**GARIS-GARIS BESAR PROGRAM PEMBELAJARAN (GBPP)**

|  |  |  |
| --- | --- | --- |
| **MATA KULIAH** | : | Desain dan Analisis Algoritma |
| **KODE MATA KULIAH / SKS** | : | IF0013 / 4 SKS |
| **DESKRIPSI SINGKAT** | : | Mata kuliah ini menjelaskan tentang dasar-dasar pemecahan masalah dengan algoritma, pembuktian kebenaran algoritma, dasar analisis efisiensi algoritma, dan macam-macam metode desain algoritma. |
| **PRASYARAT** | : | - |

**SILABUS**

|  |  |  |
| --- | --- | --- |
| **MATA KULIAH** | : | Desain dan Analisis Algoritma |
| **PROGRAM STUDI** | : | Teknik Informatika |
| **JENJANG PROGRAM** | : | Strata I (S1) |
| **TUJUAN INSTRUKSIONAL UMUM** | : | Pada akhir mata kuliah ini mahasiswa akan dapat mendesain algoritma yang baik dan efisien serta mampu menganalisis algoritma dengan metode yang baik dan benar. |
| **POKOK BAHASAN** | : | 1. Pengantar Big O & Model Matematika Big O 2. Brute Force Revisited 3. Divide and Conquer 4. Greedy / Djikstra 5. Dynamic Programming 6. Graph (Minimum Spanning Tree, BFS dan DFS, Kruskal, Prim) 7. String Matching (Suffix Array dan Suffix Trie) 8. Competitive Programming |
| **BAHAN BACAAN** | : | **Buku:**   1. Sedgewick, Robert and Wayne, Kevin, Algorithms (Fourth Edition). Addison-Wesley, 2011. ISBN: 9780321573513 2. Cormen, Thomas. Introduction to Algorithms (Third Edition). The MIT Press, 2009. ISBN: 9780262533058 3. Skiena, Steven. The Algorithm Design Manual (Second Edition). Springer, 2008. ISBN: 9781848000698 4. Abelson, Harold, Sussman, Gerald Jay, Sussman, Julie, Structure and Interpretation of Computer Programs (Second Edition). The MIT Press, 1996. ISBN: 0262510871   **Paper:**   1. Hoare, C.R., An Axiomatic Basis for Computer Programming. Communications of the ACM Volume 12 No 10, 1969. 2. Floyd, Robert, Assigning Meanings to Programs. Symposia in Applied Mathematics, 1969. 3. Djikstra, E. W., Recursive Programming. Numerische Mathematik 2, 312-318, 1960. |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **SAP DESAIN DAN ANALISIS ALGORITMA** | | | | | | | | |
| **NO** | **MINGGU KE** | **POKOK BAHASAN** | **TUJUAN INSTRUKSIONAL UMUM** | **TUJUAN INSTRUKSIONAL KHUSUS** | **MATERI** | **KEGIATAN INSTRUKSIONAL** | | **MEDIA** | **EVALUASI** | **SUMBER** | |
| **DOSEN** | **MAHASISWA** |
| 1 | 1 | Pengantar Big O & Model Matematika Big O | Mahasiswa memahami konsep dasar dari desain dan analisis efisiensi algoritma. | Mahasiswa mampu melakukan perhitungan / penentuan secara mandiri terhadap kompleksitas sebuah algoritma. | * Pengenalan Desain dan Analisis Algoritma * Pembuktian kebenaran Algoritma * Pemodelan permasalahan * Kompleksitas Algoritma * Notasi Asimtotik | Menerangkan | Memerhatikan, mencatat, mengerjakan latihan. | Papan tulis, proyektor | Latihan | 1-7 | |
| 2 | 2 | Brute Force Revisited | Mahasiswa mempelajari beberapa studi kasus untuk brute force dan menghitung analisisnya. | Mahasiswa mampu melakukan analisis terhadap metode brute force | * Pengenalan Brute Force * Contoh kasus Perkalian Matriks * Contoh kasus Selection Sort * Contoh kasus Permutasi String | Menerangkan | Memerhatikan, mencatat, mengerjakan latihan. | Papan tulis, proyektor | Latihan | 1-7 | |
| 3 | 3 | Divide and Conquer | Mahasiswa memahami desain dan analisis algoritma yang berbasis pada teknik divide-and-conquer. | Mahasiswa mampu menganalis dan memberikan kesimpulan mengenai efisiensi algoritma berbasis divide-and-conquer. | * Pengenalan D&C * Binary Search * Merge Sort * Analysis Merge Sort * Quick Sort * Analisis Quick Sort | Menerangkan | Memerhatikan, mencatat, mengerjakan latihan. | Papan tulis, proyektor | Latihan | 1-7 | |
| 4 | 4 | Greedy / Djikstra | Mahasiswa memahami konsep kerja algoritma greedy | Mahasiswa mampu memahami implementasi algoritma greedy dalam memecahkan masalah. | * Pengenalan Greedy * Decision Tree * Knapsack Problem * Algoritma Djikstra | Menerangkan | Memerhatikan, mencatat, mengerjakan latihan. | Papan tulis, proyektor | Latihan | 1-7 | |
| 5 | 5, 6 | Dynamic Programming | Mahasiswa memahami konsep metode dynamic programming. | Mahasiswa memahami contoh-contoh masalah yang diselesaikan menggunakan metode dynamic programming. | * Pengenalan DP * Contoh-contoh DP | Menerangkan | Memerhatikan, mencatat, mengerjakan latihan. | Papan tulis, proyektor | Latihan | 1-7 | |
| 6 | 7, 8, 9 | Graph | Mahasiswa memahami lebih sepenuhnya konsep graph dalam pemrograman. | Mahasiswa memahami contoh-contoh analisis terhadap graph. | * Pengenalan Graph * Analisis Graph * Minimum Spanning Tree: Kruskal dan Prim * Single source shortest path * BFS dan DFS | Menerangkan | Memerhatikan, mencatat, mengerjakan latihan. | Papan tulis, proyektor | Latihan | 1-7 | |
| 7 | 10, 11, 12 | String Matching | Mahasiswa memahami konsep dari String Matching | Mahasiswa memahami contoh-contoh suffix array, suffix trie, algoritma string matching dan analisis string matching | * Pengenalan String Matching * Suffix Array * Suffix Trie * Algoritma Knuth Morris Pratt | Menerangkan | Memerhatikan, mencatat, mengerjakan latihan. | Papan tulis, proyektor | Latihan | 1-7 | |
| 8 | 13 | Competitive Programming | Mahasiswa memahami konsep pemrograman kompetitif dan soal pemrograman kompetitif | Mahasiswa memahami contoh-contoh soal pemrograman kompetitif | * Konsep Pemrograman Kompetitif * Pembahasan soal Pemrograman Kompetitif | Menerangkan | Memerhatikan, mencatat, mengerjakan latihan. | Papan tulis, proyektor | Latihan | 1-7 | |

**Buku:**

1. Wilton, Rex, Mathematics for Computer Students (Second Edition). Blackwell 1996. ISBN: 855546671
2. Sedgewick, Robert and Wayne, Kevin, Algorithms (Fourth Edition). Addison-Wesley, 2011. ISBN: 9780321573513
3. Cormen, Thomas. Introduction to Algorithms (Third Edition). The MIT Press, 2009. ISBN: 9780262533058
4. Skiena, Steven. The Algorithm Design Manual (Second Edition). Springer, 2008. ISBN: 9781848000698
5. Abelson, Harold, Sussman, Gerald Jay, Sussman, Julie, Structure and Interpretation of Computer Programs (Second Edition). The MIT Press, 1996. ISBN: 0262510871

**Paper:**

1. Hoare, C.R., An Axiomatic Basis for Computer Programming. Communications of the ACM Volume 12 No 10, 1969.
2. Floyd, Robert, Assigning Meanings to Programs. Symposia in Applied Mathematics, 1969.
3. Djikstra, E. W., Recursive Programming. Numerische Mathematik 2, 312-318, 1960.