



PROGRAM STUDI TEKNIK INFORMATIKA
Sekolah Teknik Elektro dan Informatika

INSTITUT TEKNOLOGI BANDUNG

Pengantar Strategi Algoritma

Bahan Kuliah IF2211 Strategi Algoritma

RINALDI M

Kampus ITB yang indah...



Foto oleh Eko Purwono (AR ITB)

Inilah STEI-ITB...



LabTek V, di sini Informatika ITB berada



Foto oleh Budi Rahardjo (EL ITB)

Salah satu mata kuliahnya....

IF2211 Strategi Algoritma



Apakah Strategi Algoritma itu?

Strategi algoritma (*algorithm strategies*) adalah:

- pendekatan umum
- untuk memecahkan **persoalan** secara **algoritmik**
- sehingga pendekatan tersebut dapat diterapkan pada berbagai persoalan komputasi [Levitin, 2003]
- Nama lain: *algorithm design techniques*

Persoalan (*Problem*)

- **Persoalan** (atau masalah): **pertanyaan** atau **tugas** (*task*) yang ingin kita cari jawabannya.

Contoh-contoh persoalan:

1. [**Persoalan pengurutan**] Diberikan senarai (*list*) S yang terdiri dari n buah *integer*. Urutkan n buah *integer* tersebut sehingga terurut secara menaik!

Jawaban: barisan nilai di dalam senarai yang terurut menaik.

2. [**Persoalan pencarian**] Apakah terdapat sebuah elemen bernilai x di dalam sebuah senarai S yang berisi n buah bilangan bulat?

Jawaban: “ya” jika x ditemukan di dalam senarai, atau “tidak” jika x tidak terdapat di dalam senarai.

- **Instansiasi persoalan:** parameter nilai yang diasosiasikan di dalam persoalan tersebut.
- Jawaban terhadap instansiasi persoalan dinamakan **solusi**
- Contoh: Selesaikan persoalan pengurutan untuk
 $S = [15, 4, 8, 11, 2, 10, 19]$ $n = 7$

Solusi: $S = [2, 4, 8, 10, 11, 15, 19]$.

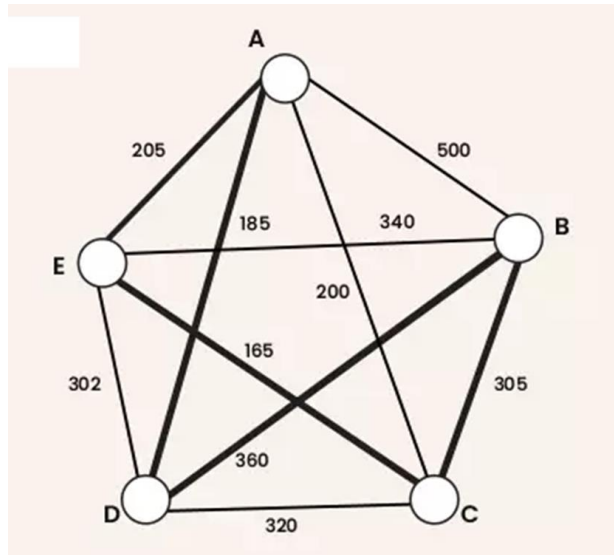
Algoritma

- Untuk persoalan dengan instansiasi yang besar, solusinya menjadi lebih sulit ditentukan.
- Perlu sebuah prosedur umum yang berisi langkah-langkah penyelesaian persoalan → **algoritma**
- **Algoritma**: urutan langkah-langkah yang sistematis untuk memecahkan suatu persoalan, yang memproses masukan menjadi luaran.

Beberapa Contoh Persoalan Klasik

1. *Travelling Salesperson Problem (TSP)*

Persoalan: Diberikan n buah kota serta diketahui jarak antara setiap kota satu sama lain. Temukan perjalanan (*tour*) terpendek yang dimulai dari sebuah kota dan melalui setiap kota lainnya hanya sekali dan kembali lagi ke kota asal keberangkatan.



Tur terpendek:

A – D – B – C – E – A

dengan total bobot:

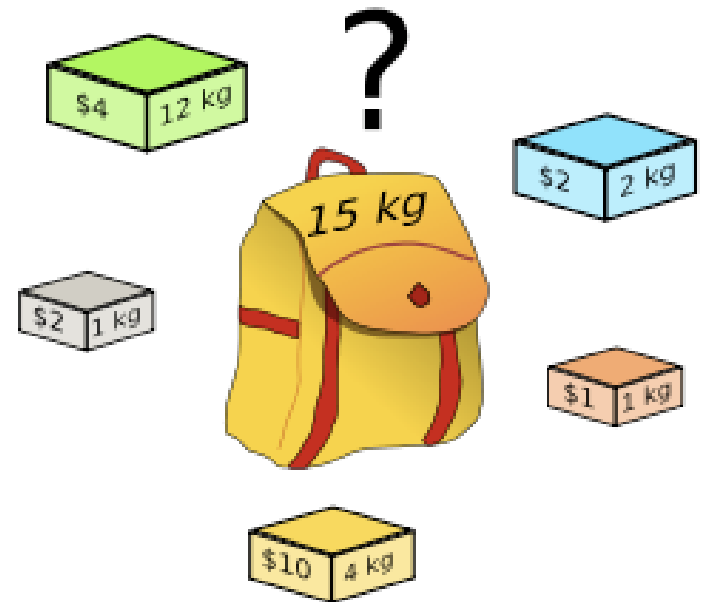
$$185 + 360 + 305 + 165 + 205 = 1220$$



2. *Integer Knapsack Problem*

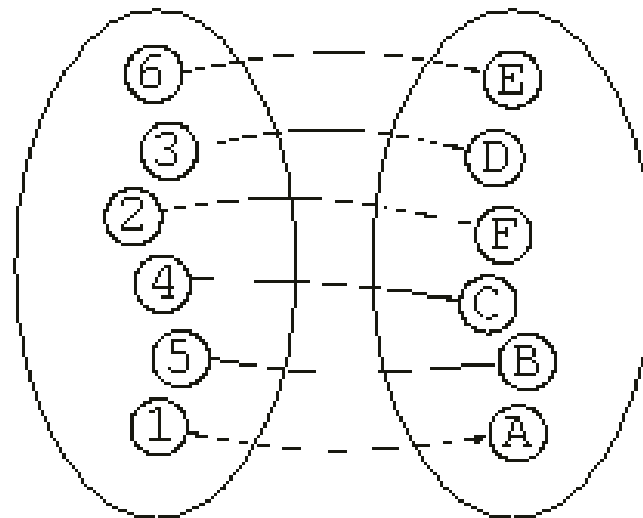
Persoalan: Diberikan n buah objek dan sebuah *knapsack* (karung, tas, ransel, dsb) dengan kapasitas bobot K . Setiap objek memiliki properti bobot (*weight*) w_i dan keuntungan (*profit*) p_i .

Bagaimana cara memilih objek-objek yang dimasukkan ke dalam *knapsack* sedemikian sehingga tidak melebihi kapasitas *knapsack* namun memberikan keuntungan maksimal?



3. Persoalan penugasan (*assignment problem*)

Misalkan terdapat n orang dan n buah pekerjaan (*job*). Setiap orang akan di-*assign* dengan sebuah pekerjaan. Penugasan orang ke- i dengan pekerjaan ke- j membutuhkan biaya sebesar $c(i, j)$. Bagaimana melakukan penugasan sehingga total biaya penugasan adalah seminimal mungkin?



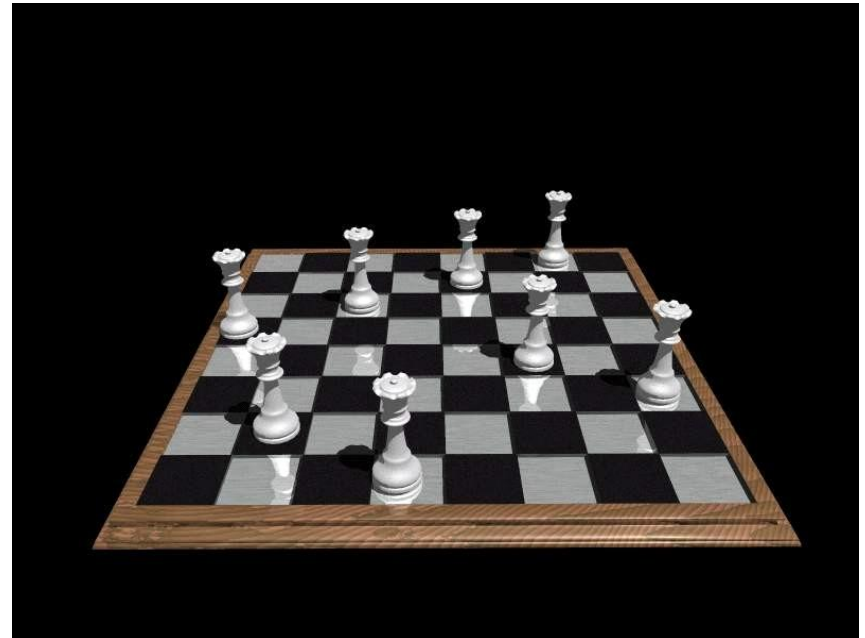
Contoh instansiasi persoalan:

$$C = \begin{bmatrix} \textit{Job 1} & \textit{Job 2} & \textit{Job 3} & \textit{Job 4} \\ 9 & 2 & 7 & 8 \\ 6 & 4 & 3 & 7 \\ 5 & 8 & 1 & 4 \\ 7 & 6 & 9 & 4 \end{bmatrix} \begin{matrix} \text{Orang } a \\ \text{Orang } b \\ \text{Orang } c \\ \text{Orang } d \end{matrix}$$



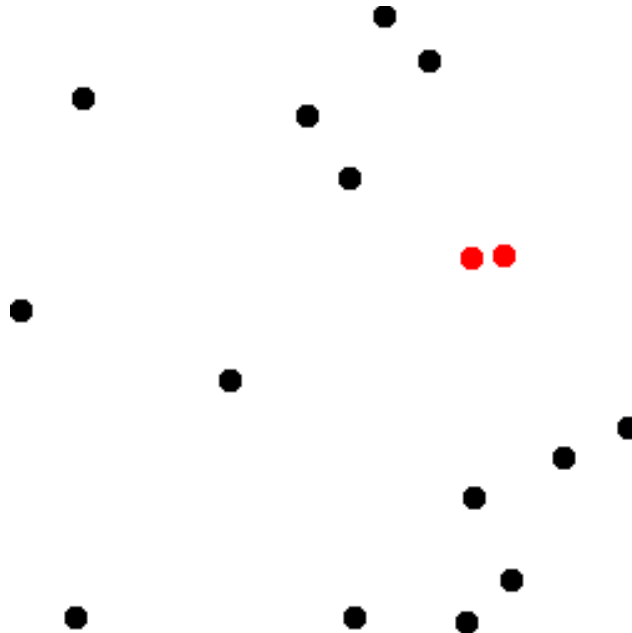
4. Persoalan N-Ratu (*The N-Queens Problem*)

Persoalan: Diberikan sebuah papan catur yang berukuran $N \times N$ dan N buah bidak ratu. Bagaimana menempatkan N buah ratu (Q) itu pada petak-petak papan catur sedemikian sehingga tidak ada dua ratu atau lebih yang terletak pada satu baris yang sama, atau pada satu kolom yang sama, atau pada satu diagonal yang sama ?



5. Mencari Pasangan Titik Terdekat (*Closest Pair*)

Persoalan: Diberikan n buah titik, tentukan dua buah titik yang terdekat satu sama lain.



6. Permainan *15-Puzzle*

Persoalan: Diberikan sebuah *15-puzzle* yang memuat 15 buah ubin (*tile*) yang diberi nomor 1 sampai 15, dan satu buah slot kosong yang digunakan untuk menggerakkan ubin ke atas, ke bawah, ke kiri, dan ke kanan. Misalkan diberikan keadaan awal dan keadaan akhir susunan ubin. Bagaimana langkah-langkah untuk mentransformasikan susunan awal menjadi susunan akhir?



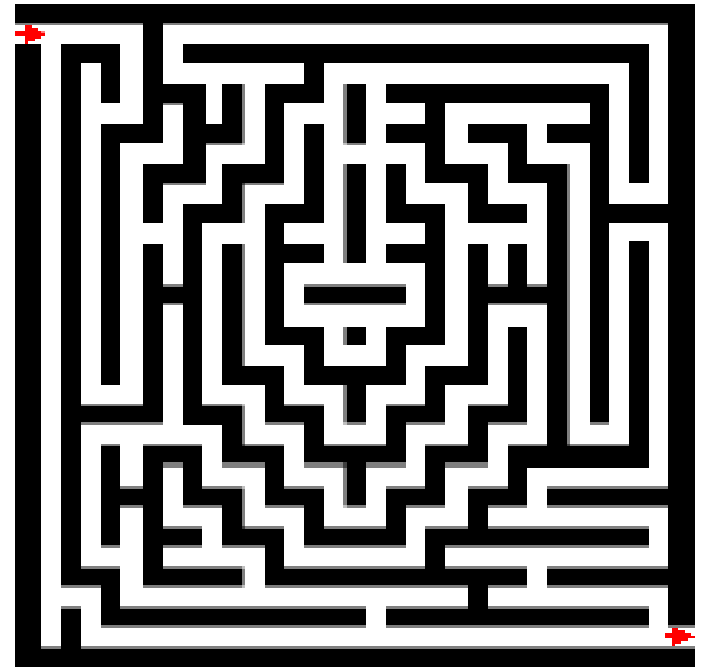
(a) Susunan awal



(b) Susunan akhir

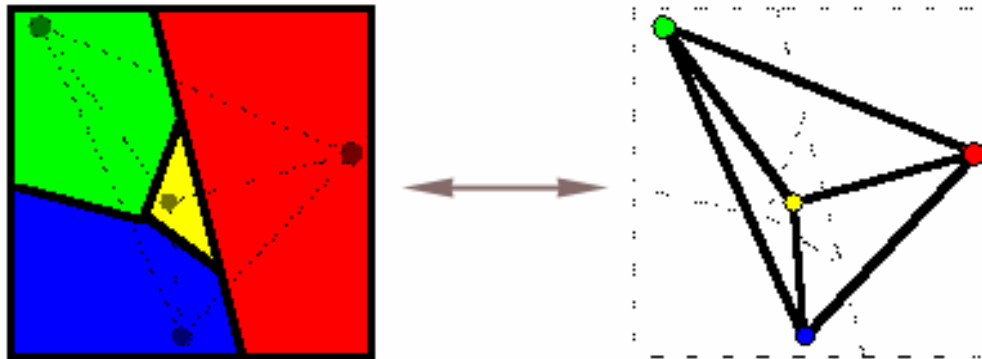
7. *Menemukan jalan keluar dari labirin (Maze Problem)*

Persoalan: Diberikan sebuah labirin dengan satu atau lebih pintu masuk dan satu atau lebih pintu keluar. Temukan jalan yang harus dilalui sehingga seseorang dapat keluar dengan selamat dari labirin tersebut (tidak tersesat di dalamnya).



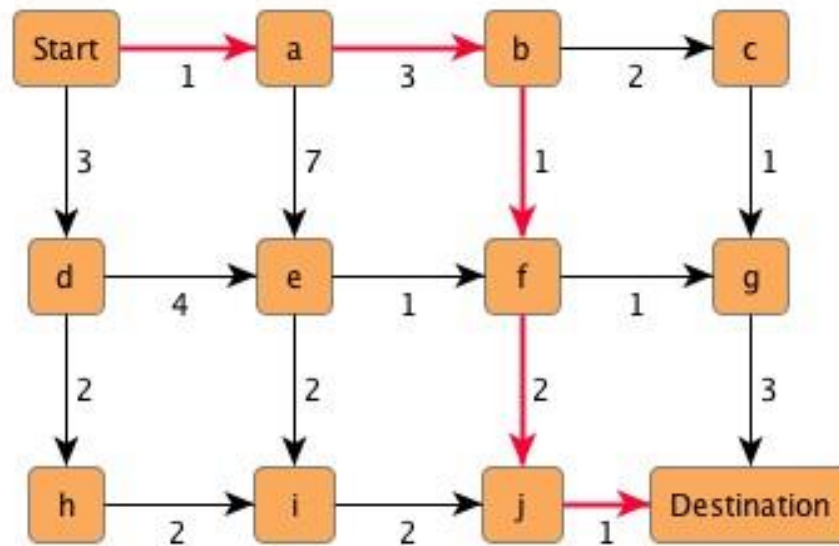
8. Pewarnaan Graf (*Graph Colouring*)

Persoalan: Diberikan sebuah graf G dengan n buah simpul dan disediakan m buah warna. Bagaimana mewarnai seluruh simpul graf G sedemikian sehingga tidak ada dua buah simpul bertetangga yang mempunyai warna sama? (Perhatikan juga bahwa tidak seluruh warna harus dipakai)



9. Lintasan terpendek (*shortest path*)

Persoalan: Diketahui n buah kota dan diberikan jarak antara dua buah kota yang bertetangga. Tentukan lintasan terpendek dari sebuah kota asal ke sebuah kota tujuan.



10. *Partition Problem*

Persoalan: Diberikan n buah bilangan bulat sembarang (mungkin ada elemen berulang). Bagaimana cara membaginya (mem-partisi) menjadi beberapa buah upa-himpunan (*subset*) sehingga jumlah seluruh nilai di dalam setiap upa-himpunan sama?

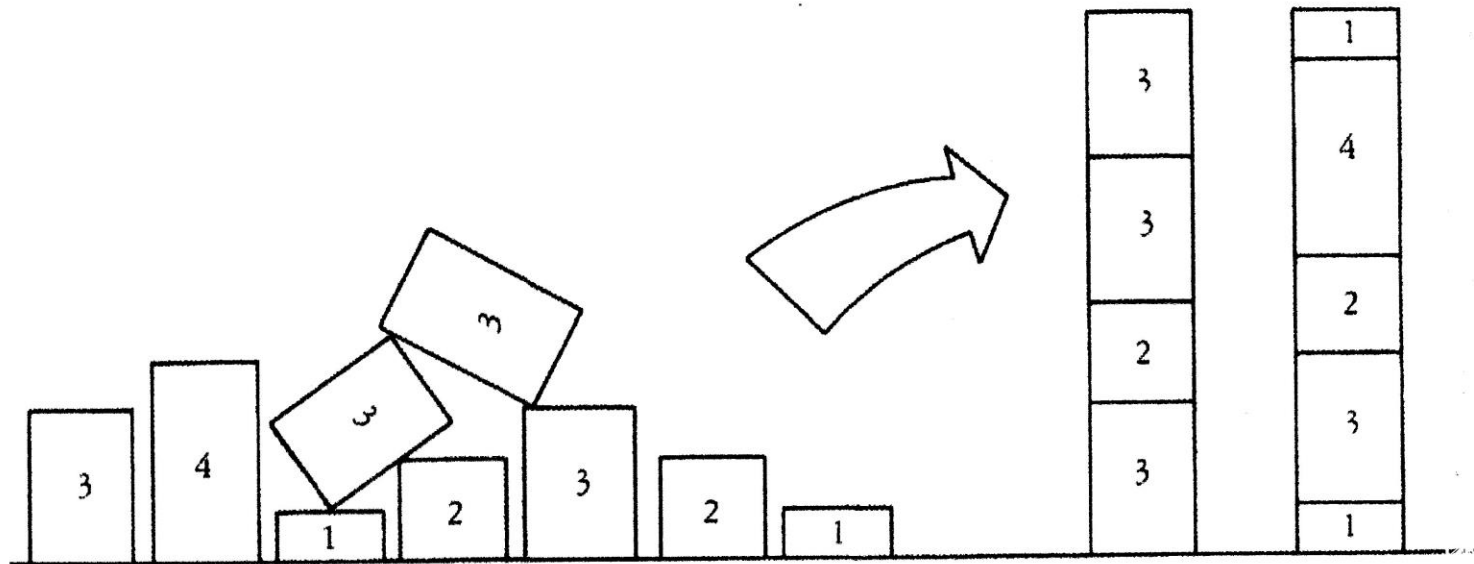


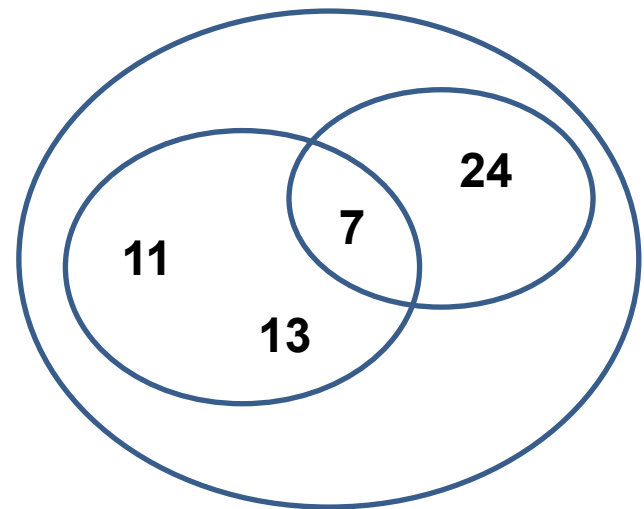
Figure 30.1 An example of the partition problem

11. Sum of Subsets Problem

- **Persoalan:** Diberikan n buah bobot (*weight*) berupa bilangan-bilangan positif (*integer*) yang berbeda $w_1, w_2, \dots, w_n$ dan sebuah bilangan bulat positif m . Tentukan semua himpunan bagian dari n bobot tersebut yang jumlahnya sama dengan m .

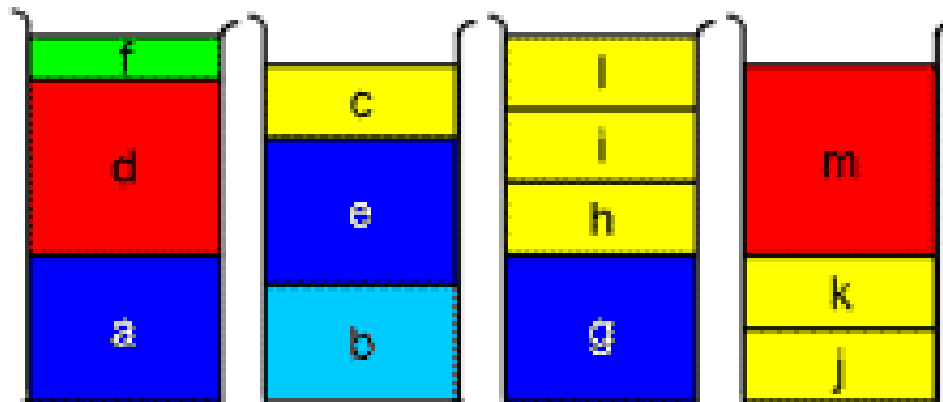
Contoh: $n = 4; (w_1, w_2, w_3, w_4) = (11, 13, 24, 7), m = 31$.

Solusi: $\{11, 13, 7\}$ dan $\{24, 7\}$.



12. Bin Packing Problem

Persoalan: Terdapat sejumlah kardus masing-masing dengan kapasitas C dan n barang berukuran $S_1, S_2, \dots, S_n$. Kemaslah n barang ke dalam M buah kardus sedemikian sehingga ukuran total barang di dalam setiap kardus tidak melebihi C . Temukan minimal M yaitu paling sedikit jumlah kardus untuk menampung n barang.



Analisis Algoritma

- Analisis algoritma bertujuan untuk mengukur kinerja (*performance*) algoritma dari sisi kesangkilannya (*efficient*)
- Parameter untuk mengukur kesangkilan algoritma:
 1. Kompleksitas waktu, $T(n)$
 2. Kompleksitas ruang, $S(n)$

n = ukuran masukan yang diproses oleh algoritma

- $T(n)$: jumlah tahapan komputasi yang dilakukan di dalam suatu algoritma sebagai fungsi dari ukuran masukan n .

Tahapan komputasi: operasi-operasi yang terdapat di dalam algoritma (operasi perbandingan, operasi aritmetika, pengaksesan larik, dll)

- $S(n)$: ruang memori yang dibutuhkan algoritma sebagai fungsi dari ukuran masukan n .
- Tiga notasi kompleksitas waktu asimptotik algoritma:
 1. $O(f(n))$: batas lebih atas kebutuhan waktu algoritma
 2. $\Omega(g(n))$: batas lebih bawah kebutuhan waktu algoritma
 3. $\Theta(h(n))$: jika dan hanya jika $O(h(n)) = \Omega(h(n))$

Strategi Algoritma

Beberapa strategi algoritma yang dipelajari di dalam kuliah ini:

1. Algoritma *Brute-Force*
2. Algoritma *Greedy*
3. Algoritma *Divide and Conquer*
4. Algoritma *Decrease and Conquer*
5. Algoritma *Backtracking*
6. Algoritma *Branch and Bound*
7. *Dynamic programming*

Mengapa Perlu Mempelajari Strategi Algoritma?

- Ada dua alasan (Levitin, 2003):
 1. Memberikan panduan (*guidance*) untuk merancang algoritma bagi persoalan baru.
 2. Dapat mengklasifikasikan algoritma berdasarkan gagasan perancangan yang mendasarinya.

Klasifikasi Strategi Algoritma:

1. Strategi solusi langsung (*direct solution strategies*)
 - Algoritma *Brute Force*
 - Algoritma *Greedy*
2. Strategi berbasis pencarian pada ruang status (*state-space base strategies*)
 - Algoritma *Backtracking*
 - Algoritma *Branch and Bound*

3. Strategi solusi atas-bawah (*top-down solution strategies*)
 - Algoritma *Divide and Conquer*.
 - Algoritma *Decrease and Conquer*
 - *Dynamic Programming*.
4. Strategi solusi bawah-atas (*bottom-up solution strategies*)
 - *Dynamic Programming*.

Pokok Bahasan Kuliah IF2211

1. Algoritma *brute force*
2. Algoritma *greedy*
3. Algoritma *divide and conquer*
4. Algoritma *decrease and conquer*
5. *DFS* dan *BFS*
6. Algoritma *backtracking*
7. Algoritma *branch and bound*
8. Algoritma *A**, *Best First Search*, dan *UCS*
9. Program dinamis (*dynamic programming*)
10. *String matching + regular expression (regex)*
11. Teori P, NP, dan NP-Completes

Buku Referensi Kuliah

1. Anany Levitin, *Introduction to the Design & Analysis of Algorithms*, Addison-Wesley, 2003.
2. Bhardwaj, Anuj; Verma, Parag, *Design and Analysis of Algorithm*, Alpha Science International Ltd., 2017, can be accessed at:
<http://portal.igpublish.com/iglibrary/obj/APSB0000252?dtbs=&searchid=1573133164973plapRvm7MQJMAwkcwThUs> from
<https://lib.itb.ac.id/>
3. Khan Academy, *Computer Science: Algorithm*, can be accessed at
<https://www.khanacademy.org/computing/computer-science/algorithms>
4. Coursera, *Data Structures and Algorithms Specialization*, 2019, can be accessed at: <https://www.coursera.org/learn/algorithmic-toolbox?specialization=data-structures-algorithms>

5. Stuart J Russell & Peter Norvig, *Resources of topics in Artificial Intelligence: A Modern Approach*, 3rd Edition, Global Edition Paperback, Pearson, 2016, <http://aima.cs.berkeley.edu/>
6. Brandons Kerritt, *All You Need to Know About Big O Notation* [Python Examples], 2019, can be accessed at <https://dev.to/brandonskerritt/all-you-need-to-know-about-big-o-notation-python-examples-2k4o>
7. Coursera, *Machine Learning Clustering and Retrieval: Complexity of Brute Force Algorithm*, 2019, can be accessed at <https://www.coursera.org/lecture/ml-clustering-and-retrieval/complexity-of-brute-force-search-5R6q3>
8. Rinaldi Munir, Diktat kuliah IF2251 Strategi Algoritmik, Teknik Informatika ITB

9. Marin Vlastelica Pogančić, *The Branch and Bound Algorithm*, 2019, can be accessed at <https://towardsdatascience.com/the-branch-and-bound-algorithm-a7ae4d227a69>
10. Jurafsky and James H. Martin, *Speech and Language Processing*, 2019, can be accessed at <https://web.stanford.edu/~jurafsky/slp3/>
11. Chua Hock-Chuan, *Regular Expressions (Regex)*, 2018, can be accessed at <https://www.ntu.edu.sg/home/ehchua/programming/howto/Regexe.html>
12. Ellis Horowitz & Sartaj Sahni, *Computer Algorithms*, Computer Science Press, 1998.
13. Richard E. Neapolitan, *Foundations of Algorithms*, D.C. Heath and Company, 1996
14. Thomas H. Cormen, *Introduction to Algorithms*, The MIT Press, 1992.

Tujuan kuliah IF2211

Tujuan umum: Mahasiswa memahami beberapa strategi algoritma yang digunakan dalam memecahkan persoalan secara sangkil sesuai dengan karakteristik persoalan tersebut.

Tujuan khusus: Mahasiswa mampu:

1. memilih strategi algoritma yang tepat untuk menyelesaikan suatu persoalan.
2. membuat program/aplikasi untuk menyelesaikan persoalan dengan strategi algoritma yang tepat.

Capaian Pembelajaran Mata Kuliah (CPMK)

- **CPMK1.** Menyusun solusi masalah komputasi dengan berbagai strategi algoritma, bergantung pada karakteristik persoalan, kemudian mendesain algoritma penyelesaian masalah
- **CPMK2.** Mengimplementasikan desain algoritma menjadi sebuah program aplikasi
- **CPMK3.** Menulis makalah ilmiah yang berisi hasil studi dan implementasi penyelesaian masalah dengan menggunakan berbagai strategi algoritma.
- **CPMK4.** Menulis laporan penyelesaian masalah komputasi yang diberikan di dalam tugas
- **CPMK5.** Mendemokan aplikasi untuk mendapatkan feedback

Prasyarat Kuliah

1. IF1220 Matematika Diskrit
2. IF2110 Algoritma dan Pemrograman 2

Penilaian Kuliah

1. Ujian Tengah Semester (UTS) – 1 kali
2. Ujian Akhir Semester (UAS) – 1 kali
3. Tugas Besar (Tubes) – 3 kali, per kelompok
4. Tugas Kecil (Tucil) – 3 kali, per orang/2 oran
5. Makalah (pada akhir semester) – 1 kali
6. Kehadiran (minimal 80%, kurang dari 80% tidak boleh ikut UAS)

Perihal Tubes dan Tucil

- Tugas pemrograman adalah tugas besar berkelompok. Program dikembangkan dengan menggunakan kakas pemrograman berbasis GUI (*Graphical User Interface*).
- Tugas pertama direncanakan menggunakan kakas pemrograman berbasis GUI menggunakan bahasa C++ (misalnya Visual C++).
- Tugas kedua menggunakan C# dengan kakas Visual Studio atau yang sejenis
- Tugas ketiga menggunakan Java (NetBeans, Eclipse) + PHP.
- Tugas Kecil adalah tugas perorangan atau per-dua orangan dengan bahasa pemrograman yang ditentukan (C, C++, Java, Python, Golang, Rust, dll).

Contoh-contoh Tupil dan Tubes

Tugas Kecil 1 IF2211 Strategi Algoritma
Semester II tahun 2024/2025
Penyelesaian IQ Puzzler Pro dengan Algoritma Brute Force

Batas pengumpulan : Hari Senin, 24 Februari 2025 pukul 13.00 WIB

Arsip pengumpulan :

- Source program yang dapat dijalankan disertai README
- Laporan (*soft copy*)

Deskripsi Tugas



Gambar 1 Permainan IQ Puzzler Pro
(Sumber: <https://www.smartgamesusa.com>)

IQ Puzzler Pro adalah permainan papan yang diproduksi oleh perusahaan Smart Games. Tujuan dari permainan ini adalah pemain harus dapat mengisi seluruh papan dengan piece (blok puzzle) yang telah tersedia.

Tugas Kecil 2 IF2211 Strategi Algoritma
Semester II tahun 2024/2025
Kompresi Gambar Dengan Metode Quadtree

Batas pengumpulan : Hari Jumat, 11 April 2025 pukul 23.59 WIB

Arsip pengumpulan :

- Source program yang dapat dijalankan disertai README
- Laporan (*soft copy*)

Deskripsi Tugas:



Gambar 1. Quadtree dalam Kompresi Gambar

(Sumber: <https://medium.com/@tannerwyork/quadtrees-for-image-processing-302536c95c00>)

Quadtree adalah struktur data hierarkis yang digunakan untuk membagi ruang atau data menjadi bagian yang lebih kecil, yang sering digunakan dalam pengolahan gambar. Dalam konteks kompresi gambar, Quadtree membagi gambar menjadi blok-blok kecil berdasarkan keseragaman warna atau intensitas piksel. Prosesnya dimulai dengan membagi gambar menjadi empat bagian, lalu memeriksa apakah setiap bagian memiliki nilai yang seragam berdasarkan

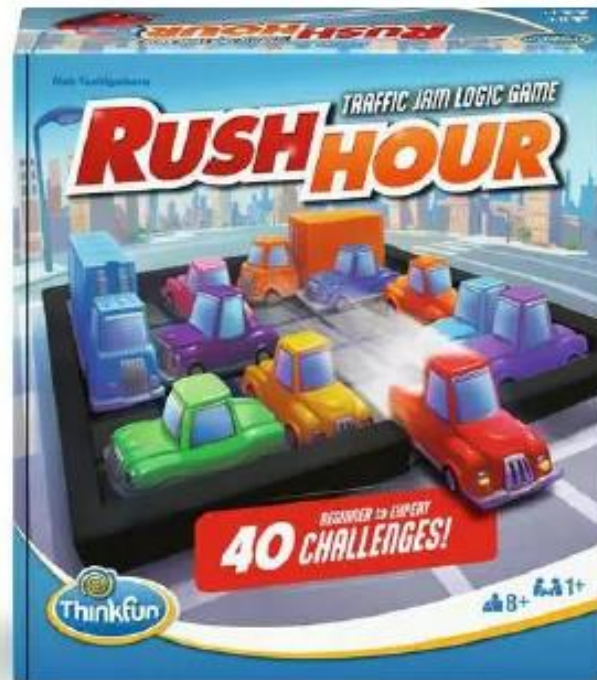
Tugas Kecil 3 IF2211 Strategi Algoritma
Semester II tahun 2024/2025
Penyelesaian Puzzle Rush Hour Menggunakan Algoritma Pathfinding

Batas pengumpulan : Hari Rabu, 21 Mei 2025 pukul 12.00 WIB

Arsip pengumpulan :

- Source program yang dapat dijalankan disertai README
- Laporan (*soft copy*)

Deskripsi Tugas:



Gambar 1. Rush Hour Puzzle

(Sumber: <https://www.thinkfun.com/en-US/products/educational-games/rush-hour-76582>)

Rush Hour adalah sebuah permainan puzzle logika berbasis grid yang menantang pemain untuk menggeser kendaraan di dalam sebuah kotak (biasanya berukuran 6x6) agar mobil utama (biasanya berwarna merah) dapat keluar dari kemacetan melalui pintu keluar di sisi papan.

Tugas Besar 1 IF2211 Strategi Algoritma
Semester II tahun 2024/2025

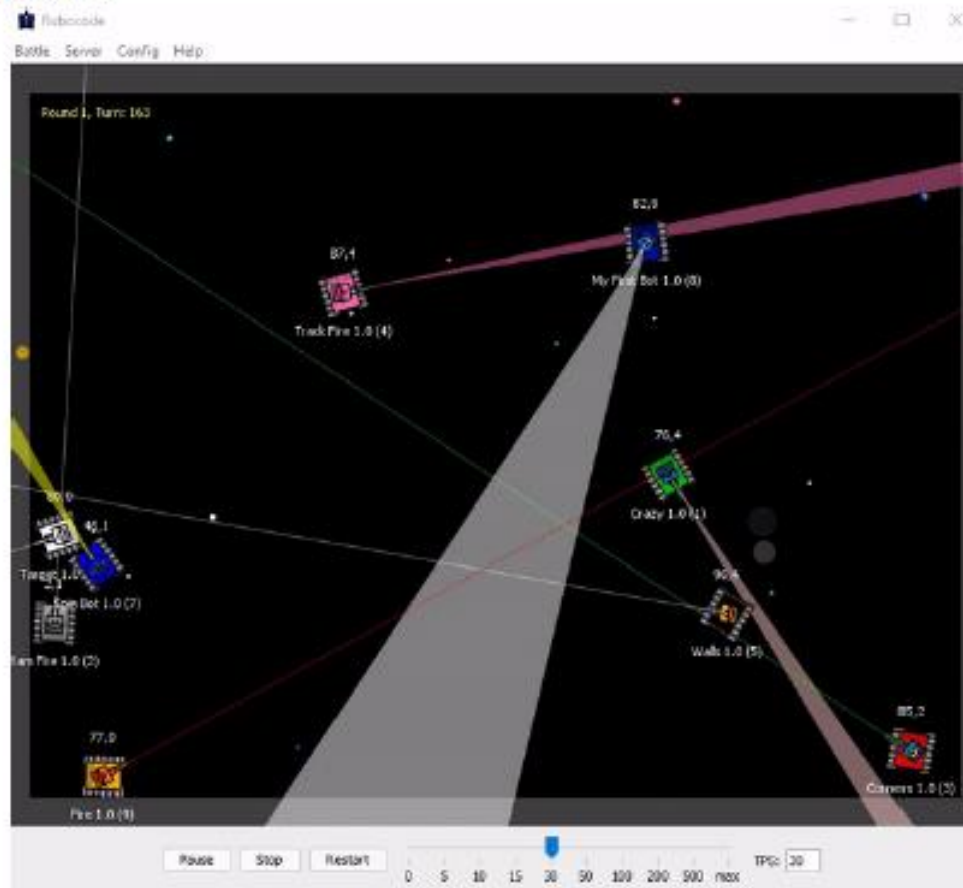
Pemanfaatan Algoritma *Greedy* dalam pembuatan *bot* permainan Robocode Tank Royale

Batas pengumpulan : Hari Senin, 24 Maret 2025 pukul 22.11 WIB

Arsip pengumpulan :

- *Source* program yang dapat dijalankan disertai README
- Laporan (*soft copy*)

Deskripsi Tugas:



Gambar 1 Robocode Tank Royale

Little Alchemy 2

Batas pengumpulan : Hari Senin, 12 Mei 2025 pukul 22.11 WIB

Arsip pengumpulan :

- *Source program* yang dapat dijalankan disertai README
- Laporan (*soft copy*)

Deskripsi Tugas:



Gambar 1. Little Alchemy 2
(sumber: <https://www.thegamer.com>)

Little Alchemy 2 merupakan permainan berbasis web / aplikasi yang dikembangkan oleh Recloak yang dirilis pada tahun 2017, permainan ini bertujuan untuk membuat 720 elemen dari 4 elemen dasar yang tersedia yaitu *air*, *earth*, *fire*, dan *water*. Permainan ini merupakan sekuel dari permainan sebelumnya yakni Little Alchemy 1 yang dirilis tahun 2010.

**Pemanfaatan *Pattern Matching* untuk Membangun Sistem ATS (*Applicant Tracking System*)
Berbasis CV Digital**

Batas pengumpulan : Hari Minggu, 15 Juni 2025 pukul 22.11 WIB

Arsip pengumpulan :

- *Source* program yang dapat dijalankan disertai README
- Laporan (*soft copy*)

Deskripsi Tugas



Gambar 1. CV ATS dalam Dunia Kerja
(Sumber: <https://www.antaranews.com/>)

Di era digital ini, keamanan data dan akses menjadi semakin penting. Perkembangan proses rekrutmen tenaga kerja telah mengalami perubahan signifikan dengan memanfaatkan teknologi untuk meningkatkan efisiensi dan akurasi. Salah satu inovasi yang menjadi solusi utama adalah

SELAMAT BELAJAR