

RENCANA PEMBELAJARAN SEMESTER (RPS)

CSG3F3

DESAIN DAN ANALISIS ALGORITMA



Disusun oleh:

Gia Septiana Wulandari
Rimba Widhiana Cipta Sari

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

LEMBAR PENGESAHAN

Rencana Pembelajaran Semester (RPS) ini telah disahkan untuk mata kuliah sbb:

Kode Mata Kuliah : **CSG3F3**

Nama Mata Kuliah : **Desain dan Analisis Algoritma**

Mengetahui
Kaprosdi S1 <Nama Prodi>

M. Arif Bijaksana, PhD

Bandung, 2015
Menyetujui
Ketua KK ICM

Ari M. Barmawi, Ph.D.

DAFTAR ISI

| | |
|--|-----|
| LEMBAR PENGESAHAN..... | ii |
| DAFTAR ISI..... | iii |
| A. PROFIL MATA KULIAH..... | 1 |
| B. RENCANA PEMBELAJARAN SEMESTER (RPS) | 2 |
| C. RANCANGAN INTERAKSI DOSEN–MAHASISWA | 9 |
| D. RANCANGAN TUGAS | 9 |
| E. PENILAIAN DENGAN RUBRIK | 22 |
| F. PENENTUAN NILAI AKHIR MATA KULIAH | 25 |

A. PROFIL MATA KULIAH

IDENTITAS MATA KULIAH

| | | | |
|----------------------|---|--|------------------|
| Nama Mata Kuliah | : | Desain dan Analisis Algoritma | |
| Kode Mata Kuliah | : | CSG3F3 | |
| SKS | : | 3 (tiga) | |
| Jenis | : | MK Wajib | |
| Jam pelaksanaan | : | Tatap muka di kelas | 3 jam per minggu |
| | | Tutorial / responsi | 1 jam per minggu |
| Semester / Tingkat | : | 6 / 3 | |
| <i>Pre-requisite</i> | : | Algoritma Pemrograman, Matematika Diskret, Kalkulus, Struktur Data | |
| <i>Co-requisite</i> | : | | |
| Bidang Kajian | : | Algorithm, Intelligent Computing | |

DESKRIPSI SINGKAT MATA KULIAH

Mata kuliah Desain dan Analisis Algoritma (DAA) mencakup pembahasan algoritma yang ditinjau dari sisi kebenaran (*correctness*), kompleksitas waktu (*time complexity*), dan efisiensi memori (*storage efficiency*). Perkuliahan sedikit menyinggung tentang pembuktian kebenaran program dengan *loop invariant*, pembahasan mengenai notasi asimtotik, dan pemodelan waktu eksekusi (running-time) dari algoritma rekursif dengan relasi rekurensi serta penyelesaiannya. Metode penyelesaian masalah (*problem solving*) yang diberikan mencakup: *brute force/ exhaustive search, divide and conquer technique, branch and bound technique, dynamic programming, greedy methods*, serta beberapa metode untuk menyelesaikan masalah *string matching/ pattern matching*.

DAFTAR PUSTAKA

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. *Introduction to Algorithms – 3rd Edition*, MIT Press, 2009.
2. A. Levitin. *Introduction to The Design and Analysis of Algorithms – 3rd Edition*, Pearson, 2011.
3. R. Neapolitan, K. Naimipour. *Foundations of Algorithms – 5th Edition*, Jones and Bartlett Learning, 2014.

Referensi

1. *Diktat Strategi Algoritmik IF2251*
Ir. Rinaldi Munir, M.T
Departemen Teknik Informatika, Institut Teknologi Bandung

B. RENCANA PEMBELAJARAN SEMESTER (RPS)

| Pertemuan ke- | Kemampuan Akhir yang Diharapkan | Bahan Kajian (Materi Ajar) | Bentuk/ Metode/ Strategi Pembelajaran | Kriteria Penilaian (Indikator) | Bobot Nilai |
|---------------|---|--|--|---|-------------|
| 1 | Mampu memahami apa yang dimaksud dengan Desain dan Analisis Algoritma, tahapan penyelesaian masalah, kaitan struktur data terhadap efisiensi algoritma. | <ul style="list-style-type: none"> - Silabus, TIU, TIK, dll - Top-down & bottom-up programming - Proving correctness of algorithm - Transforming & optimizing algorithm | <ul style="list-style-type: none"> - Ceramah - Diskusi | <ul style="list-style-type: none"> • Mahasiswa mampu mengerti inti permasalahan fundamental dalam bahasan problem solving. • Mahasiswa memahami hubungan antara pemilihan struktur data dan efisiensi sebuah algoritma. | |
| 2 | Mampu membuktikan kebenaran algoritma baik yang bersifat recursive maupun nonrecursive | <ul style="list-style-type: none"> - Mathematical induction (selection sort) → standish | <ul style="list-style-type: none"> - Ceramah - Diskusi | <ul style="list-style-type: none"> • Mahasiswa memahami dan mampu mengidentifikasi loop invariant • Mahasiswa mampu membuktikan kebenaran algoritma | 7.5% |
| 3 | | <ul style="list-style-type: none"> - Correctness of nonrecursive algorithm. Case study: the sum of the integers in the array A[1: n] - Homework: nonrecursive algorithm of Fibonacci → lecture notes | <ul style="list-style-type: none"> - Ceramah - Diskusi - Latihan soal | | |
| 4 | | Quiz | | | |

| | | | | | |
|---|---|---|--|--|-----|
| 5 | <ul style="list-style-type: none"> • Memahami cara penentuan ukuran masukan suatu (<i>input size</i>) algoritma. • Memahami jenis-jenis operasi dasar (<i>basic operation</i>) dan biaya komputasinya (<i>computational cost</i>). • Memahami kerangka yang digunakan dalam mengukur kompleksitas waktu. | <ul style="list-style-type: none"> - problems types (pengelompokan persoalan) dalam computer science [levitin] - pentingnya algoritma yang efisien [cormen] - Analysis framework: measuring an input's size, units for measuring running time, & worst-case, best-case, average-case efficiencies. [levitin] | <ul style="list-style-type: none"> - Ceramah - Diskusi - Latihan soal | <p>Mahasiswa mampu mengukur secara kuantitatif kompleksitas sebuah problem.</p> <p>Mahasiswa mampu menganalisa scope sebuah problem dengan mengidentifikasi worst, best, dan average-case nya.</p> | 5% |
| 6 | <ul style="list-style-type: none"> • Mampu menghitung kompleksitas waktu asimptotik untuk <i>running time</i> suatu algoritma. • Mampu membandingkan <i>asymptotic running time</i> dari dua buah algoritma atau lebih. | <ul style="list-style-type: none"> - Definisi formal notasi asimptotik, notasi: O besar, o kecil, Omega besar, omega kecil, dan Theta besar. - Memahami kelas-kelas dasar kompleksitas suatu algoritma: <i>polynomial-time</i>, <i>exponential-time</i>, <i>subexponential-time</i>, dan <i>logarithmic-time</i>. | <ul style="list-style-type: none"> - Ceramah - Diskusi - Latihan soal | <p>Mahasiswa mampu menganalisa dan menghitung (secara matematis) kompleksitas sebuah algoritma.</p> | 5% |
| 7 | <ul style="list-style-type: none"> • Memahami cara | <ul style="list-style-type: none"> - Mathematical analysis of nonrecursive | <ul style="list-style-type: none"> - Ceramah - Diskusi | <p>Mahasiswa mampu menganalisa dan</p> | 10% |

| | | | | | |
|-------|--|---|---|---|------|
| | memodelkan <i>running time</i> dari suatu algoritma rekursif dalam suatu relasi rekurensi/ persamaan rekurensi. | algorithms | | menghitung kompleksitas algoritma rekursif dengan beberapa teknik matematis tertentu. | |
| 8 | | <ul style="list-style-type: none"> - Solving recurrences by substitution and characteristic equation - Mathematical analysis of recursive algorithms | <ul style="list-style-type: none"> - Ceramah - Diskusi - Latihan soal | | |
| 9 | <ul style="list-style-type: none"> • Mampu menyelesaikan relasi rekurensi dengan beberapa metode: evaluasi iteratif, substitusi, persamaan karakteristik, dan | <ul style="list-style-type: none"> - Mathematical analysis of recursive algorithms | <ul style="list-style-type: none"> - Ceramah - Diskusi - Latihan soal | | |
| 10 | <ul style="list-style-type: none"> • Memahami pengertian pemecahan masalah dengan strategi <i>brute force/ exhaustive search</i>. | <ul style="list-style-type: none"> - Definisi strategi <i>brute force/exhaustive search</i> - karakteristik brute force - contoh penggunaan (a^n algorithm, $n!$ algorithm, etc) | <ul style="list-style-type: none"> - Ceramah - Diskusi - Praktek Pemrograman | Mahasiswa memahami karakteristik metode Brute Force dalam melakukan problem solving. | 7.5% |
| 11-12 | <ul style="list-style-type: none"> • Memahami karakteristik strategi <i>brute force/ exhaustive search</i> dalam penyelesaian suatu permasalahan. | <ul style="list-style-type: none"> - String matching + demo (optional) - Closest-pair - Exhaustive search [rinaldi's slide] | <ul style="list-style-type: none"> - Ceramah - Diskusi - Tugas kecil | Mahasiswa mampu mendesain algoritma Brute Force untuk menyelesaikan problem tertentu. | |
| 13 | <ul style="list-style-type: none"> • Memahami pengertian | <ul style="list-style-type: none"> - Definisi <i>greedy algorithms</i>. - Studi kasus | <ul style="list-style-type: none"> - Ceramah - Diskusi | | 10% |

| | | | | | |
|----|--|---|---|--|-----|
| | <p>pemecahan masalah dengan strategi <i>greedy algorithms</i>.</p> <ul style="list-style-type: none"> • Memahami karakteristik strategi <i>greedy algorithms</i> • Mampu menentukan <i>greedy choice property</i> dari suatu permasalahan. | <p>permasalahan: Coin-changing problem; Scheduling problem; 0/1 knapsack problem: greedy by profit, weight, and density; Fractional knapsack problem)</p> <ul style="list-style-type: none"> - Greedy with Heuristic - Perbandingan dengan penyelesaian exhaustive search | | | |
| 14 | | <p>Studi kasus: Penjadwalan dengan deadline ;MST : prim & kruskal algorithm; Shortest path: dijkstra; Huffman code</p> | <ul style="list-style-type: none"> - Ceramah - Diskusi - Praktek Pemrograman | | |
| 15 | <ul style="list-style-type: none"> • Memahami pengertian pemecahan masalah dengan strategi <i>divide-and-conquer</i> • Memahami langkah-langkah pemecahan masalah dengan | <ul style="list-style-type: none"> - Pengertian DC, skema umum DC - Persoalan pencarian nilai ekstrem, bandingkan dg brute force - Persoalan closest-pair, bandingkan dengan brute force | <ul style="list-style-type: none"> - Ceramah - Diskusi - Tugas | <p>Mahasiswa memahami karakteristik metode Divide and Conquer dalam melakukan problem solving.</p> <p>Mahasiswa mampu mendesain algoritma Divide and Conquer untuk menyelesaikan problem</p> | 10% |

| | | | | | |
|----|---|--|---|-----------|-------|
| | <p>metode <i>divide-and-conquer</i>.</p> <ul style="list-style-type: none"> • Mampu memodelkan kompleksitas waktu asimptotik dari algoritma yang memakai pendekatan <i>divide-and-conquer</i>. | <ul style="list-style-type: none"> - Tugas kelompok topik2 DC - Perbedaan DC dg REKURSIF. C/ sum of elements, exponentiation | | tertentu. | |
| 16 | <ul style="list-style-type: none"> • Memahami karakteristik strategi <i>divide-and-conquer</i> dalam penyelesaian suatu masalah. | <ul style="list-style-type: none"> - Algoritma pengurutan: quick-sort & selection sort - Perpangkatan - Perkalian matriks | <ul style="list-style-type: none"> - Ceramah - Diskusi - Praktek Pemrograman | | |
| 17 | <p>Setelah mengikuti mata kuliah pokok bahasan <i>Backtracking</i>, mahasiswa mampu memahami karakteristik strategi Backtracking dalam menyelesaikan suatu persoalan.</p> | <ul style="list-style-type: none"> - Pengertian backtracking, properti umum backtracking - Prinsip pencarian solusi - The N-queens problem & algoritmanya - The maze problem | <ul style="list-style-type: none"> - Ceramah - Diskusi - Praktek Pemrograman | | 12.5% |
| 18 | <ul style="list-style-type: none"> • Memahami penyelesaian masalah dengan metode <i>branch and bound</i>. • Memahami | <ul style="list-style-type: none"> - Pengertian branch bound - Knapsack problem - Prinsip breadth first search, struktur data, algoritma | <ul style="list-style-type: none"> - Ceramah - Diskusi - Tugas | | 12.5% |

| | | | | | |
|----|---|---|---|--|-------|
| | <p>karakteristik metode <i>branch and bound</i> dalam menyelesaikan masalah.</p> | <ul style="list-style-type: none"> - Tracing algoritma u/ memahami pembentukan pohon ruang status | | | |
| 19 | <ul style="list-style-type: none"> • Memahami karakteristik masalah yang dapat diselesaikan dengan metode <i>branch and bound</i>. | <ul style="list-style-type: none"> - 0/1 knapsack dg best first search, tracing algoritma - TSP dg best first search, tracing algoritma | <ul style="list-style-type: none"> - Ceramah - Diskusi - Praktek Pemrograman | | |
| 20 | <ul style="list-style-type: none"> • Memahami penyelesaian masalah dengan metode <i>dynamic programming</i>. • Mampu mengenali <i>optimal substructure</i> dari suatu masalah yang akan diselesaikan dengan <i>dynamic programming</i>. | <ul style="list-style-type: none"> - Pengertian dynamic programming - Pendekatan DP - Shortest path problem - Permutasi | <ul style="list-style-type: none"> - Ceramah - Diskusi | | 12.5% |
| 21 | <ul style="list-style-type: none"> • Mampu menyusun solusi rekursif dari suatu masalah yang akan diselesaikan dengan <i>dynamic programming</i>. | <ul style="list-style-type: none"> - Persoalan penganggaran modal - Integer knapsack | <ul style="list-style-type: none"> - Ceramah - Diskusi - Praktek Pemrograman | | |

| | | | | | |
|----|--|---|---|--|------|
| | <ul style="list-style-type: none"> • Memahami teknik <i>memoization</i> pada <i>dynamic programming</i>. • Memahami perbedaan antara <i>dynamic programming</i> dan <i>divide-and-conquer</i>. | | | | |
| 22 | <ul style="list-style-type: none"> • Memahami permasalahan <i>pattern matching</i>. • Dapat menyelesaikan masalah <i>pattern matching</i> dengan algoritma Horspool, Boyer-Moore, dan KMP. | <ul style="list-style-type: none"> - Definisi masalah <i>pattern matching</i> - Knuth-Morris-Pratt (KMP) dan perbandingannya dengan brute force | <ul style="list-style-type: none"> - Ceramah - Diskusi - Praktek Pemrograman | | 7.5% |
| 23 | | <ul style="list-style-type: none"> - Horspool, Booyer-Moore dan perbandingannya dengan brute force | <ul style="list-style-type: none"> - Ceramah - Diskusi - Latihan soal | | |

C. RANCANGAN INTERAKSI DOSEN–MAHASISWA

1. Materi dasar-dasar penyelesaian masalah (*problem solving*) secara algoritmik.

| | |
|---|--|
| Kemampuan Akhir yang Diharapkan | Mampu memahami apa yang dimaksud dengan Desain dan Analisis Algoritma, tahapan penyelesaian masalah, kaitan struktur data terhadap efisiensi algoritma. |
| Nama Kajian | <ol style="list-style-type: none"> 1. Silabus, TIU, TIK, dll 2. Top-down & bottom-up programming 3. Proving correctness of algorithm 4. Transforming & optimizing algorithm |
| Nama Strategi | Ceramah dan diskusi |
| Minggu Penggunaan Strategi (Metode) | 1 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> . |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> <p>Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan</p> <p>Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa.</p> <p>Menyimpulkan materi</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> <p>Menjawab pertanyaan yang diberikan.</p> <p>Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas.</p> <p>Menyimak kesimpulan.</p> |

2. Materi Pembuktian Algoritma (Correctness)

| | |
|---------------------------------|---|
| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> • Mampu membuktikan kebenaran algoritma baik yang bersifat recursive maupun |
|---------------------------------|---|

| | nonrecursive |
|---|--|
| Nama Kajian | <ol style="list-style-type: none"> 1. Mathematical induction (selection sort) → standish 2. Correctness of nonrecursive algorithm. Case study: the sum of the integers in the array A[1: n] 3. Homework: nonrecursive algorithm of Fibonacci → lecture |
| Nama Strategi | Ceramah, diskusi, dan pemberian latihan soal. |
| Minggu Penggunaan Strategi (Metode) | 1-2 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian latihan soal dilakukan untuk membantu mahasiswa berlatih soal. |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> <p>Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan</p> <p>Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa.</p> <p>Menyimpulkan materi</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> <p>Menjawab pertanyaan yang diberikan.</p> <p>Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas.</p> <p>Menyimak kesimpulan.</p> |

3. Materi kerangka pengukuran waktu eksekusi algoritma.

| | |
|---------------------------------|--|
| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> • Memahami cara penentuan ukuran masukan suatu (<i>input size</i>) algoritma. • Memahami jenis-jenis operasi dasar (<i>basic operation</i>) dan biaya komputasinya (<i>computational cost</i>). • Memahami kerangka yang digunakan dalam |
|---------------------------------|--|

| | mengukur kompleksitas waktu. |
|---|--|
| Nama Kajian | <ol style="list-style-type: none"> 1. problems types (pengelompokkan persoalan) dalam computer science [levitin] 2. pentingnya algoritma yang efisien [cormen] 3. Analysis framework: measuring an input's size, units for measuring running time, & worst-case, best-case, average-case efficiencies. [levitin] |
| Nama Strategi | Ceramah, diskusi, dan pemberian latihan soal. |
| Minggu Penggunaan Strategi (Metode) | 3 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian latihan soal dilakukan untuk membantu mahasiswa berlatih soal. |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> <p>Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan</p> <p>Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa.</p> <p>Menyimpulkan materi</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> <p>Menjawab pertanyaan yang diberikan.</p> <p>Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas.</p> <p>Menyimak kesimpulan.</p> |

4. Materi kompleksitas waktu asimptotik untuk *running time* suatu algoritma.

| | |
|---------------------------------|--|
| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> • Mampu menghitung kompleksitas waktu asimptotik untuk <i>running time</i> suatu algoritma. • Mampu membandingkan <i>asymptotic running time</i> dari dua buah algoritma atau |
|---------------------------------|--|

| | lebih. |
|---|--|
| Nama Kajian | <ol style="list-style-type: none"> 1. Definisi formal notasi asimptotik, notasi: O besar, o kecil, Ω besar, ω kecil, dan Θ besar. 2. Memahami kelas-kelas dasar kompleksitas suatu algoritma: <i>polynomial-time</i>, <i>exponential-time</i>, <i>subexponential-time</i>, dan <i>logarithmic-time</i>. |
| Nama Strategi | Ceramah, diskusi, dan pemberian latihan soal. |
| Minggu Penggunaan Strategi (Metode) | 3 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian latihan soal dilakukan untuk membantu mahasiswa berlatih soal. |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> <p>Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan</p> <p>Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa.</p> <p>Menyimpulkan materi</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> <p>Menjawab pertanyaan yang diberikan.</p> <p>Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas.</p> <p>Menyimak kesimpulan.</p> |

5. Materi pemodelan *running time* algoritma rekursif dan iteratif.

| | |
|---------------------------------|--|
| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> • Memahami cara memodelkan <i>running time</i> dari suatu algoritma rekursif dalam suatu relasi rekurensi/ persamaan rekurensi. • Mampu menyelesaikan relasi rekurensi dengan beberapa metode: evaluasi iteratif, |
|---------------------------------|--|

| | |
|---|--|
| | substitusi, persamaan karakteristik, dan metode master (<i>master theorem</i>). |
| Nama Kajian | <ol style="list-style-type: none"> 1. Mathematical analysis of nonrecursive algorithms 2. Solving recurrences by substitution and characteristic equation 3. Mathematical analysis of recursive algorithms 4. Mathematical analysis of recursive algorithms |
| Nama Strategi | Ceramah, diskusi, dan pemberian latihan soal. |
| Minggu Penggunaan Strategi (Metode) | 4-5 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian latihan soal dilakukan untuk membantu mahasiswa berlatih soal. |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> <p>Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan</p> <p>Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa.</p> <p>Menyimpulkan materi</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> <p>Menjawab pertanyaan yang diberikan.</p> <p>Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas.</p> <p>Menyimak kesimpulan.</p> |

6. Materi *brute force/ exhaustive search method*.

| | |
|---------------------------------|--|
| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> • Memahami pengertian pemecahan masalah dengan strategi <i>brute force/ exhaustive search</i>. |
|---------------------------------|--|

| | <ul style="list-style-type: none"> Memahami karakteristik strategi <i>brute force/ exhaustive search</i> dalam penyelesaian suatu permasalahan. |
|---|--|
| Nama Kajian | <ol style="list-style-type: none"> Definisi strategi <i>brute force/exhaustive search</i> karakteristik brute force contoh penggunaan (a^n algorithm, $n!$ algorithm, etc) String matching + demo (optional) Closest-pair Exhaustive search [rinaldi's slide] |
| Nama Strategi | Ceramah, diskusi, dan praktik pemrograman. |
| Minggu Penggunaan Strategi (Metode) | 5-6 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian tugas pemrograman diberikan untuk membantu mahasiswa memahami sebuah algoritma. |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> <p>Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan</p> <p>Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa.</p> <p>Menyimpulkan materi</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> <p>Menjawab pertanyaan yang diberikan.</p> <p>Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas.</p> <p>Menyimak kesimpulan.</p> |

7. Materi *greedy approach*.

| | |
|---------------------------------|---|
| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> Memahami pengertian pemecahan masalah |
|---------------------------------|---|

| | |
|---|--|
| | <p>dengan strategi <i>greedy algorithms</i>.</p> <ul style="list-style-type: none"> • Memahami karakteristik strategi <i>greedy algorithms</i> • Mampu menentukan <i>greedy choice property</i> dari suatu permasalahan. |
| Nama Kajian | <ol style="list-style-type: none"> 1. Definisi <i>greedy algorithms</i>. 2. Studi kasus permasalahan: Coin-changing problem; Scheduling problem; 0/1 knapsack problem: greedy by profit, weight, and density; Fractional knapsack problem, Penjadwalan dengan deadline ; MST : prim & kruskal algorithm; Shortest path: dijkstra; Huffman code) 3. Greedy with Heuristic 4. Perbandingan dengan penyelesaian exhaustive search |
| Nama Strategi | Ceramah, diskusi, dan praktik pemrograman. |
| Minggu Penggunaan Strategi (Metode) | 7 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian tugas pemrograman diberikan untuk membantu mahasiswa memahami sebuah algoritma. |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> <p>Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan</p> <p>Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa.</p> <p>Menyimpulkan materi</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> <p>Menjawab pertanyaan yang diberikan.</p> <p>Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas.</p> <p>Menyimak kesimpulan.</p> |

8. Materi *divide-and-conquer technique*.

| | |
|--|---|
| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> • Memahami pengertian pemecahan masalah dengan strategi <i>divide-and-conquer</i> • Memahami langkah-langkah pemecahan masalah dengan metode <i>divide-and-conquer</i>. • Mampu memodelkan kompleksitas waktu asimptotik dari algoritma yang memakai pendekatan <i>divide-and-conquer</i>. • Memahami karakteristik strategi <i>divide-and-conquer</i> dalam penyelesaian suatu masalah. |
| Nama Kajian | <ol style="list-style-type: none"> 1. Pengertian DC, skema umum DC 2. Persoalan pencarian nilai ekstrem, bandingkan dg brute force 3. Persoalan closest-pair, bandingkan dengan brute force 4. Perbedaan DC dg REKURSIF. 5. Algoritma pengurutan: quick-sort & selection sort 6. Perpangkatan 7. Perkalian matriks |
| Nama Strategi | Ceramah, diskusi, dan praktik pemrograman. |
| Minggu Penggunaan Strategi (Metode) | 8 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian tugas pemrograman diberikan untuk membantu mahasiswa memahami sebuah algoritma. |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> |

| | |
|--|--|
| Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan | Menjawab pertanyaan yang diberikan. |
| Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa. | Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas. |
| Menyimpulkan materi | Menyimak kesimpulan. |

9. Materi metode *backtracking*.

| | |
|--|---|
| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> • Memahami pengertian penyelesaian masalah dengan metode <i>backtracking</i>. • Memahami karakteristik <i>backtracking</i> dalam menyelesaikan masalah. • Memahami karakteristik masalah yang dapat diselesaikan dengan metode <i>backtracking</i>. |
| Nama Kajian | <ol style="list-style-type: none"> 1. Pengertian <i>backtracking</i>, properti umum <i>backtracking</i> 2. Prinsip pencarian solusi 3. The N-queens problem & algoritmanya 4. The maze problem |
| Nama Strategi | Ceramah, diskusi, dan praktik pemrograman. |
| Minggu Penggunaan Strategi (Metode) | 9 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian tugas pemrograman diberikan untuk membantu mahasiswa memahami sebuah algoritma. |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> |

| | |
|--|--|
| Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan | Menjawab pertanyaan yang diberikan. |
| Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa. | Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas. |
| Menyimpulkan materi | Menyimak kesimpulan. |

10. Materi metode *branch and bound*.

| | |
|---|--|
| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> • Memahami penyelesaian masalah dengan metode <i>branch and bound</i>. • Memahami karakteristik metode <i>branch and bound</i> dalam menyelesaikan masalah. • Memahami karakteristik masalah yang dapat diselesaikan dengan metode <i>branch and bound</i>. |
| Nama Kajian | <ol style="list-style-type: none"> 1. Pengertian branch bound 2. Knapsack problem 3. Prinsip breadth first search, struktur data, algoritma 4. Tracing algoritma u/ memahami pembentukan pohon ruang status 5. 0/1 knapsack dg best first search, tracing algoritma 6. TSP dg best first search, tracing algoritma |
| Nama Strategi | Ceramah, diskusi, dan praktik pemrograman. |
| Minggu Penggunaan Strategi (Metode) | 9-10 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian tugas pemrograman diberikan untuk membantu mahasiswa memahami sebuah algoritma. |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran. | Menyimak penjelasan dosen. |

| | |
|--|--|
| Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran. | Menyiapkan diri menerima materi yang akan disampaikan. |
| Membahas materi. | Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen. |
| Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan | Bertanya apabila ada materi yang kurang jelas. |
| Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa. | Menjawab pertanyaan yang diberikan. |
| Menyimpulkan materi | Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas. |
| | Menyimak kesimpulan. |

11. Materi *dynamic programming*.

| | |
|--|---|
| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> • Memahami penyelesaian masalah dengan metode <i>dynamic programming</i>. • Mampu mengenali <i>optimal substructure</i> dari suatu masalah yang akan diselesaikan dengan <i>dynamic programming</i>. • Mampu menyusun solusi rekursif dari suatu masalah yang akan diselesaikan dengan <i>dynamic programming</i>. • Memahami teknik <i>memoization</i> pada <i>dynamic programming</i>. • Memahami perbedaaan antara <i>dynamic programming</i> dan <i>divide-and-conquer</i>. |
| Nama Kajian | <ol style="list-style-type: none"> 1. Pengertian dynamic programming 2. Pendekatan DP 3. Shortest path problem 4. Permutasi 5. Persoalan penganggaran modal 6. Integer knapsack |
| Nama Strategi | Ceramah, diskusi, dan praktik pemrograman. |
| Minggu Penggunaan Strategi (Metode) | 10-11 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian tugas pemrograman diberikan untuk membantu mahasiswa memahami |

| | sebuah algoritma. |
|---|--|
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> <p>Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan</p> <p>Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa.</p> <p>Menyimpulkan materi</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> <p>Menjawab pertanyaan yang diberikan.</p> <p>Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas.</p> <p>Menyimak kesimpulan.</p> |

12. Materi algoritma-algoritma *pattern matching*.

| Kemampuan Akhir yang Diharapkan | <ul style="list-style-type: none"> • Memahami permasalahan <i>pattern matching</i>. • Dapat menyelesaikan masalah <i>pattern matching</i> dengan algoritma Horspool, Boyer-Moore, dan KMP. |
|--|---|
| Nama Kajian | <ol style="list-style-type: none"> 1. Definisi masalah <i>pattern matching</i>. 2. Studi kasus pemecahan masalah <i>pattern matching</i> dengan algoritma Horspool, Boyer Moore, dan KMP. |
| Nama Strategi | Ceramah, diskusi, dan praktik pemrograman. |
| Minggu Penggunaan Strategi (Metode) | 11-12 |
| Deskripsi Singkat Strategi (Metode) pembelajaran | Dosen memberikan ceramah mengenai materi yang diajarkan; diskusi dilakukan di kelas maupun IDEA sebagai media <i>e-learning</i> ; pemberian tugas pemrograman diberikan untuk membantu mahasiswa memahami sebuah algoritma. |
| RANCANGAN INTERAKSI DOSEN–MAHASISWA | |
| Aktivitas Dosen | Aktivitas Mahasiswa |

| | |
|---|--|
| <p>Menjelaskan tentang tujuan pembelajaran dari kegiatan pembelajaran.</p> <p>Mengarahkan mahasiswa untuk melibatkan diri dan aktif dalam kegiatan pembelajaran.</p> <p>Membahas materi.</p> <p>Mengajukan sejumlah pertanyaan terkait materi yang telah diberikan</p> <p>Memberikan tugas sebagai sarana berlatih dan evaluasi diri kepada mahasiswa.</p> <p>Menyimpulkan materi</p> | <p>Menyimak penjelasan dosen.</p> <p>Menyiapkan diri menerima materi yang akan disampaikan.</p> <p>Menyimak dan mencatat hal-hal penting dari materi yang disampaikan oleh dosen.</p> <p>Bertanya apabila ada materi yang kurang jelas.</p> <p>Menjawab pertanyaan yang diberikan.</p> <p>Mengerjakan tugas dengan baik sesuai dengan arahan dosen, tidak melakukan tindak plagiarisme dalam pengerjaan tugas.</p> <p>Menyimak kesimpulan.</p> |
|---|--|

D. RANCANGAN TUGAS

| | |
|---------------------------------|---|
| Kode mata Kuliah | CSG3F3 |
| Nama Mata Kuliah | Desain dan Analisis Algoritma |
| Kemampuan Akhir yang Diharapkan | Mampu menyelesaikan sebuah permasalahan dengan dasar analisis kompleksitas algoritma. Memahami dan mampu memilih strategi-strategi umum penyelesaian masalah algoritmik. |

| | |
|---|---------|
| Minggu/Pertemuan ke | 2 / 3-4 |
| Tugas ke | 1 |
| <p>1. Tujuan tugas: Mahasiswa mampu membuktikan kebenaran suatu algoritma, baik rekursif maupun nonrekursif</p> <p>2. Uraian Tugas:</p> <ol style="list-style-type: none">Obyek garapan: pembuktian algoritmaYang harus dikerjakan dan batasan-batasan: Mahasiswa membuktikan suatu algoritma rekursif dan nonrekursif yang diberikanMetode/ cara pengerjaan, acuan yang digunakan: Dikerjakan secara individu, dapat mengacu pada contoh penyelesaian yang telah dibahas di kelas.Deskripsi luaran tugas yang dihasilkan/ dikerjakan: Bukti kebenaran algoritma yang diberikan <p>3. Kriteria penilaian: Kebenaran isi: 80% Kesesuaian dengan tugas: 20%</p> | |

| | |
|---|--------|
| Minggu/Pertemuan ke | 3/ 5-6 |
| Tugas ke | 2 |
| <p>1. Tujuan tugas: Mahasiswa mampu menentukan notasi asimtotik untuk suatu nilai kompleksitas waktu</p> <p>2. Uraian Tugas:</p> <ol style="list-style-type: none">Obyek garapan: Notasi asimtotikYang harus dikerjakan dan batasan-batasan: Mahasiswa menentukan notasi asimtotik dari beberapa nilai kompleksitas waktuMetode/ cara pengerjaan, acuan yang digunakan: Dikerjakan secara individu, dapat mengacu pada contoh penyelesaian yang telah dibahas di kelas.Deskripsi luaran tugas yang dihasilkan/ dikerjakan: Notasi asimtotik kompleksitas waktu yang diberikan <p>3. Kriteria penilaian: Kebenaran isi: 80% Kesesuaian dengan tugas: 20%</p> | |

| | |
|--|---------|
| Minggu/Pertemuan ke | 4 / 7-8 |
| Tugas ke | 3 |
| <p>1. Tujuan tugas: Mahasiswa mampu mengklasifikasikan kelas efisiensi suatu algoritma</p> <p>2. Uraian Tugas:</p> <ol style="list-style-type: none"> Obyek garapan: Kelas efisiensi algoritma Yang harus dikerjakan dan batasan-batasan: Mahasiswa menentukan kelas efisiensi algoritma yang diberikan Metode/ cara pengerjaan, acuan yang digunakan: Dikerjakan secara individu, dapat mengacu pada contoh penyelesaian yang telah dibahas di kelas. Deskripsi luaran tugas yang dihasilkan/ dikerjakan: Kelas efisiensi algoritma yang diberikan <p>3. Kriteria penilaian: Kebenaran isi: 80% Kesesuaian dengan tugas: 20%</p> | |

| | |
|---|-------------|
| Minggu/Pertemuan ke | 5-11/ 10-21 |
| Tugas ke | 4-9 |
| <p>1. Tujuan tugas: Mahasiswa dapat menerapkan teknik penyelesaian masalah (<i>brute force, greedy, divide and conquer, backtracking, branch and bound, dan dynamic programming</i>) dengan tepat</p> <p>2. Uraian Tugas:</p> <ol style="list-style-type: none"> Obyek garapan: Penyelesaian masalah dengan teknik tertentu Yang harus dikerjakan dan batasan-batasan: Mahasiswa menyelesaikan suatu permasalahan dengan menggunakan teknik tertentu Metode/ cara pengerjaan, acuan yang digunakan: Dikerjakan secara individu, dapat mengacu pada contoh penyelesaian yang telah dibahas di kelas. Deskripsi luaran tugas yang dihasilkan/ dikerjakan: Kelas efisiensi algoritma yang diberikan <p>3. Kriteria penilaian: Kebenaran isi: 80% Kesesuaian dengan tugas: 20%</p> | |

| | |
|--|---------------|
| Minggu/Pertemuan ke | 12-14 / 21-28 |
| Tugas ke | TUGAS BESAR |
| <p>1. Tujuan tugas:</p> <ul style="list-style-type: none"> - Mahasiswa memahami suatu masalah - Mahasiswa dapat memilih teknik yang tepat (di antara <i>brute force, greedy, divide and conquer, backtracking, branch and bound, dan dynamic programming</i>) untuk menyelesaikan suatu masalah sederhana <p>2. Uraian Tugas:</p> <ol style="list-style-type: none"> Obyek garapan: <i>brute force, greedy, divide and conquer, backtracking, branch and bound, dan dynamic</i> | |

programming

b. Yang harus dikerjakan dan batasan-batasan:

- Mahasiswa diberikan suatu permasalahan tanpa diberitahu teknik penyelesaiannya.
- Mahasiswa menyelesaikan permasalahan tersebut dengan teknik *brute force* dan satu teknik lainnya di antara *greedy, divide and conquer, backtracking, branch and bound, dan dynamic programming* yang dianggap paling tepat

c. Metode/ cara pengerjaan, acuan yang digunakan:

- Dikerjakan secara berkelompok (4-5 orang)
- Setiap anggota harus mempunyai kontribusi
- Permasalahan yang diberikan dapat dikerjakan dalam waktu yang diberikan
- Penyelesaian masalah dilakukan dengan membuat program
- Laporan tulis dikumpulkan, dan hasil dipresentasikan

d. Deskripsi luaran tugas yang dihasilkan/ dikerjakan:

- Laporan berisi analisis dan solusi permasalahan yang didapatkan, termasuk deskripsi teknik pengerjaan yang dilakukan
- Program untuk mencari solusi masalah yang diberikan

3. Kriteria penilaian:

Ketepatan teknik: 25%

Kesesuaian penerapan teknik: 25%

Pengerjaan *brute force*: 20%

Laporan tertulis: 15%

Teknik presentasi: 15%

E. PERSENTASE KOMPONEN PENILAIAN

1. Kuis : 10%
2. Tugas Besar : 20%
3. UTS : 30%
4. UAS : 40%

F. PENILAIAN DENGAN RUBRIK

Untuk setiap tugas atau learning outcome dilengkapi dengan tabel rubrik penilain sbb:

| Jenjang (Grade) | Angka (Skor) | Deskripsi perilaku (Indikator) |
|-----------------|--------------|--|
| | 7 | Mahasiswa menguasai konsep, jawaban benar sempurna |
| | 6 | Mahasiswa menguasai konsep namun ada sedikit kesalahan pada jawaban |
| | 5 | Mahasiswa mengerti konsep namun ada sedikit kesalahan pada proses penerapannya |
| | 4 | Mahasiswa mengerti konsep namun tidak mampu menerapkannya |
| | 3 | Mahasiswa hanya mengerti sebagian konsep |
| | 2 | Mahasiswa salah mengerti konsep, jawaban benar tetapi prosesnya salah |

| | | |
|--|---|--------------------------|
| | 1 | Mahasiswa tidak mengerti |
|--|---|--------------------------|

G. PENENTUAN NILAI AKHIR MATA KULIAH

Penentuan Nilai Akhir menggunakan standar Universitas Telkom:

| Nilai Skor Matakuliah (NSM) | Nilai Mata Kuliah (NMK) |
|-----------------------------|-------------------------|
| $80 < NSM$ | A |
| $70 < NSM \leq 80$ | AB |
| $65 < NSM \leq 70$ | B |
| $60 < NSM \leq 65$ | BC |
| $50 < NSM \leq 60$ | C |
| $40 < NSM \leq 50$ | D |
| $NSM \leq 40$ | E |