**RENCANA PROGRAM KEGIATAN PERKULIAHAN SEMESTER**

**(RPKPS)**

Kode / Nama Mata Kuliah :CMI 501/Pemrograman Revisi ke : -

Satuan Kredit Semester :3 SKS Tgl revisi : -

Jml Jam kuliah dalam seminggu :150 menit Tgl mulai berlaku :

Penyusun : NUM

Jml Jam kegiatan laboratorium :150 menit Penanggungjawab Keilmuan : Dr. Ir. Gerry Firmansyah

Deskripsi Mata kuliah : Matakuliah ini mempelajari tentang beberapa paradigma pemrograman. Setiap paradigma memiliki sudut pandang tertentu dalam mendeskripsikan konsep, cara memformulasikan algoritma, dan bagaimana solusi terhadap suatu persoalan diberikan. Karena bahasa pemrograman sangat berkembang, maka kuliah ini memberikan beberapa contoh bahasa dalam setiap paradigma pemrograman. Kuliah ini juga memberikan strategi penyelesaian persoalan (strategi algoritma) dan tingkat efisiensi dari tiap strategi untuk menyelesaikan persoalan.

Standar Kompetensi: Setelah mempelajari matakuliah Pemrograman mahasiswa diharapkan mampu

1. Membuat program dengan paradigma tertentu
2. Menyelesaikan persoalan komputasi dengan pendekatan algoritma yang sesuai
3. Mengukur tingkat efisiensi algoritma untuk menyelesaikan suatu persoalan
4. Menentukan kelas kompleksitas dari suatu persoalan yang diberikan

| **Perte-muan ke :** | **Kompetensi Dasar** | **Indikator** | **Pokok Bahasan/Materi** | **Aktifitas Pembelajaran** | **Rujukan** |
| --- | --- | --- | --- | --- | --- |
| 1 | Mahasiswa dapat menjelaskan dan memahami pengertian berpikir komputasional | 1. Mampu menjelaskan definisi berpikir komputasional 2. Mampu menjelaskan tahapan dalam berpikir komputasional | * Definisi dan motivasi berpikir komputasional * Contoh berpikir komputasional dalam menyelesaikan persoalan | **Pendahuluan**   1. Melakukan perkenalan diri 2. Menjelaskan cakupan perkuliahan selama satu semester   **Penyajian**   1. Menjelaskan aturan kuliah selama satu semester 2. Menjelaskan penugasan-penugasan yang ada pada mata kuliah 3. Menjelasan definisi berpikir komputasional, motivasi pentingnya berpikir komputasional dalam era teknologi informasi 4. Penjelasan tahapan dalam berpikir komputasional, dan memberikan contoh dalam kehidupan sehari-hari, dan dalam penyelesaian persoalan dengan memanfaatkan komputer   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 1 |
| 2 | Mahasiswa mengenal beberapa paradigma pemrograman dan perbedaan antar paradigma | 1. Mahasiswa dapat menjelaskan minimal empat paradigma pemrograman 2. Mahasiswa dapat menjelaskan perbedaan dari paradigma pemrograma yang disebutkan | * Definisi dari paradigma pemrograman * Pengenalan paradigma pemrograman prosedural, fungsional, deklaratif, berorientasi objek, *event-driven/ reactive* | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**   1. Memperkenalkan beberapa paradigma pemrograman yang ada di dunia saat ini 2. Fokus pada pengenalan lima paradigma pemrograman dan perbedaannya yaitu prosedural, fungsional, deklaratif, berorientasi objek, dan event-driven 3. Memberikan contoh untuk lima paradigma pemrograman yang diperkenalkan   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 2 |
| 3 | Mahasiswa mampu membuat berpikir abstrak dan fungsional, dan mampu membuat program sederhana dalam bahasa pemrograman Haskell | 1. Mahasiswa mampu membuat design solusi persoalan dengan paradigma fungsional 2. Mahasiswa mampu membuat program sederhana dengan bahasa Haskell | * Abstraksi dan “dekomposisi” dalam konteks fungsional: data (type bentukan), fungsi * Ekspresi aritmatika, logika, dan kondisional * Analisis rekurens * Konsep list sebagai struktur rekursif * Operasi dasar list dengan elemen tertentu: integer, character, type bentukan | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menjelaskan:   1. Abstraksi dan “dekomposisi” dalam konteks fungsional: data (type bentukan), fungsi 2. Ekspresi aritmatika, logika, dan kondisional 3. Analisis rekurens 4. Konsep list sebagai struktur rekursif 5. Operasi dasar list dengan elemen tertentu: integer, character, type bentukan   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 2,3 |
| 4 | Mahasiswa memahami pendekatan deklaratif dengan dasar logika, dan mampu membuat program sederhana dalam Prolog | 1. Mahasiswa mampu membuat design solusi persoalan dengan paradigma deklaratif 2. Mahasiswa mampu membuat program sederhana dengan bahasa Prolog | * Input (clause, declarative, query) * Struktur Data (Fakta dan Aturan) * Identifier, Operator, Tipe * Struktur Program * Deklaratif Murni dan Tidak Murni * Analisis Kasus * Analisis Rekurens * Cut & Fail | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menjelaskan:   1. Input (clause, declarative, query) 2. Struktur Data (Fakta dan Aturan) 3. Identifier, Operator, Tipe 4. Struktur Program 5. Deklaratif Murni dan Tidak Murni 6. Analisis Kasus 7. Analisis Rekurens 8. Cut & Fail   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 2, 4 |
| 5 | Mahasiswa memahami pendekatan berorientasi objek, dan mampu membuat program sederhana dalam C++ | 1. Mahasiswa mampu membuat design solusi persoalan dengan paradigma berorientasi objek 2. Mahasiswa mampu membuat program sederhana dengan bahasa C++ | * Konsep Class, Object; * Object life time (creation, allocation, destruction), * OperasidasarObjek (attachment, comparaison); * Genericity; Inheritance & polymorphism; Exception; * Konsep ADT, Mesin, Proses danimplementasidalam OOP; * PemrogramanBerorientasiObjekdalambahasa C++ | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menjelaskan:   1. Konsep Class, Object; 2. Object life time (creation, allocation, destruction), 3. Operasi dasar Objek (attachment, comparaison); 4. Genericity; Inheritance & polymorphism; Exception; 5. Konsep ADT, Mesin, Proses dan implementasi dalam OOP; 6. Pemrograman Berorientasi Objek dalam bahasa C++   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 2,5 |
| 6 | Mahasiswa memahami pendekatan berbasis *event*, dan mampu membuat program sederhana dalam bahasa Java | 1. Mahasiswa mampu membuat design solusi persoalan dengan paradigma berbasis *event/ reactive* 2. Mahasiswa mampu membuat program sederhana dengan bahasa Java | * *Events*dan*event handlers* * Pemanfaatan *reactive framework*:   + Mendefinisikan *event handlers/listeners*   + Lelaran event utama tidak berada pada kontrol pembuat event-handler * Events yang dibangkitkan dari entitas eksternal dan yang dibangkitkan program * Pemisahan model, view, dan controller * Pemrograman dengan bahasa Java | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menjelaskan:   1. Events dan event handlers 2. Pemanfaatan reactive framework :    1. Mendefinisikan event handlers/listeners    2. Lelaran event utama tidak berada pada kontrol pembuat event-handler 3. Events yang dibangkitkan dari entitas eksternal dan yang dibangkitkan program 4. Pemisahan model, view, dan controller 5. Pemrograman dengan bahasa Java   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 6 |
| 7 | Mahasiswa mampu menyelesaikan sebuah persoalan dengan paradigma yang sesuai dengan informasi yang tersedia | Diberikan persoalan tertentu dan ketersediaan informasi/ pengetahuan yang ada, mahasiswa mampu menentukan paradigma yang sesuai untuk persoalan tersebut dan mengimplementasikan solusinya. | * Analisis Kasus * Diskusi mahasiswa dan tugas kelompok | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**   1. Diskusi beberapa kasus 2. Mahasiswa dibagi kelompok dan tiap kelompok menyelesaikan suatu kasus 3. Diskusi antar kelompok untuk kasus-kasus yang dibahas   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 2-6 |
| **Ujian Tengah Semester** | | | | | |
| 8 | Mahasiswa mampu menyelesaikan persoalan dengan paradigma prosedural dengan bahasa Pascal | 1. Mahasiswa mampu membuat design solusi persoalan dengan paradigma prosedural 2. Mahasiswa mampu membuat program sederhana dengan bahasa Pascal | * Beberapa construct dasardalampemrogramanprosedural: I/O, sekuens, analisiskasus, pengulangan, subprogram, array/matriks, file eksternal * Beberapa skema standar program (searching, sorting, pemrosesan file I/O) * Pemrograman dalam bahasa Pascal | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menjelaskan:   1. Beberapa construct dasar dalam pemrograman prosedural: I/O, sekuens, analisis kasus, pengulangan, subprogram, array/matriks, file eksternal 2. Beberapa skema standar program (searching, sorting, pemrosesan file I/O) 3. Pemrograman dalam bahasa Pascal   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 7 |
| 9 | Mahasiswa memahami beberapa tipe persoalan yang penting, dan beberapa jenis struktur data utama | 1. Mahasiswa dapat menjelaskan beberapa jenis tipe persoalan untuk diselesaikan secara komputasional 2. Untuk kasus tertentu mahasiswa dapat menentukan jenis persoalan yang sesuai 3. Mahasiswa mampu menentukan struktur data yang sesuai untuk kasus yang diberikan | * Tipe persoalan:   + Sorting   + Searching   + String Processing   + Persoalan Graf   + Persoalan Kombinatorial   + Persoalan Numerik * Struktur Data Dasar   + Linear   + Graf   + Pohon   + Himpunan (array, matriks) | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menjelaskan dan Diskusi:   1. Tipe persoalan:    1. Sorting    2. Searching    3. String Processing    4. Persoalan Graf    5. Persoalan Kombinatorial    6. Persoalan Numerik 2. Struktur Data Dasar    1. Linear    2. Graf    3. Pohon    4. Himpunan (array, matriks)   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 8 |
| 10 | Mahasiswa mampu melakukan analisis untuk menentukan tingkat efisiensi suatu algoritma penyelesaian persoalan berbasis komputer | 1. Mahasiswa dapat menjelaskan framework untuk melakukan analisis 2. Mahasiswa mampu memberikan analisis terhadap efisiensi suatu algoritma dengan notasi asimptotik | * Framework Analisis   + Pengukuran Input   + Unit untuk mengukur running time   + Orde pertumbuhan   + Efisiensi kasus terburuk, kasus terbaik, dan kasus rata-rata * Notasi Asimptotik   + Notasi O   + Notasi Ω   + Notasi Θ   + Kelas efisiensi dasar | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menjelaskan:   1. Framework Analisis    1. Pengukuran Input    2. Unit untuk mengukur running time    3. Orde pertumbuhan    4. Efisiensi kasus terburuk, kasus terbaik, dan kasus rata-rata 2. Notasi Asimptotik    1. Notasi O    2. Notasi Ω    3. Notasi Θ    4. Kelas efisiensi dasar   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 8 |
| 11 | Mahasiswa mampu menyelesaikan persoalan dengan pendekatan Brute Force dan Exhaustive Search | 1. Mahasiswa mampu menerapkan algoritma brute force untuk menyelesaikan persoalan yang diberikan 2. Mahasiswa mampu menerapkan algoritma exhaustive search untuk menyelesaikan persoalan yang diberikan 3. Mahasiswa mampu menentukan tingkat efisiensi algoritma Burte Force dan Exhaustive Search dengan notasi O, dan menganalisis pendekatan mana yang lebih baik untuk suatu persoalan | * Selection Sort & Bubble Sort * Sequential Search & Brute Force String Matching * Closest Pair & Convex Hull dengan Brute Force * Travelling Salesman Problem, Knapsack Problem, Assignment Problem | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menjelaskan:   1. Selection Sort & Bubble Sort 2. Sequential Search & Brute Force String Matching 3. Closest Pair & Convex Hull dengan Brute Force 4. Travelling Salesman Problem, Knapsack Problem, Assignment Problem   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 8 |
| 12 | Mahasiswa mampu menyelesaikan persoalan dengan pendekatan Decrease and Conquer, Divide and Conquer, serta Transform and Conquer | 1. Mahasiswa mampu menerapkan teknik Decrease and Conquer untuk menyelesaikan persoalan yang diberikan 2. Mahasiswa mampu menerapkan teknik Divide and Conquer untuk menyelesaikan persoalan yang diberikan 3. Mahasiswa mampu menerapkan teknik Transform and Conquer untuk menyelesaikan persoalan yang diberikan 4. Mahasiswa mampu menentukan tingkat efisiensi Decrease and Conquer, Divide and Conquer, serta Transform and Conquer dengan notasi O, dan menganalisis pendekatan mana yang lebih baik untuk suatu persoalan | * + Insertion Sort   + Topological Sort   + Binary Search   + Persoalan Fake Coin   + Menghitung Median dan Persoalan seleksi   + Persoalan Interpolasi   + MergeSort & Quicksort   + Perkalian Matriks Strassen   + Eliminasi Gaussian   + Heap & Heapsort | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menjelaskan:   1. Insertion Sort 2. Topological Sort 3. Binary Search 4. Persoalan Fake Coin 5. Menghitung Median dan Persoalan seleksi 6. Persoalan Interpolasi 7. MergeSort & Quicksort 8. Perkalian Matriks Strassen 9. Eliminasi Gaussian 10. Heap & Heapsort   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 8 |
| 13 | Mahasiswa mampu menyelesaikan persoalan dengan pendekatan Teknik Greedy dan Dynamic Programming | 1. Mahasiswa mampu menerapkan teknik Greedy untuk menyelesaikan persoalan yang diberikan 2. Mahasiswa mampu menerapkan Dynamic Programming untuk menyelesaikan persoalan yang diberikan 3. Mahasiswa mampu menentukan tingkat efisiensi Teknik Greedy dan Dynamic Programming dengan notasi O, dan menganalisis pendekatan mana yang lebih baik untuk suatu persoalan | * Algoritma Prism & Kruskal * Algoritma Djikstra * Algoritma Huffman * Persoalan Knapsack dengan Dynamic Programming * Pohon pencarian biner optimal * Algoritma Warshall & Floyd | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menjelaskan:   1. Algoritma Prism & Kruskal 2. Algoritma Djikstra 3. Algoritma Huffman 4. Persoalan Knapsack dengan Dynamic Programming 5. Pohon pencarian biner optimal   **Algoritma Warshall & Floyd**  **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan materi pertemuan selanjutnya | 8 |
| 14 | Mahasiswa dapat menunjukkan kelebihan dan kekurangan dari aspek ruang dan waktu untuk tiap pendekatan, dan mahasiswa mampu mendefinisikan kelas kompleksitas dari suatu persoalan | 1. Mahasiswa bisa menjelaskan dan menghitung sumber daya waktu yang diperlukan suatu pendekatan untuk menyelesaikan persoalan 2. Mahasiswa bisa menjelaskan dan menghitung sumber daya ruang yang diperlukan suatu pendekatan untuk menyelesaikan persoalan 3. Mahasiswa bisa menganalisis pendekatan terbaik untuk suatu persoalan berdasarkan kebutuhan sumber daya waktu dan ruang 4. Mahasiswa bisa mendefinisikan kelas kompleksitas dari suatu persoalan, dan menentukan pendekatan terbaik untuk kelas kompleksitas tersebut. | * Sorting dengan Counting * Algoritma Horspool dan Boyer Moore untuk String Matching * Hashing * Persoalan Polinomial (P) & Non Polinomial (NP) * Persoalan NP Complete | **Pendahuluan**   1. Melakukan review pertemuan sebelumnya dan kaitannya dengan pertemuan saat ini 2. Menjelaskan kompetensi dasar pertemuan saat ini dan manfaatnya   **Penyajian**  Menyajikan:   1. Sorting dengan Counting 2. Algoritma Horspool dan Boyer Moore untuk String Matching 3. Hashing 4. Persoalan Polinomial (P) & Non Polinomial (NP) 5. Persoalan NP Complete   **Penutup**   1. Merangkum materi perkuliahan pertemuan saat ini 2. Menginformasikan kisi-kisi ujian akhir semester | 8 |
| **Ujian Akhir Semester** | | | | | |

**Level Taksonomi :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Kognitif** | | **Psikomotor** | | **Afektif** | |
| Pengetahuan (knowledge) |  | Peniruan (immitation) |  | Menerima (receiving) |  |
| Pemahaman (comprehension) | 15 % | Manipulasi (manipulation) | 10 % | Menanggapi (responding) | 5 % |
| Penerapan (application) | 15 % | Ketepatan (precision) |  | Menilai (valuing) |  |
| Analisis (analysis) | 30 % | Artikulasi (articulation) |  | Mengelola (organizing) | 5 % |
| Sintesis (synthesis) | 10 % | Pengalamiahan (naturalization) | 10 % | Menghayati (characterizing) |  |
| Evaluasi (evaluation) |  |  |  |  |  |

**Komposisi Penilaian** :

|  |  |
| --- | --- |
| **Aspek Penilaian** | **Prosentase** |
| Ujian Akhir Semester | 30 % |
| Ujian Tengah Semester | 25 % |
| Tugas Mandiri | 20 % |
| Kuis | 15 % |
| Kehadiranan Mahasiswa | 5 % |
| Sikap | 5 % |
| **Total** | **100 %** |

**Daftar Referensi**

1. Computational Thinking for the Modern Problem Solver (Chapman & Hall/CRC Textbooks in Computing) 1st Edition, David D. Riley  &Kenny A. Hunt, Chapman and Hall/CRC; 1 edition (March 27, 2014)
2. Programming Languages: Principles and Paradigms (Undergraduate Topics in Computer Science), Maurizio Gabbrielli, Simone Martini, Springer; 2010 edition (April 15, 2010)
3. Learn You a Haskell for Great Good!: A Beginner's Guide 1st Edition, MiranLipovaca,  No Starch Press; 1 edition (April 21, 2011)
4. Prolog Programming Success in a Day: Beginners Guide to Fast, Easy and Efficient Learning of Prolog Programming, Sam Key, CreateSpace Independent Publishing Platform (August 12, 2015)
5. The C++ Programming Language 3rd Edition, Stroustrup, Addison-Wesley (1997)
6. Learning Reactive Programming with Java 8, NickolayTsvetinov, Packt Publishing (June 24, 2015)
7. Pascal: An Introduction to the Art and Science of Programming (4th Edition), Walter Savitch, Addison Wesley; 4 edition (December 31, 1994)
8. Introduction to the Design and Analysis of Algorithms (2nd Edition), AnanyLevitin, Addison Wesley; 2 edition (February 24, 2006)