengalami kerusakan. Perusahaan tersebut memiliki dua gadget identik untuk bereksperimen. Jika salah satunya rusak, maka gadget tersebut tidak dapat diperbaiki, dan eksperimen harus diselesaikan dengan gadget yang tersisa. (Catatan: kompleksitas waktu dihitung berdasarkan banyaknya percobaan yang dilakukan.)

- (a) (5 poin) Rancang sebuah algoritma dengan pendekatan brute-force untuk masalah ini. Jelaskan algoritmanya, kemudian tulis pseudocode untuk algoritma tersebut, dan tentukan kompleksitas waktunya!
- (b) (10 poin) Modifikasi algoritma tersebut untuk meningkatkan efisiensinya. (Hint: kita bisa membuat algoritma dengan kompleksitas waktu *worst-case* $\mathcal{O}(\sqrt{n})$.)

2. (Have fun with recursion, 20 poin)

Tulis program dalam bahasa python (buat di jupyter notebook) untuk latihan berikut (kerjakan soal secara berurutan). Pastikan untuk memperhatikan kasus dasar (*base case*) dan pemanggilan rekursi (*recursive call*) Anda!

Tambahkan tabel evaluasi sebagai berikut di laporan file pdf Anda.

Nama program	Poin 1	Poin 2	Poin 3	Poin 4	Keterbatasan program
a. Integer multiplication					
b. Powering					
c. Print Down					
d. Print Up					
e. Reverse string					
f. Prime checking					
g. Fibonacci					

Poin penilaian pada tabel (diisi dengan **Ya/Tidak**)

1. Program berhasil dikompilasi tanpa kesalahan (no syntax error)
2. Program berhasil *running*
3. Program dapat membaca file masukan dan menuliskan luaran.
4. Program dapat mengatasi ketika input tidak sesuai dengan kriteria

Spesifikasi program:

- (a) Misal a dan b adalah bilangan bulat tak negatif. Saat di SD, kita diajarkan bahwa nilai $b \times a$ ekuivalen $\underbrace{a + a + \dots + a}_{\text{sebanyak } b}$. Manfaatkan sifat penjumlahan tersebut untuk membuat fungsi yang mengambil input dua bilangan bulat tak negatif dan mengalikannya secara rekursif.
- (b) Buat fungsi yang memberikan input bilangan bulat X dan $n \geq 0$, dan menghitung X^n secara rekursif. Anda tidak diperbolehkan menggunakan operator $**$ (operator pangkat pada python)!
- (c) Buat fungsi menggunakan rekursi untuk mencetak angka dari n ke 0.
- (d) Modifikasi fungsi sebelumnya untuk membuat sebuah fungsi menggunakan rekursi untuk mencetak angka dari 0 hingga n .
- (e) Tulis fungsi rekursif yang mengambil input sebuah string dan memberikan return string dalam urutan terbalik (Contoh: input = "Salam" maka output = "malaS"). Satu-satunya operasi string yang boleh Anda gunakan adalah penggabungan string (atau *concatenation*, dengan menggunakan operasi "+").
- (f) Tulis sebuah fungsi rekursif untuk mengecek apakah suatu bilangan n adalah bilangan prima (Anda harus memeriksa apakah n habis dibagi dengan bilangan di bawah n).
- (g) Tulis fungsi rekursif yang mengambil satu argumen $n \geq 1$ dan menghitung $F(n)$, yakni nilai ke- n dari barisan Fibonacci. Barisan Fibonacci didefinisikan oleh relasi:

$$F(n) = \begin{cases} 1, & n = 1 \\ 1, & n = 2 \\ F_{n-1} + F_{n-2}, & n \geq 3 \end{cases}$$

3. (Permutation, 30 poin)

Permutasi adalah penyusunan kembali suatu kumpulan objek dalam urutan yang berbeda dari urutan yang semula. Misalnya, $A = [1, 2, 3]$ dapat dipermutasikan menjadi $[1, 3, 2]$, $[2, 1, 3]$, $[2, 3, 1]$, $[3, 1, 2]$, atau $[3, 2, 1]$ (6 macam permutasi). Diberikan algoritma permutasi sebagai berikut.

Algorithm 1 Unknown algorithm

```
1: procedure UNKNOWN( $A[0..n-1], n$ )
2:   input: an array  $A = [1, 2, 3, \dots, n]$ , and a positive integer  $n$ 
3:   if  $n = 1$  then
4:     print  $A$ 
5:   else
6:     for  $i \leftarrow 0$  to  $n - 1$  do
7:       UNKNOWN( $A[0..n-1], n - 1$ )
8:       if  $n$  is odd then
9:         swap  $A[0]$  and  $A[n - 1]$ 
10:      else
11:        swap  $A[i]$  and  $A[n - 1]$ 
12:      end if
13:    end for
14:  end if
15: end procedure
```

- (a) (10 poin) Jalankan algoritma tersebut dengan memberikan input $n = 2$ dan A adalah array dengan elemen integer $1, 2, 3, \dots, n$. Tuliskan proses pada setiap iterasinya untuk setiap nilai n tersebut! Lakukan hal yang sama dengan nilai $n = 3$.
- (b) (5 poin) Implementasikan algoritma tersebut dalam program komputer. Lalu tes dengan memberikan input integer n dan sebuah array $A = [1, 2, 3, \dots, n]$. Tes program dengan nilai n yang berbeda. Jelaskan apa yang menjadi output program Anda (atau algoritma di atas)! Selidiki apakah hasil pada soal (a) sama dengan output yang diberikan program Anda!

Lengkapi tabel berikut, dan tulis dalam laporan di file pdf Anda. Tambahkan keterbatasan program setelah tabel.

Poin	Ya	Tidak
Program berhasil dikompilasi tanpa kesalahan (no syntax error)		
Program berhasil <i>running</i>		
Program dapat membaca file masukan dan menuliskan luaran		
Program memberikan output yang sama dengan soal (a)		
Program dapat mengatasi ketika input tidak sesuai dengan kriteria		

Keterbatasan program: _____

- (c) (10 poin) Berdasarkan jawaban pada soal (b), buktikan secara formal (dengan induksi matematika) kebenaran (*correctness*) algoritma tersebut!
- (d) (5 poin) Tentukan kompleksitas waktunya (nyatakan dalam $\mathcal{O}(n)$). Jelaskan jawaban Anda!
4. (Experiment with tower of Hanoi, 35 poin)
- (a) (5 poin) Dalam versi asli “Tower of Hanoi problem”, seperti yang diterbitkan pada tahun 1890-an oleh Édouard Lucas, seorang ahli matematika Prancis, dunia akan berakhir setelah 64 cakram dipindahkan dari Menara Brahma yang mistis. Berdasarkan algoritma yang didiskusikan di kelas, perkiraan berapa tahun yang diperlukan jika para bhikkhu dapat memindahkan satu cakram per menit. (Asumsikan bahwa para bhikkhu dapat bekerja sepanjang waktu, tidak makan, tidur, atau mati.)
- (b) (5 poin) Berapa banyak gerakan yang dilakukan oleh piringan terbesar ke- i ($1 \leq i \leq n$) dalam algoritma ini? Jelaskan jawaban Anda!
- (c) (5 poin) Implementasikan dalam bahasa python algoritma rekursif Anda. Masukan program adalah banyaknya cakram (atau n) dan list nomor cakram pada setiap tower, yaitu “source”, “target”, “helper”. Output program adalah list cakram di ketiga tower “source”, “target”, “helper”.
- (d) (5 poin) Buat program rekursif “Tower of Hanoi”, dengan spesifikasi: diberikan input n yaitu banyaknya cakram, program akan memberikan output berupa langkah-langkah memindahkan semua cakram dari “source” ke “target”. Misalnya untuk $n = 2$, maka outputnya adalah:
- Pindahkan cakram 1 dari source ke helper
 - Pindahkan cakram 2 dari source ke target
 - Pindahkan cakram 1 dari helper ke target
- (e) (5 poin) Untuk masing-masing program, buatlah sebuah eksperimen dengan memberikan input banyaknya cakram $n = 0, 1, 2, \dots, 30$ (jika program Anda berjalan cukup cepat, berikan masukan hingga $n = 50$ atau lebih), dan uji performance-nya dengan mencatatkan *running time* untuk setiap nilai n yang diinput. Visualisasikan *running time* masing-masing program dengan menyajikan grafik garis keduanya dalam satu bingkai, untuk membandingkan performance kedua program tersebut.
- (f) (5 poin) Selidiki apakah grafik *running time* yang ditampilkan sesuai dengan fungsi kompleksitas waktu secara teoritis? Buat analisis dan simpulkan hasil eksperimen Anda!
- (g) (5 poin) Desain sebuah algoritma non-rekursif untuk permasalahan “Tower of Hanoi” (Anda tidak diharuskan untuk menuliskan pseudocodenya).

Untuk memudahkan pengamatan, tambahkan tabel evaluasi sebagai berikut di laporan file pdf Anda. Tuliskan keterbatasan program di bawah tabel.

Poin	Ya	Tidak
Program rekursif pertama berhasil dikompilasi tanpa kesalahan (no syntax error)		
Program rekursif kedua berhasil dikompilasi tanpa kesalahan (no syntax error)		
Program rekursif pertama berhasil <i>running</i>		
Program rekursif kedua berhasil <i>running</i>		
Program rekursif pertama memberikan output yang benar		
Program rekursif kedua memberikan output yang benar		
Program dapat menampilkan grafik perbandingan efisiensi		
Kedua program dapat mengatasi ketika input tidak sesuai dengan kriteria		

Keterbatasan program: Jelaskan batas maksimum nilai n agar program Anda tidak *slow running*, dan hal-hal lainnya

5. (Bonus: Mencari selebgram, 10 poin)

Seorang selebgram di antara sekelompok n orang, adalah dia yang tidak mengenal siapa pun, tetapi dikenal oleh semua orang lainnya. Bagaimanakah cara mengidentifikasi orang yang merupakan selebgram dengan hanya mengajukan pertanyaan kepada orang-orang dalam bentuk “Apakah Anda mengenalnya?” (kata ‘nya’ disini mengacu pada seseorang yang ditunjuk secara acak).

1. Rancang sebuah algoritma yang efisien untuk mengidentifikasi selebgram atau menentukan bahwa suatu grup tidak memiliki orang seperti itu.
2. Berapa banyak pertanyaan yang dibutuhkan algoritma Anda dalam kasus terburuk?

TAHAPAN Pengerjaan Tugas

1. Mengikuti perkuliahan materi *Strategi Brute-Force* dan *Strategi Rekursif*
2. Berdiskusi, mencari sumber pembelajaran di internet, dan mengerjakan tugas

Bentuk dan Format Luaran

Jawaban dikumpulkan dalam format pdf, source code dikumpulkan dalam file berekstensi zip

Indikator, Kriteria dan Bobot Penilaian

Penilaian dengan angka skor (0-100)

Kriteria penilaian:

1. Ketepatan jawaban dan keruntutan langkah penyelesaian soal (50%)
2. Penulisan laporan (25%)
3. Penulisan source code (25%)
4. Kejujuran/integritas dan ketepatan waktu: penambahan/pengurangan skor

Jadwal Pelaksanaan

Waktu

Pemberian tugas dan materi terkait

Pertemuan 4

Pengumpulan tugas

Pertemuan 6

Lain-lain

Kebenaran dan kejelasan penyelesaian tugas, kejelasan source-code, integritas dalam pengerjaan tugas, ketepatan waktu penyelesaian tugas

Daftar Rujukan

- Materi perkuliahan pertemuan 2
- Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc



Institusi : Universitas Pendidikan Ganesha
Fakultas : Teknik dan Kejuruan
Prodi : Ilmu Komputer

Rencana Tugas Mahasiswa (RTM)

MATA KULIAH	Desain dan Analisis Algoritma	SKS:	3	SEMESTER:	VI
KODE MATA KULIAH	KOMS119601				
DOSEN PENGAMPU	Ni Luh Dewi Sintuari, Ph.D.				
TUGAS KE	4	MINGGU KE:		10 - 14	
BENTUK TUGAS	Aktivitas kelompok				

JUDUL TUGAS

Tugas presentasi kelompok

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Sub-CPMK6 sampai dengan Sub-CPMK10

DESKRIPSI TUGAS

Tugas dikerjakan secara berkelompok (3 atau 4 orang).

Aturan pengerjaan tugas:

1. Buat presentasi singkat (ppt/tulis tangan). Setiap anggota kelompok menjelaskan secara bergantian.
2. Waktu presentasi ± 20 menit

Kelompok 1 dan 2:

Topik presentasi

1. KNAPSACK PROBLEM

- Jelaskan tiga strategi penyelesaian masalah Integer Knapsack problem Greedy by profit, Greedy by weight, dan Greedy by density, dengan menggunakan contoh.
- Tuliskan pseudocode algoritma Greedy (ketiga strategi) untuk Integer knapsack problem.
- Jelaskan definisi "Fractional knapsack problem", dan strategi penyelesaiannya dengan algoritma Greedy.
- Tuliskan pseudocode algoritma Greedy untuk Fractional knapsack problem.
- Jelaskan mengapa algoritma Greedy untuk Fractional knapsack problem memberikan solusi optimal!

2. KODE HUFFMAN

- Jelaskan definisi encoding/decoding (atau kompresi/dekompresi), fixed-length encoding, dan variable-length encoding
- Jelaskan formulasi masalah dan prinsip kode Huffman (*Huffman coding*)
- Jelaskan algoritma dan tuliskan pseudocode algoritma kode Huffman
- Berikan contoh implementasi algoritma kode Huffman, beserta visualisasi permasalahannya
- Jelaskan keunggulan dan pemanfaatan kode Huffman

Kelompok 3, 4, 5, dan 6

1. Greedy Algorithm for Minimum Spanning Tree

- Jelaskan prinsip algoritma Greedy untuk Minimum Spanning Tree problem
- Berikan contoh implementasi algoritma dengan menggunakan graf (dengan 8-10 titik dan ≥ 15 sisi). Jelaskan prosedur pada setiap langkah/iterasi-nya dengan menggunakan gambar.

2. Kruskal's Algorithm

- Jelaskan prinsip algoritma Kruskal untuk Minimum Spanning Tree problem
- Berikan contoh implementasi algoritma dengan menggunakan graf (dengan 8-10 titik dan ≥ 15 sisi). Jelaskan prosedur pada setiap langkah/iterasi-nya dengan menggunakan gambar.

3. Prim's Algorithm

- Jelaskan prinsip algoritma Prim untuk Minimum Spanning Tree problem
- Berikan contoh implementasi algoritma dengan menggunakan graf (dengan 8-10 titik dan ≥ 15 sisi). Jelaskan prosedur pada setiap langkah/iterasi-nya dengan menggunakan gambar.

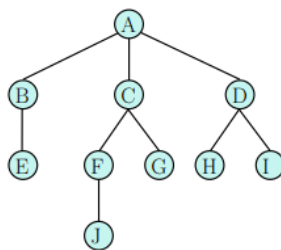
4. Dijkstra Algorithm

- Jelaskan prinsip algoritma Dijkstra untuk Shortest Path problem (pada graf berarah/graf tidak berarah, pilih salah satu)
- Berikan contoh implementasi algoritma dengan menggunakan graf (dengan 8-10 titik dan ≥ 15 sisi). Jelaskan prosedur pada setiap langkah/iterasi-nya dengan menggunakan gambar.

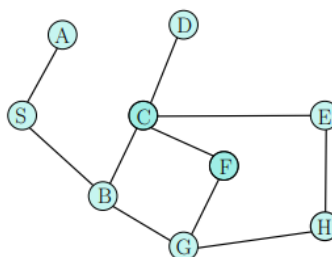
Kelompok 7, 8 dan 9

1. Algoritma DFS

- Jelaskan prinsip dasar algoritma DFS.
- Jelaskan secara singkat dan jelas ide dari algoritma DFS beserta pseudocode DFS (pseudocode versi metode rekursif).
- Ilustrasikan DFS pada graf pohon berikut, lalu jelaskan dengan gambar bagaimana algoritma DFS pada tree tersebut (dimulai dari titik A).



- Ilustrasikan DFS pada graf yang bukan tree: jelaskan dengan gambar bagaimana algoritma DFS pada graf berikut (dimulai dari titik S).



- Jelaskan pembentukan *pohon DFS* (atau *DFS tree*) berdasarkan pada contoh DFS pada graf di atas. Pohon DFS adalah graf pohon (tree) yang menggambarkan urutan simpul yang dikunjungi.

2. Algoritma BFS

Soal yang sama seperti di atas, tapi pada BFS.

3. Membangun state-space tree (pohon ruang status)

- Jelaskan tentang definisi pohon ruang status (state-space tree), komponennya (seperti root/akar, leaves/daun, branch/cabang), ruang status (state space), dan ruang solusi (solution space).
- Jelaskan tentang pembentukan pohon ruang status, ambil kasus berikut.

Problem 1. Diberikan himpunan dengan 3 elemen yaitu A, B, dan C. Kita ingin membuat list permutasi dari ketiga elemen tersebut, yaitu: ABC, ACB, BAC, BCA, CAB, CBA dengan menggunakan operasi 'addition', yakni dimulai dari himpunan kosong \emptyset , pada setiap langkah, sebuah elemen ditambahkan. Jelaskan bagaimana membangun pohon ruang status untuk permasalahan ini.

- Jelaskan perbedaan cara membangun pohon ruang status dengan metode DFS dan BFS.
- Baca tentang permainan 8-puzzle. Jelaskan pembangunan pohon ruang status dengan metode DFS dan BFS hingga solusi ditemukan.

2	8	3
1	6	4
7		5

initial state

1	2	3
8		4
7	6	5

goal state

Figure 1: State awal 8-puzzle dan state akhirnya (setelah puzzle diselesaikan/permainan berakhir)

Kelompok 10, 11, dan 12

1. Prinsip dasar algoritma backtracking

- Jelaskan definisi dan alur kerja metode backtracking.
- Jelaskan prinsip dasar pembentukan pohon ruang status pada metode backtracking, dan bagaimana peran metode DFS (*Depth First Search*) dalam proses ini.
- Jelaskan komponen apa saja yang membentuk pohon ruang status untuk metode backtracking (seperti *root* (akar), *internal nodes* (simpul dalam), dan *leaves* (simpul daun)).
- Baca tentang permasalahan n -ratu (n -queens problem). Jelaskan bagaimana solusi masalah ini untuk nilai $n = 1, 2, 3$, kemudian jelaskan secara runut bagaimana metode backtracking dapat diaplikasikan untuk menyelesaikan masalah ini untuk nilai $n > 4$. Untuk menjelaskan algoritma ini, Anda bisa mengambil kasus $n = 4$, dan ilustrasikan prosedurnya dengan membangun pohon ruang statusnya.

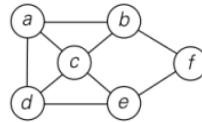
2. Penerapan algoritma backtracking

- Masalah sirkuit Hamiltonian didefinisikan sebagai berikut:

Problem 1. Diberikan sebuah graf tidak berarah (dan tidak berbobot). Kita ingin mencari sirkuit yang melintasi setiap titik pada graf tepat satu kali.

Perhatikan bahwa masalah sirkuit Hamilton dapat dipandang sebagai masalah Traveling Salesman Problem dimana graf input adalah graf yang semua sisinya diberikan bobot 1.

- Diberikan graf berikut:



Implementasikan metode backtracking untuk mencari sirkuit Hamilton pada graf tersebut. Anda bisa memilih sebarang titik sebagai *starting vertex* (misalnya titik *a*). Gambarkan pohon ruang status sesuai dengan prosedur yang Anda jalankan.

- Masalah subset sum atau *subset-sum problem* didefinisikan sebagai berikut.

Problem 2. Diberikan himpunan integer positif $\{a_1, a_2, \dots, a_n\}$ dan integer d . Kita ingin mencari sebuah sub-himpunan yang jumlah semua elemennya adalah d .

Sebagai contoh, jika inputnya adalah $\{1, 2, 4, 6, 8\}$, maka solusinya adalah $\{1, 2, 6\}$ atau $\{1, 8\}$.

Soal: diberikan himpunan input $\{3, 5, 6, 7\}$ dan nilai $d = 15$. Implementasikan metode backtracking untuk mendapatkan solusi subset-sum. Gambarkan pohon ruang statusnya.

3. Metode branch-and-bound

- Jelaskan secara singkat prinsip dasar metode branch-and-bound, dan perbedaannya dengan metode backtracking.
- Jelaskan prosedur pembangunan pohon ruang status pada metode branch-and-bound.
- Metode branch-and-bound dapat digunakan untuk menyelesaikan masalah Integer Knapsack. Pertama-tama, objek diurutkan dengan urutan menurun (non-ascending) berdasarkan nilai $\frac{v_i}{w_i}$. Selanjutnya dibentuk pohon ruang status seperti pada masalah penugasan di atas. Batas atas (upper bound) dihitung berdasarkan total biaya pada masing-masing percabangan ditambah dengan total profit dari objek yg dipilih, sesuai dengan rumus berikut:

$$ub = v + (W - w) \cdot \frac{v_{i+1}}{w_{i+1}}$$

dimana v dan w adalah total profit dan total bobot pada objek yang sudah dipilih sampai dengan tahap ke- i , dan W adalah kapasitas knapsack.

Diberikan sebuah instance untuk permasalahan Integer Knapsack sebagai berikut. Aplikasikan metode branch-and-bound untuk menyelesaikan masalah tersebut. Gambar pohon ruang statusnya.

item	weight	value	$\frac{\text{value}}{\text{weight}}$
1	4	\$40	10
2	7	\$42	6
3	5	\$25	5
4	3	\$12	4

The knapsack's capacity W is 10

- Berikan analisis Anda terkait dengan kelebihan serta kekurangan metode branch-and-bound untuk menyelesaikan masalah Integer Knapsack jika dibandingkan metode lain yang sudah dibahas (seperti brute force atau greedy)

TAHAPAN Pengerjaan Tugas	
1. Membaca materi terkait dengan topik presentasi pada buku rujukan yang diberikan. 2. Berdiskusi dengan kelompok, menyiapkan file presentasi, dan mempresentasikan topik tersebut.	
Bentuk dan Format Luaran	
File presentasi dalam format pdf, dan presentasi serta tanya jawab saat pelaksanaan perkuliahan.	
Indikator, Kriteria dan Bobot Penilaian	
Penilaian dengan angka skor (0-100) Kriteria penilaian: 1. Kesesuaian materi presentasi dengan permasalahan yang diberikan (50%) 2. Kejelasan presentasi setiap anggota kelompok (25%) 3. Kemampuan menjawab pertanyaan, menanggapi pertanyaan dari mahasiswa lain (25%) 4. Kerjasama kelompok: tambahan skor	
Jadwal Pelaksanaan	Waktu
Pemberian tugas dan materi terkait	Setiap minggu sebelum pertemuan
Pengumpulan tugas	Setiap minggu berikutnya
Lain-lain	
Kriteria penilaian sesuai dengan deskripsi pada lembar penugasan. Penilaian secara umum meliputi kebenaran dan kejelasan penyelesaian tugas, integritas dalam pengerjaan tugas, ketepatan waktu penyelesaian tugas, dan kontribusi mahasiswa dalam penyelesaian tugas.	
Daftar Rujukan	
<ul style="list-style-type: none"> • Materi perkuliahan seluruh pertemuan • Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc • Sumber internet sesuai dengan permasalahan/topik yang dibahas 	



Institusi : Universitas Pendidikan Ganesha
Fakultas : Teknik dan Kejuruan
Prodi : Ilmu Komputer

Rencana Tugas Mahasiswa (RTM)

MATA KULIAH	Desain dan Analisis Algoritma	SKS:	3	SEMESTER:	VI
KODE MATA KULIAH	KOMS119601				
DOSEN PENGAMPU	Ni Luh Dewi Sintiar, Ph.D.				
TUGAS KE	5	MINGGU KE:		10	
BENTUK TUGAS	Tugas Mandiri				

JUDUL TUGAS

Tugas tertulis penerapan dan analisis strategi Greedy

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu menjelaskan konsep algoritma Greedy, membuktikan optimalitas atau menunjukkan ketak-optimalan algoritma Greedy, mengaplikasikan metode Greedy dalam pemecahan masalah dan mengimplementasikannya dalam program komputer dengan baik dan benar.

DESKRIPSI TUGAS

Aturan pengerjaan tugas:

1. Kerjakan **semua** soal yang ada secara singkat, padat, dan jelas.
2. Tugas **ditulis tangan** (pastikan bisa dibaca), boleh menggunakan Bahasa Indonesia/Inggris. Hindari menggunakan tinta merah, discan (tidak difoto), kemudian dikompresi untuk memperkecil ukuran file. Tulis jawaban pada satu file pdf
3. Format penamaan tugas: **NamaLengkap_Kelas.NIM**. Contoh: **GedeGanesha_6A_1610101001.pdf**. Pengumpulan tugas melalui e-learning Undiksha.
4. Anda diizinkan untuk berdiskusi dengan rekan Anda. Namun Anda harus menuliskan/menjelaskan jawaban Anda sendiri, dan paham dengan baik apa yang Anda tulis. Anda siap bertanggung jawab terhadap hasil pekerjaan Anda. Hasil pekerjaan yang memiliki kemiripan yang tinggi dengan pekerjaan mahasiswa lain mempengaruhi poin penilaian.
5. Tugas dinilai berdasarkan kerapian penulisan, dan kejelasan serta kesesuaian jawaban/penjelasan dengan pertanyaan yang diajukan. Keterlambatan dalam pengumpulan tugas mengurangi poin penilaian.

1 Job scheduling with deadlines

Given n jobs that will be done by a machine. Each job is processed by the machine in one unit time and deadline of every job i is $d_i \geq 0$. Job i will give profit p_i if and only if the job is done not after the deadline. How to choose the jobs that will be processed by the machine so that the profit is maximum?

1. Given 4 jobs ($n = 4$) with the following characteristics:

- $(p_1, p_2, p_3, p_4) = (50, 10, 15, 30)$
- $(d_1, d_2, d_3, d_4) = (2, 1, 2, 1)$

Let the machine starts to work at 6 am, then we have the following constraints:

Job	Deadline (d_i)	Must be done before
1	2 hours	8 am
2	1 hour	7 am
3	2 hours	8 am
4	1 hour	7 am

Let J be the set of jobs, then the objective function of this problem is:

$$\text{Maximize } F = \sum_{i \in J} p_i$$

- A solution set J is *feasible* if every job in J is done before the deadline.
- An *optimal solution* is a feasible solution that maximize F .

TASK: Solve the problem with exhaustive search. For this, create a table containing, set of solutions, order of processing, total profit, and feasibility.

- *Set of solution:* possible subset of chosen jobs
- *Order of processing:* the order of job processed so that the deadline is respected.
- *Total profit:* total of profit based on the set of feasible solutions
- *Feasibility:* whether the set of solution is feasible/not feasible

Set of jobs	Order	Total profit (F)	Description
{ }	-	0	feasible
{1}	1	50	feasible
\vdots	\vdots	\vdots	\vdots

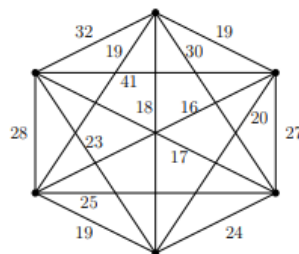
2. What is the complexity of the exhaustive search approach?
3. Consider the following Greedy strategy for the problem. To choose a job: “at each step, choose job i with the largest p_i to increase the objective value F ”. Solve the problem with the Greedy strategy and find the *optimal solution* and the *optimal profit*. (The optimal solution is a solution set that gives the optimal profit.)
4. Write a pseudocode to implement the Greedy strategy above!
5. What is the complexity of your Greedy algorithm?
6. Explain how to improve the Greedy algorithm above?

2 Traveling salesman problem

TSP: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?

Let the vertices of the input graph G be: v_1, v_2, \dots, v_n . Let the tour is started from v_1 . The next vertex is chosen “greedily”: At each step i , choose the vertex v_j (among the available vertices) for which the weight of edge (v_i, v_j) is minimized.

Given the following graph. Implement the Greedy algorithm to find a TSP solution for this graph. Write the TSP solution and compute the total weight of the solution.



3 Integer Knapsack problem

Recall that in the Integer (1/0) Knapsack Problem, there are three greedy strategies that can be implemented to obtain a solution for the knapsack problem: Greedy by profit, Greedy by weight, and Greedy by density. However, it is claimed that none of the strategies guaranteed an optimal solution. Give an instance of the Integer Knapsack Problem, where the optimal solution is different than the solution obtained by implementing the three Greedy strategies (by profit, by weight, and by density). For this question, you should give an instance, the solution of the three strategies (as explained in the lecture), and the optimal solution (give a proof/an argument to show that it is indeed an optimal solution). You are not allowed to give the same example as your fellow.

TAHAPAN Pengerjaan Tugas

1. Mengikuti perkuliahan materi *Strategi Greedy* (pertemuan 9 dan 10).
2. Membaca catatan perkuliahan dan buku rujukan untuk menyelesaikan permasalahan yang diberikan.

BENTUK DAN FORMAT LUARAN

Tugas tertulis dalam format pdf

INDIKATOR, KRITERIA DAN BOBOT PENILAIAN

Penilaian dengan angka skor (0-100)

Kriteria penilaian:

1. Ketepatan jawaban (30%)
2. Keruntutan langkah penyelesaian soal (70%)
3. Kejujuran/integritas dan ketepatan waktu: penambahan/pengurangan skor

JADWAL PELAKSANAAN

WAKTU

Pemberian tugas dan materi terkait

Pertemuan 9

Pengumpulan tugas

Pertemuan 11

LAIN-LAIN

Kriteria penilaian sesuai dengan deskripsi pada lembar penugasan. Penilaian secara umum meliputi kebenaran dan kejelasan penyelesaian tugas, integritas dalam pengerjaan tugas, ketepatan waktu penyelesaian tugas, dan kontribusi mahasiswa dalam penyelesaian tugas.

DAFTAR RUJUKAN

- Materi perkuliahan pertemuan 9 dan 10 tentang Strategi Greedy
- Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc



Institusi : Universitas Pendidikan Ganesha
Fakultas : Teknik dan Kejuruan
Prodi : Ilmu Komputer

Rencana Tugas Mahasiswa (RTM)

MATA KULIAH	Desain dan Analisis Algoritma	SKS: 3	SEMESTER: VI
KODE MATA KULIAH	KOMS119601		
DOSEN PENGAMPU	Ni Luh Dewi Sintuari, Ph.D.		
TUGAS KE	6	MINGGU KE:	11
BENTUK TUGAS	Tugas Mandiri		

JUDUL TUGAS

Tugas tertulis penerapan algoritma Greedy berbasis Cut property dan algoritma Prim untuk mencari Minimum Spanning Tree

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu menjelaskan konsep algoritma Greedy, membuktikan optimalitas atau menunjukkan ketak-optimalan algoritma Greedy, mengaplikasikan metode Greedy dalam pemecahan masalah dan mengimplementasikannya dalam program komputer dengan baik dan benar.

DESKRIPSI TUGAS

Aturan pengerjaan tugas

1. Tugas dikerjakan secara individu. Graf yang Anda buat tidak boleh sama/memiliki kemiripan yang tinggi dengan mahasiswa lain. Anda juga tidak diperbolehkan membuat graf yang sama, dan hanya memberikan bobot sisi yang berbeda.
2. Tugas ditulis tangan dengan jelas. Hindari penggunaan tinta berwarna merah.
3. Pada lembar tugas, gambar graf yang Anda buat, kemudian solusi dari hasil implementasi algoritma Prim, dan solusi dari hasil implementasi algoritma Greedy cut property.
4. Buatlah sebuah video yang memperlihatkan bagaimana Anda mengimplementasikan algoritma Prim dan algoritma Greedy pada graf tersebut. Video yang dibuat berdurasi ± 5 menit.
5. Pada video, ditampilkan proses penyelesaian soal secara jelas. Wajah Anda saat menjelaskan kedua algoritma harus terlihat jelas pada video. Pastikan juga suara Anda terdengar dengan jelas.
6. Unggah video yang Anda buat di youtube/gdrive/media lainnya, kemudian link video dikirim melalui E-learning, beserta dengan lembar jawaban tulis tangan pada poin nomor 3 di atas. Pastikan link yang Anda kirimkan dapat dibuka.

Rincian tugas

1. Buatlah sebuah graf dengan 10-12 verteks dan 15-20 sisi. Berikan bobot pada sisi-sisi graf berupa integer positif, dimana setiap sisi memiliki bobot yang berbeda.
2. Implementasikan algoritma Prim untuk menemukan minimum spanning tree pada graf yang Anda buat.
3. Implementasikan algoritma Greedy berbasis *Cut Property* untuk menemukan minimum spanning tree pada graf yang Anda buat. Berikut dicantumkan lagi Lemma "Cut property".

Lemma -1.1 (Cut property). *Diberikan sebarang cut pada graf, sisi dengan bobot minimum pada cut tersebut merupakan sisi dari minimum spanning tree.*

TAHAPAN Pengerjaan Tugas	
1. Mengikuti perkuliahan materi <i>Strategi Greedy untuk Menyelesaikan Masalah pada Graf</i> (pertemuan 11). 2. Membaca catatan perkuliahan dan buku rujukan untuk menyelesaikan permasalahan yang diberikan. 3. Menuliskan jawaban secara runut, dan menjelaskannya dalam bentuk video presentasi	
Bentuk dan Format Luaran	
Tugas tertulis dalam format pdf	
Indikator, Kriteria dan Bobot Penilaian	
Penilaian dengan angka skor (0-100) Kriteria penilaian: 1. Ketepatan jawaban (20%) 2. Keruntutan langkah penyelesaian soal (40%) 3. Kejelasan presentasi (40%) 4. Kejujuran/integritas dan ketepatan waktu: penambahan/pengurangan skor	
Jadwal Pelaksanaan	Waktu
Pemberian tugas dan materi terkait	Pertemuan 11
Pengumpulan tugas	Pertemuan 12
Lain-lain	
Kriteria penilaian sesuai dengan deskripsi pada lembar penugasan. Penilaian secara umum meliputi kebenaran dan kejelasan penyelesaian tugas, integritas dalam pengerjaan tugas, ketepatan waktu penyelesaian tugas, dan kontribusi mahasiswa dalam penyelesaian tugas.	
Daftar Rujukan	
<ul style="list-style-type: none"> Materi perkuliahan pertemuan 11 tentang Penerapan Strategi Greedy untuk Menyelesaikan Masalah pada Graf Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc 	



Institusi : Universitas Pendidikan Ganesha
Fakultas : Teknik dan Kejuruan
Prodi : Ilmu Komputer

Rencana Tugas Mahasiswa (RTM)

MATA KULIAH	Desain dan Analisis Algoritma	SKS:	3	SEMESTER:	VI
KODE MATA KULIAH	KOMS119601				
DOSEN PENGAMPU	Ni Luh Dewi Sintiar, Ph.D.				
TUGAS KE	7	MINGGU KE:		11	
BENTUK TUGAS	Tugas Kelompok				

JUDUL TUGAS

Tugas tertulis dan pemrograman tentang Implementasi Algoritma Kruskal, Prim, dan Dijkstra

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu menjelaskan konsep algoritma Greedy, membuktikan optimalitas atau menunjukkan ketak-optimalan algoritma Greedy, mengaplikasikan metode Greedy dalam pemecahan masalah dan mengimplementasikannya dalam program komputer dengan baik dan benar.

DESKRIPSI TUGAS

Aturan pengerjaan tugas

1. Tugas dikerjakan secara berkelompok.
2. Dilarang melakukan copy-paste program dari sumber manapun, termasuk dari rekan Anda.
3. Untuk kelompok yang mendapatkan program yang sama, wajib memberikan contoh yang berbeda dengan kelompok lainnya.
4. Laporan dibuat dalam Bahasa Indonesia, diketik dengan rapi, dengan spesifikasi: font Times New Roman size 12, spasi 1.25, margin 2.5cm (top/bottom/right) dan 3.0cm (left).
5. Laporan dikumpulkan dalam format pdf dan source file dikumpulkan dalam format zip (berserta dengan semua file yang dibutuhkan dalam eksperimen).

Rincian tugas

1. Pilihlah salah satu topik berikut. Setiap kelompok mengerjakan satu topik, dan setiap topik harus dikerjakan secara merata. Misal jika ada 11 kelompok, maka setiap topik dikerjakan oleh maksimal 4 kelompok.
 - Algoritma Kruskal
 - Algoritma Prim
 - Algoritma Dijkstra
2. Buatlah sebuah program yang mengimplementasikan algoritma tersebut.
3. Berikan 10 instance yang berbeda sebagai input dari program yang Anda buat dengan menggunakan graf dengan 10-20 titik secara bervariasi.
4. Di antara 10 instance tersebut, pilihlah dua instance, dan implementasikan algoritma secara manual. Cek apakah output program Anda untuk kedua instance tersebut sama dengan hasil penghitungan Anda secara manual
5. Lakukan eksperimen pada graf yang Anda buat, dan catat running time untuk masing-masing percobaan. Lakukan analisis untuk mengecek apakah running time dari eksperimen Anda sesuai dengan kompleksitas waktu secara teoritis dari algoritma yang Anda gunakan?
6. Buat sebuah laporan yang berisi:
 - (a) Spesifikasi dari program Anda: petunjuk teknis secara singkat tentang bagaimana program tersebut dijalankan, serta kelebihan dan kekurangan/keterbatasan program.
 - (b) Screen capture source code program.
 - (c) Deskripsi instance dan output dari eksperimen yang dilakukan. Cantumkan hasil screen capture dari setiap percobaan yang dilakukan.
 - (d) Implementasi manual dari dua instance beserta hasil screen capture hasil program untuk instance tersebut.
 - (e) Analisis kompleksitas waktu.
 - (f) Hal-hal lain yang Anda pandang perlu.
7. Pada bagian akhir laporan, tambahkan assesmen individu, berupa persentase kontribusi setiap anggota pada kelompok tersebut. Misal 100% dibagi menjadi 40% anggota pertama, 30% anggota kedua, dan 30% anggota ketiga.

TAHAPAN Pengerjaan Tugas

1. Mengikuti perkuliahan materi *Strategi Greedy untuk Menyelesaikan Masalah pada Graf*.
2. Membaca catatan perkuliahan dan buku rujukan untuk menyelesaikan permasalahan yang diberikan.
3. Berdiskusi dengan anggota kelompok, untuk selanjutnya membuat program sesuai dengan topik yang dibahas berdasarkan rincian penugasan.
4. Secara berkelompok menulis laporan sesuai dengan hasil diskusi dan pengerjaan tugas.
5. Mengumpulkan laporan dalam format pdf diwakilkan oleh seorang anggota kelompok.

Bentuk dan Format Luaran

Laporan dalam format pdf dan source code dalam ekstensi .zip.

Indikator, Kriteria dan Bobot Penilaian

Penilaian dengan angka skor (0-100)

Kriteria penilaian:

1. Penulisan laporan (30%)
2. Penulisan source code (35%)
3. Ketepatan jawaban dan keruntutan langkah penyelesaian soal (35%)
4. Kejujuran/integritas dan ketepatan waktu: penambahan/pengurangan skor

JADWAL PELAKSANAAN		WAKTU
Pemberian tugas dan materi terkait		Pertemuan 11
Pengumpulan tugas		Pertemuan 13
LAIN-LAIN		
Kriteria penilaian sesuai dengan deskripsi pada lembar penugasan. Penilaian secara umum meliputi kebenaran dan kejelasan penyelesaian tugas, integritas dalam pengerjaan tugas, ketepatan waktu penyelesaian tugas, dan kontribusi mahasiswa dalam penyelesaian tugas.		
DAFTAR RUJUKAN		
<ul style="list-style-type: none">• Materi perkuliahan pertemuan 11 tentang Penerapan Strategi Greedy untuk Menyelesaikan Masalah pada Graf• Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc		



Institusi : Universitas Pendidikan Ganesha
Fakultas : Teknik dan Kejuruan
Prodi : Ilmu Komputer

Rencana Tugas Mahasiswa (RTM)

MATA KULIAH	Desain dan Analisis Algoritma	SKS:	3	SEMESTER:	VI
KODE MATA KULIAH	KOMS119601				
DOSEN PENGAMPU	Ni Luh Dewi Sintiar, Ph.D.				
TUGAS KE	7	MINGGU KE:		13	
BENTUK TUGAS	Tugas Kelompok				

JUDUL TUGAS

Tugas tertulis tentang Penerapan Strategi Backtracking dan strategi Branch-and-Bound

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu menjelaskan konsep algoritma Backtracking dan Branch-and-Bound, serta mengaplikasikannya dalam pemecahan masalah algoritmik dengan baik dan benar.

DESKRIPSI TUGAS

Aturan pengerjaan tugas:

1. Kerjakan **semua** soal yang ada secara singkat, padat, dan jelas.
2. Tugas **ditulis tangan** (pastikan bisa dibaca), boleh menggunakan Bahasa Indonesia/Inggris. Hindari menggunakan tinta merah, discan (tidak difoto), kemudian dikompresi untuk memperkecil ukuran file. Tulis jawaban pada satu file pdf
3. Format penamaan tugas: **NamaLengkap_Kelas_NIM**. Contoh: **GedeGanesha_6A_1610101001.pdf**. Pengumpulan tugas melalui e-learning Undiksha.
4. Anda diizinkan untuk berdiskusi dengan rekan Anda. Namun Anda harus menuliskan/menjelaskan jawaban Anda sendiri, dan paham dengan baik apa yang Anda tulis. Anda siap bertanggung jawab terhadap hasil pekerjaan Anda. Hasil pekerjaan yang memiliki kemiripan yang tinggi dengan pekerjaan mahasiswa lain mempengaruhi poin penilaian.
5. Tugas dinilai berdasarkan kerapian penulisan, dan kejelasan serta kesesuaian jawaban/penjelasan dengan pertanyaan yang diajukan. Keterlambatan dalam pengumpulan tugas mengurangi poin penilaian.

Soal

Untuk soal di bawah ini, setiap instance yang digunakan **harus berbeda** dengan mahasiswa lain.

1. Buatlah sebuah instance berupa graf dengan 6-8 titik, dan aplikasikan algoritma Backtracking untuk menemukan **sebuah** sirkuit Hamilton di dalamnya. Gambarkan pohon ruang statusnya, dengan memberikan nomor untuk menunjukkan urutan node yang dibangkitkan.
2. Buatlah sebuah instance untuk Integer Knapsack Problem dengan 4-6 item, dan aplikasikan algoritma Branch-and-Bound untuk menyelesaikan permasalahan tersebut dengan menggunakan fungsi pembatas seperti yang dibahas di kelas untuk menentukan *upper bound* pada setiap node-nya. Gambarkan pohon ruang statusnya, dengan memberikan nomor untuk menunjukkan urutan node yang dibangkitkan.
3. Buatlah sebuah instance untuk Assignment Problem dengan 4 job dan 4 staff, dan aplikasikan algoritma Branch-and-Bound untuk menyelesaikan permasalahan tersebut dengan menggunakan fungsi pembatas seperti yang tertera pada slide pembelajaran (textbook Levitin), untuk menentukan *lower bound* pada setiap node-nya. Gambarkan pohon ruang statusnya, dengan memberikan nomor untuk menunjukkan urutan node yang dibangkitkan.

TAHAPAN Pengerjaan Tugas

1. Mengikuti perkuliahan materi *Analisis Kompleksitas Waktu*.
2. Mengikuti quiz pada pertemuan berikutnya.

Bentuk dan Format Luaran

Jawaban isian pada formular Google Form

Indikator, Kriteria dan Bobot Penilaian

Penilaian dengan angka skor (0-100)

Kriteria penilaian:

1. Ketepatan jawaban (30%)
2. Keruntutan langkah penyelesaian soal (70%)
3. Kejujuran/integritas dan ketepatan waktu: penambahan/pengurangan skor

JADWAL PELAKSANAAN

WAKTU

Pemberian tugas dan materi terkait

Pertemuan 13

Pengumpulan tugas

Pertemuan 14

LAIN-LAIN

Kriteria penilaian sesuai dengan deskripsi pada lembar penugasan. Penilaian secara umum meliputi kebenaran dan kejelasan penyelesaian tugas, integritas dalam pengerjaan tugas, ketepatan waktu penyelesaian tugas, dan kontribusi mahasiswa dalam penyelesaian tugas.

DAFTAR RUJUKAN

- Materi perkuliahan pertemuan 14 tentang Strategi Backtracking dan Strategi Branch-and-Bound
- Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc



Institusi : Universitas Pendidikan Ganesha
Fakultas : Teknik dan Kejuruan
Prodi : Ilmu Komputer

Rencana Tugas Mahasiswa (RTM)

MATA KULIAH	Desain dan Analisis Algoritma	SKS: 3	SEMESTER: VI
KODE MATA KULIAH	KOMS119601		
DOSEN PENGAMPU	Ni Luh Dewi Sintuari, Ph.D.		
TUGAS KE	Quiz 1	MINGGU KE:	3
BENTUK TUGAS	Tugas Mandiri		

JUDUL TUGAS

Quiz tentang Analisis Kompleksitas Waktu

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu menghitung kompleksitas waktu algoritma (worst-case, best-case, average-case), menggunakan notasi Big-O, Big-Omega, dan Big-Theta, dan mengklasifikasikan algoritma berdasarkan kompleksitas waktunya dengan benar.

DESKRIPSI TUGAS

Quiz diberikan dalam bentuk pengisian Google Form pada pertemuan perkuliahan. Hasil pengerjaan Quiz dikumpulkan melalui Google Form.

Soal nomor 1

Which growth that matches each function?

	Constant	Linear	Polynomial	Exponential
$(3/2)n$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$(3/2)^n$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$2n^3$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1000	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$3n$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2^n	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
$3n^2$	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Soal nomor 2

*** For the functions $\log_2 n$ and $\log_8 n$, what is the asymptotic relationship between these functions?

☐ $\log_2 n$ is in $O(\log_8 n)$

☐ $\log_2 n$ is in $\Omega(\log_8 n)$

☐ $\log_2 n$ is in $\Theta(\log_8 n)$

Add an explanation for question ***

Long answer text

TAHAPAN Pengerjaan Tugas

1. Quiz dilaksanakan sebelum pembelajaran pada pertemuan 3 dimulai. Quiz bertujuan untuk mengecek pemahaman mahasiswa terkait dengan materi yang didiskusikan pada pertemuan sebelumnya.
2. Mahasiswa mengakses link Google Form yang diberikan, mengerjakan quiz, dan mengumpulkan melalui Google Form tersebut.

Bentuk dan Format Luaran

Jawaban tertulis pada Google Form

Indikator, Kriteria dan Bobot Penilaian

Penilaian dengan angka skor (0-100), dengan kriteria penilaian:

1. Ketepatan jawaban (100%)

JADWAL PELAKSANAAN	WAKTU
Pemberian tugas dan materi terkait	Pertemuan 6
Pengumpulan tugas	Pertemuan 6

LAIN-LAIN

Kriteria penilaian sesuai dengan deskripsi pada lembar penugasan. Penilaian secara umum meliputi kebenaran dan kejelasan penyelesaian tugas, integritas dalam pengerjaan tugas, ketepatan waktu penyelesaian tugas.

DAFTAR RUJUKAN

- Materi perkuliahan pertemuan 2 tentang Analisis Kompleksitas Waktu
- Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc



Institusi : Universitas Pendidikan Ganesha
Fakultas : Teknik dan Kejuruan
Prodi : Ilmu Komputer

Rencana Tugas Mahasiswa (RTM)

MATA KULIAH	Desain dan Analisis Algoritma	SKS: 3	SEMESTER: VI
KODE MATA KULIAH	KOMS119601		
DOSEN PENGAMPU	Ni Luh Dewi Sintuari, Ph.D.		
TUGAS KE	Quiz 2	MINGGU KE:	6
BENTUK TUGAS	Tugas Mandiri		

JUDUL TUGAS

Quiz tentang Strategi Brute-Force dan Strategi Rekursif

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

- Mahasiswa mampu menjelaskan tentang konsep strategi brute-force/exhaustive search dan teknik heuristik dengan baik, menganalisis kebenaran dan kompleksitas waktu algoritma brute-force, serta mengaplikasikan strategi tersebut dalam pemecahan masalah dengan baik dan benar
- Mahasiswa mampu menjelaskan konsep algoritma rekursif, menuliskan pseudocode, menganalisis kebenaran, memformulasikan bentuk rekursif dari fungsi kompleksitas waktunya dan menghitung rumus eksplisit fungsi tersebut, , serta mengaplikasikan metode rekursif dalam pemecahan masalah dan mengimplementasikannya dalam program komputer dengan baik dan benar

DESKRIPSI TUGAS

Quiz diberikan dalam bentuk pengisian Google Form pada pertemuan perkuliahan. Hasil pengerjaan Quiz dikumpulkan melalui Google Form.

Soal nomor 1

Jelaskan komponen utama pada algoritma rekursif

Long answer text

Soal nomor 2

Jelaskan perbedaan metode rekursif dan metode iteratif

Long answer text

Soal nomor 3

Jelaskan konsep prosedur algoritma rekursif untuk menghitung jumlah suatu array

Long answer text

Soal nomor 4

Jelaskan konsep prosedur algoritma rekursif untuk mencari max dari array dengan cara membagi dua array

Long answer text

Soal nomor 5

Jelaskan bagaimana cara menghitung kompleksitas waktu pada algoritma max array tersebut, sehingga fungsi eksplisit kompleksitas waktu ditemukan

Long answer text

TAHAPAN Pengerjaan Tugas

1. Quiz dilaksanakan sebelum pembelajaran pada pertemuan 6 dimulai. Quiz bertujuan untuk mengecek pemahaman mahasiswa terkait dengan materi yang didiskusikan pada pertemuan sebelumnya.
2. Mahasiswa mengakses link Google Form yang diberikan, mengerjakan quiz, dan mengumpulkan melalui Google Form tersebut.

Bentuk dan Format Luaran

Jawaban tertulis pada Google Form

Indikator, Kriteria dan Bobot Penilaian

Penilaian dengan angka skor (0-100), dengan kriteria penilaian:

1. Ketepatan jawaban (100%)

Jadwal Pelaksanaan	Waktu
Pemberian tugas dan materi terkait	Pertemuan 6
Pengumpulan tugas	Pertemuan 6

Lain-lain

Kriteria penilaian sesuai dengan deskripsi pada lembar penugasan. Penilaian secara umum meliputi kebenaran dan kejelasan penyelesaian tugas, integritas dalam pengerjaan tugas, ketepatan waktu penyelesaian tugas.

Daftar Rujukan

- Materi perkuliahan pertemuan 2 tentang Strategi Brute-Force dan Strategi Rekursif
- Introduction to The Design & Analysis of Algorithms, Anany Levitin, Pearson Education, Inc