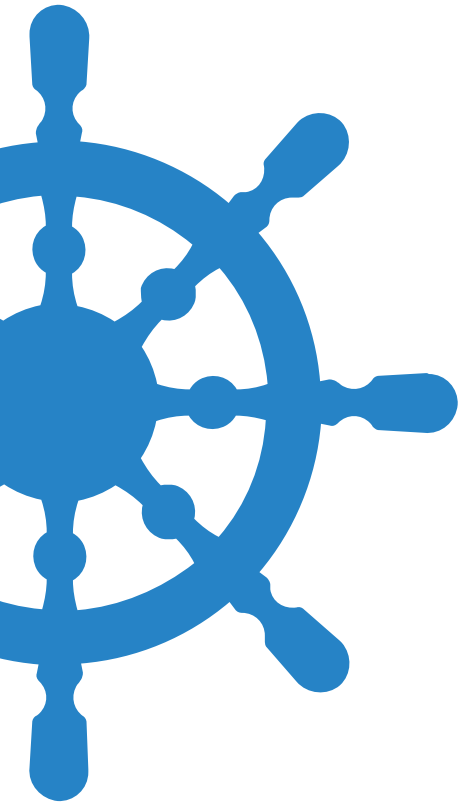
GuruVirtual.ID

Algoritma, Struktur Data, dan Implementasi dalam Perangkat Lunak

Hartono, S.Pd., M.T.I

Universitas Muhammadiyah Kotabumi

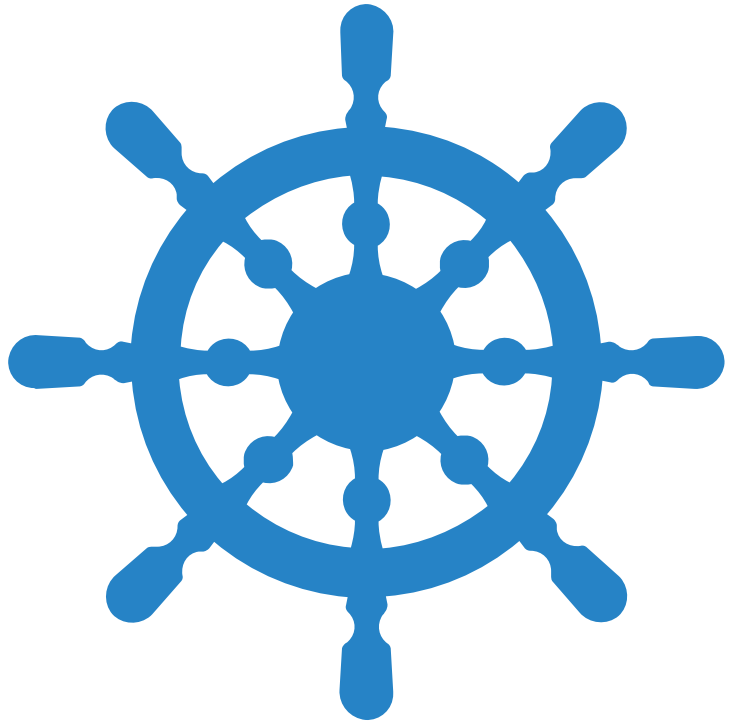
Apa itu Algoritma?



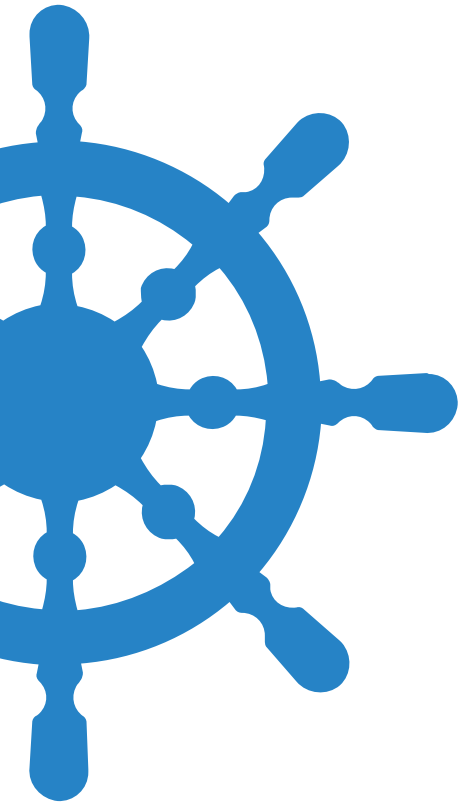
- Algoritma adalah serangkaian langkah-langkah terstruktur yang dirancang untuk menyelesaikan suatu masalah atau tugas tertentu.
- Algoritma memberikan petunjuk tentang urutan tindakan yang harus diambil untuk mencapai hasil yang diinginkan.
- Algoritma secara singkat dapat disebut sebagai serangkaian Langkah untuk menyelesaikan masalah atau tugas tertentu.

Algoritma Memasak

- Bayangkan algoritma sebagai suatu resep dalam memasak.
- Seperti resep, algoritma juga memiliki urutan langkah-langkah yang harus diikuti dengan cermat untuk mencapai hasil yang baik.
- Pada dasarnya, seperti cara resep memberi tahu Anda bagaimana mengolah bahan-bahan menjadi hidangan lezat, algoritma memberi petunjuk bagaimana mengolah data atau langkah-langkah menjadi solusi.



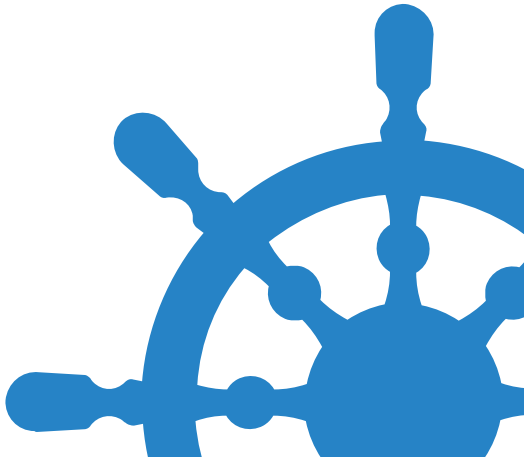
Algoritma Membuat Secangkir Teh



- Didihkan Air
- Siapkan Teh
- Tuangkan Air Panas
- Rendam Teh
- Tambahkan Gula atau Susu (Opsional)
- Aduk dan Nikmati

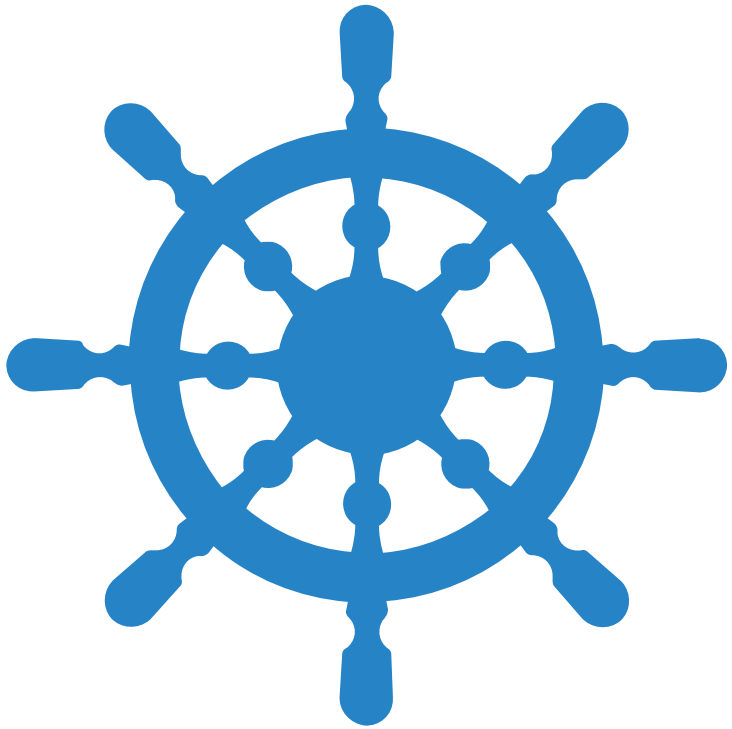
Algoritma Membuat Secangkir Teh (Lebih Spesifik)

1. Didihkan Air:
   - Isi teko dengan air.
   - Panaskan teko hingga air mendidih.
2. Siapkan Teh:
   - Letakkan kantong teh di dalam cangkir.
3. Tuangkan Air Panas:
   - Tuangkan air mendidih ke dalam cangkir dengan kantong teh.
4. Rendam Teh:
   - Biarkan kantong teh terendam dalam air panas selama beberapa meni.
5. Tambahkan Gula atau Susu (Opsional):
   - Jika diinginkan, tambahkan gula atau susu ke dalam teh sesuai selera.
6. Aduk dan Nikmati:
   - Aduk teh dengan sendok dan nikmati secangkir teh yang lezat!

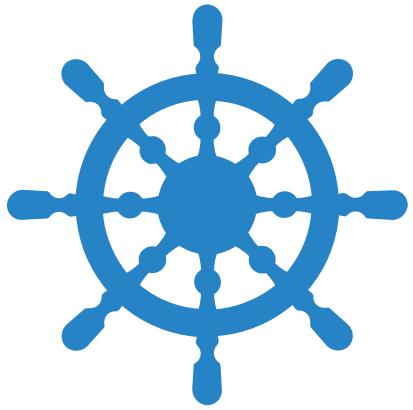


Ciri-ciri Algoritma

1. Algoritma harus berhenti setelah mengerjakan sejumlah langkah terbatas.
2. Setiap langkah harus didefinisikan dengan tepat dan tidak berarti-dua (Ambiguitas).
3. Algoritma memiliki masukkan.
4. Algoritma memiliki satu atau lebih keluaran.
5. Algoritma harus efektif (setiap langkah harus sederhana sehingga dapat dikerjakan dalam waktu yang masuk akal).



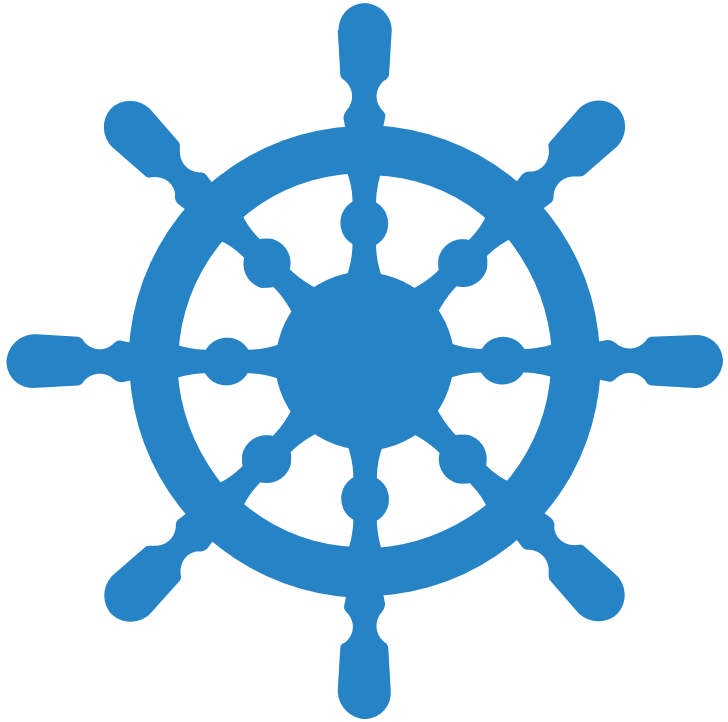
Sejarah Algoritma



1. "Algoritma" berasal dari Muhammad ibn Musa al-Khwarizmi, matematikawan Persia abad ke-9.
2. Al-Khwarizmi: matematikawan Muslim, "Buku Ringkas" tentang aljabar & penyelesaian masalah.
3. Kata "algorithm" muncul dalam terjemahan Latin karya al-Khwarizmi, terkait langkah-langkah matematika.
4. Algoritma: langkah-langkah sistematis selesai masalah; diperluas ke ilmu komputer.
5. Al-Khwarizmi kontribusi besar dalam matematika, kata "algoritma" mencuat dari karyanya.

Data

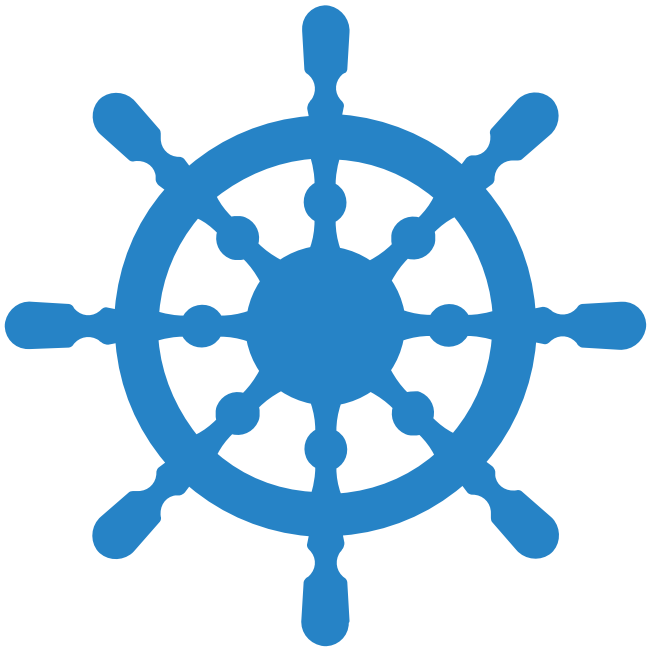
- Data adalah kumpulan fakta, informasi, atau nilai yang diperoleh dari pengamatan, pengukuran, atau sumber lainnya.
- Data dapat berupa angka, teks, gambar, suara, atau bentuk lainnya yang merepresentasikan suatu keadaan, peristiwa, atau objek.
- Data merupakan bahan mentah yang menjadi dasar untuk analisis, pengolahan, dan pengambilan keputusan.
- Dalam konteks komputer dan teknologi informasi, data adalah informasi yang dapat diproses oleh sistem komputer untuk menghasilkan hasil yang bermakna.



Struktur Data

- Struktur data adalah cara yang terorganisir dan terstruktur untuk menyimpan, mengatur, dan mengelola data dalam komputer atau sistem informasi.
- Struktur data menyediakan kerangka kerja untuk mengatur data dalam bentuk yang memungkinkan akses, manipulasi, dan pengolahan yang efisien.
- Tujuannya adalah untuk mengoptimalkan penggunaan ruang penyimpanan serta memungkinkan operasi data seperti pencarian, penyisipan, penghapusan, dan pengurutan dilakukan dengan efisien.

Data dan Struktur Data

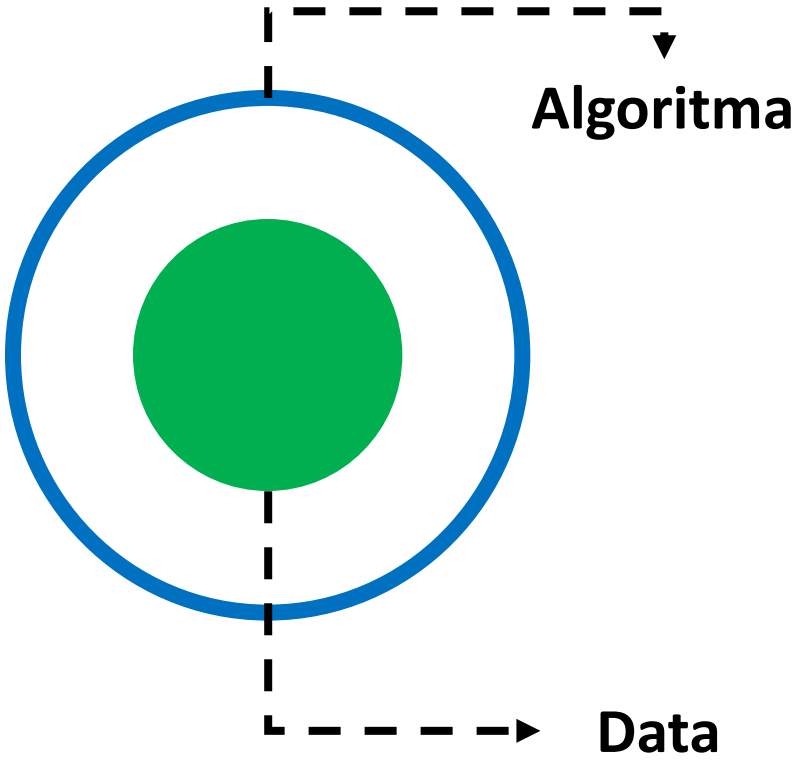


- Data adalah informasi mentah atau fakta yang dapat berupa angka, teks, gambar, suara, atau bentuk lainnya.
- Data adalah bahan dasar yang dibutuhkan untuk diolah, dianalisis, atau disajikan.
- Data tidak memiliki organisasi atau susunan tertentu, dan belum tentu memiliki makna tanpa konteks.
- Contoh data: angka 10, "Hello, world!", gambar matahari, rekaman suara burung bernyanyi.

… Data dan Struktur Data

- Struktur data adalah cara terstruktur untuk mengatur dan mengelola data.
- Struktur data memberikan organisasi pada data dengan mengatur bagaimana data dihubungkan satu sama lain dan bagaimana akses ke data tersebut dilakukan.
- Struktur data memungkinkan operasi tertentu seperti penyisipan, penghapusan, dan pencarian data dilakukan dengan efisien.
- Contoh struktur data: array, linked list, stack, queue, dll

Asosiasi Antara Algoritma dan Struktur Data



- Algoritma dapat diibaratkan sebagai “cara untuk mengolah” dan data adalah “bahan yang diolah”;
- Hubungan ini mempengaruhi kinerja, efisiensi, dan hasil akhir dari suatu solusi permasalahan yang dihadapi.

Kesimpulan Secara Umum

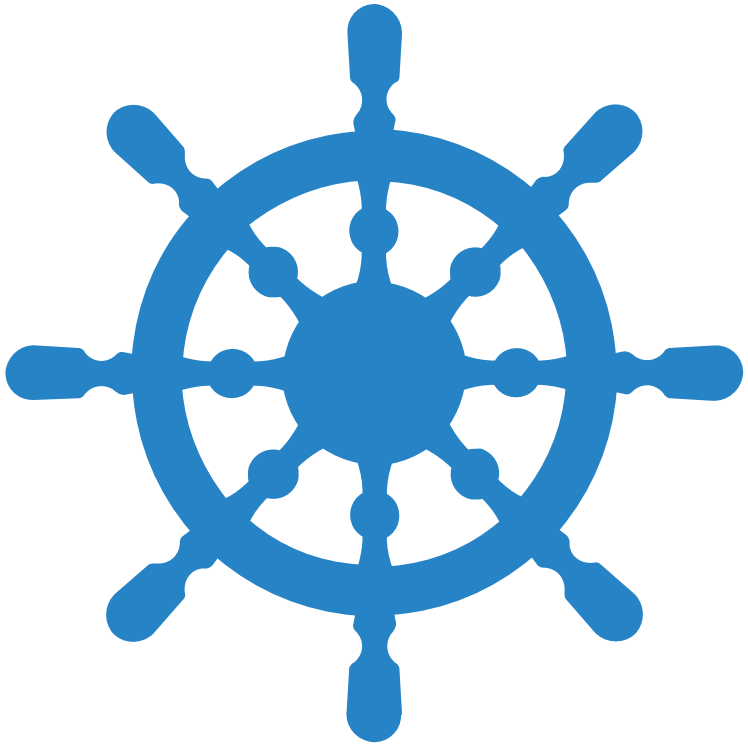
- Data adalah sesuatu/objek mentah yang dapat diolah atau dibentuk oleh algoritma;
- Struktur data adalah hasil dari olahan algoritma;
- Algoritma dapat mengolah struktur data untuk tujuan-tujuan lain yang berkaitan dengan penggunaan data;
- Algoritma dan struktur data adalah hal yang sangat berkaitan karena berhubungan dengan efektivitas dan efisiensi dalam melakukan suatu proses.

Kesimpulan Berdasarkan Definisi dan Fokus

- Algoritma: Sekumpulan langkah-langkah yang terstruktur untuk menyelesaikan masalah atau mencapai tujuan tertentu.
- Struktur Data: Cara terorganisir untuk menyimpan, mengelola, dan mengakses data dalam komputer.

Kesimpulan Berdasarkan Fungsi Utama

- Algoritma: Menentukan cara untuk melakukan tugas tertentu, mengolah data, dan mencapai hasil yang diinginkan.
- Struktur Data: Memungkinkan pengaturan data dalam bentuk yang efisien untuk operasi seperti penyisipan, penghapusan, dan pencarian.



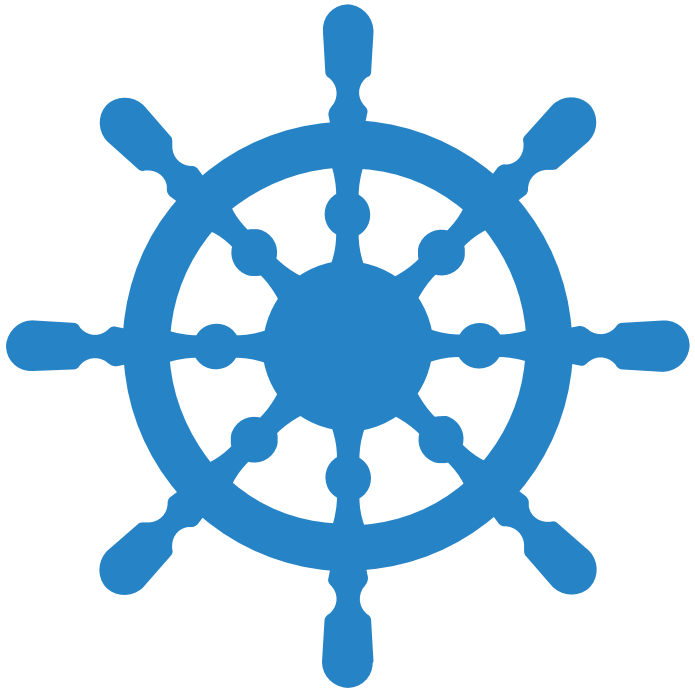
Kesimpulan Berdasarkan Aplikasi

- Algoritma: Berfokus pada pemecahan masalah dan pengolahan data dengan langkah-langkah logis.
- Struktur Data: Berfokus pada pengorganisasian data agar dapat diakses dan dikelola dengan baik.

Asosiasi Algoritma dan Data

- **Efisiensi Eksekusi Algoritma:** Pemilihan struktur data yang tepat dapat mempengaruhi efisiensi waktu dan sumber daya yang dibutuhkan oleh algoritma.
- Misalnya, algoritma pencarian linier dan algoritma pencarian biner pada array terurut memiliki perbedaan signifikan dalam efisiensi, tergantung pada struktur data yang digunakan.

… Asosiasi Algoritma dan Data



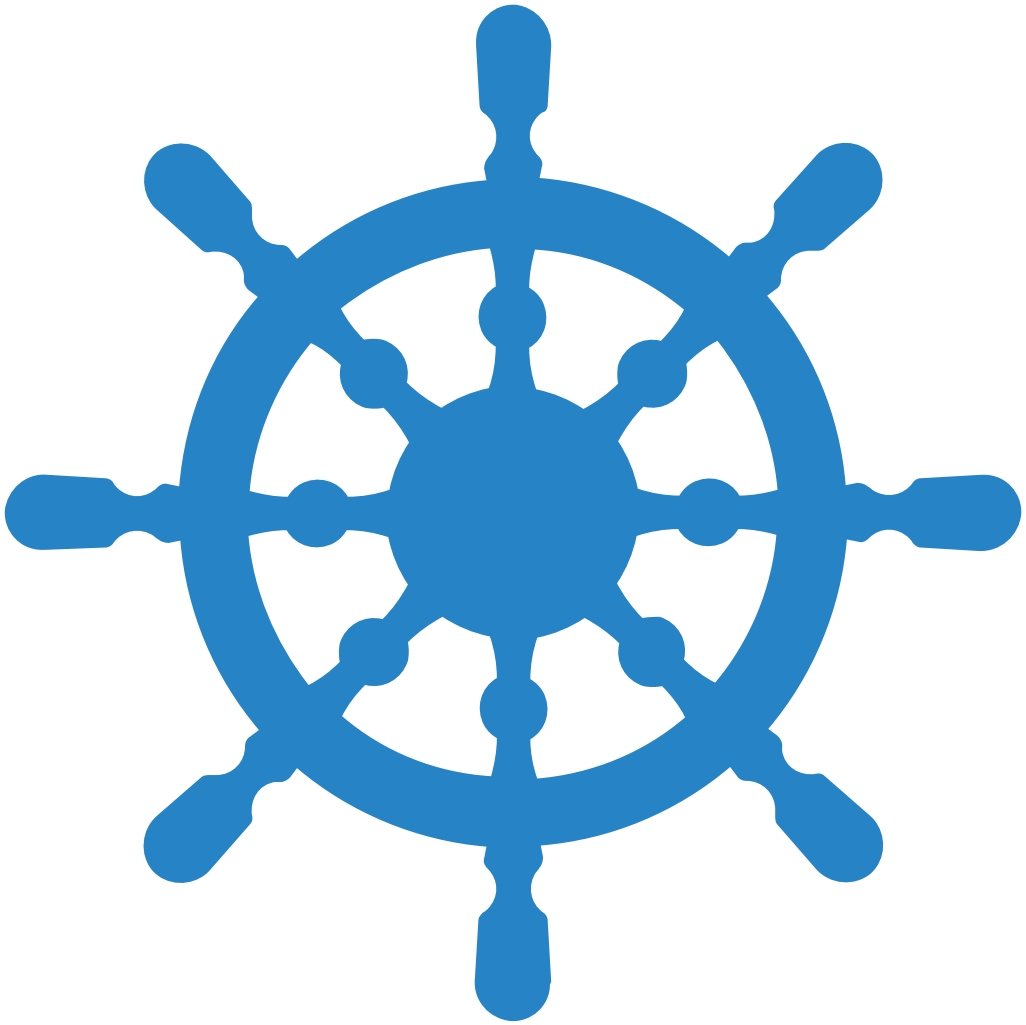
- **Operasi Data yang Digunakan:** Struktur data akan memengaruhi operasi yang dapat dilakukan pada data.
- Sebagai contoh, dalam implementasi stack atau queue, operasi push dan pop dilakukan pada ujung tertentu, sedangkan dalam linked list, operasi ini mungkin melibatkan penyesuaian pointer.

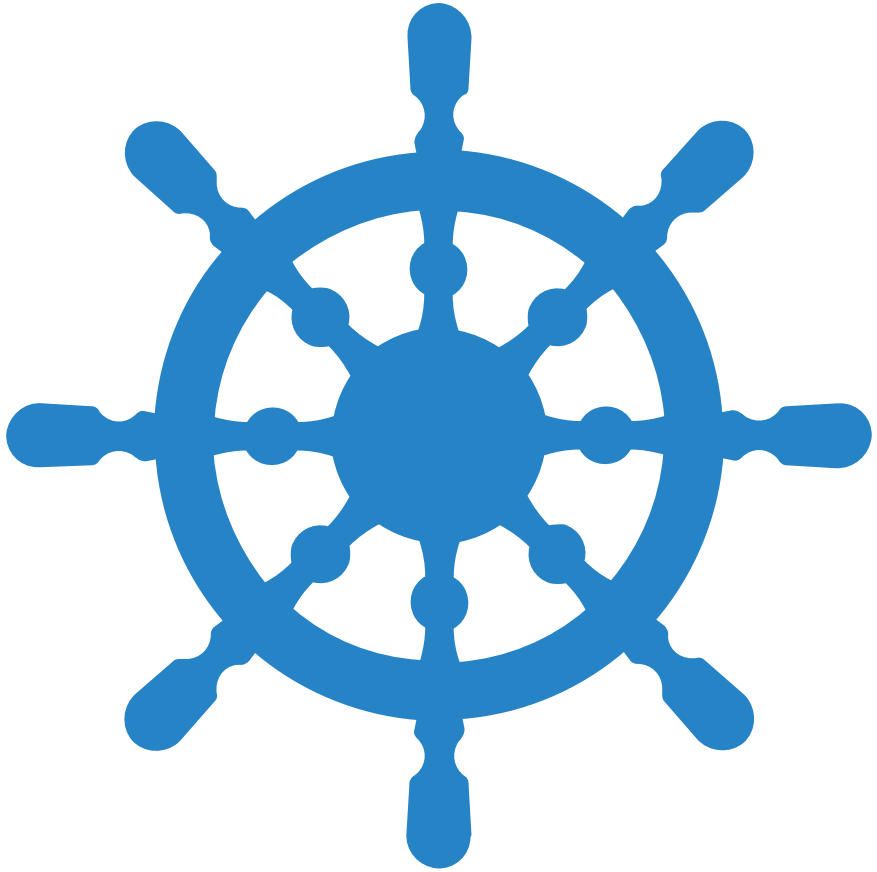
… Asosiasi Algoritma dan Data

- **Kemudahan Penggunaan Algoritma:** Pemilihan struktur data yang sesuai dapat membuat algoritma lebih mudah diimplementasikan.
- Misalnya, algoritma BFS (Breadth-First Search) pada graf seringkali lebih mudah diimplementasikan dengan menggunakan antrian (queue) sebagai struktur data pendukung.

… Asosiasi Algoritma dan Data

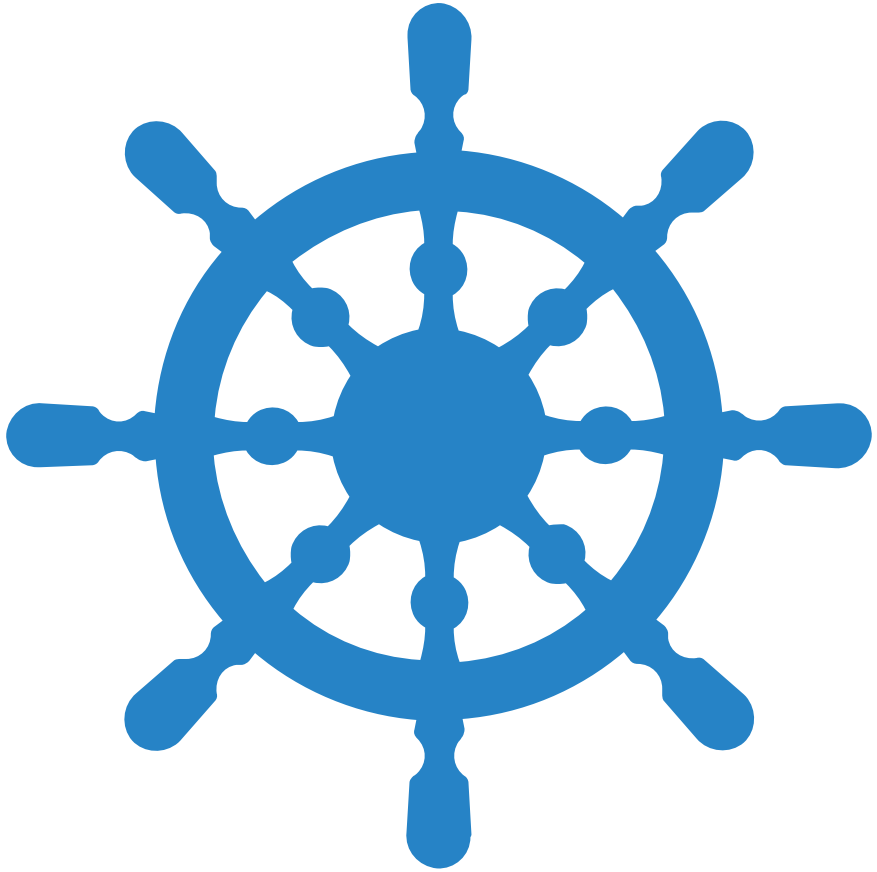
- **Penyimpanan Data:** Struktur data juga mempengaruhi bagaimana data disimpan dalam memori.
- Beberapa struktur data menggunakan alokasi memori kontigu seperti array, sementara yang lain menggunakan alokasi dinamis seperti linked list.





… Asosiasi Algoritma dan Data

- **Kecepatan Akses Data:** Struktur data berdampak pada kecepatan akses data.
- Misalnya, akses ke elemen-elemen array dilakukan dengan konstanta waktu (O(1)), sementara akses ke elemen tengah linked list membutuhkan waktu linear (O(n)).



… Asosiasi Algoritma dan Data

- **Optimasi untuk Tugas Tertentu:** Pemilihan struktur data dapat dioptimalkan untuk tugas tertentu.
- Misalnya, dalam pencarian data tercepat, menggunakan struktur data hash table bisa lebih efisien daripada menggunakan array atau linked list.

… Asosiasi Algoritma dan Data

- **Perbandingan Kinerja Algoritma:** Implementasi algoritma yang sama dengan struktur data yang berbeda dapat memiliki perbedaan kinerja yang signifikan.
- Ini dapat mempengaruhi waktu eksekusi dan penggunaan memori.

Contoh 1: Algoritma Pengurutan dalam Aplikasi Pengelolaan Data

Tujuan : Mengurutkan angka dari terkecil hingga terbesar.

Algoritma : Pengurutan **Bubblesort**

Implementasi:

1. Membaca daftar angka dari pengguna.
2. Mulai iterasi untuk setiap pasangan angka bersebelahan dalam daftar.
3. Jika angka pertama lebih besar dari angka kedua, tukar posisinya.
4. Ulangi langkah 2 dan 3 untuk setiap pasangan angka hingga tidak ada lagi perubahan.
5. Hasilnya adalah daftar angka yang terurut.

Contoh 2: Algoritma Pencarian dalam Aplikasi Pencarian Data

Tujuan : Mencari elemen tertentu dalam daftar angka.

Algoritma : Pencarian **Binary Search**

Implementasi:

1. Mengurutkan daftar angka secara terurut.
2. Menentukan batas atas dan batas bawah dari rentang pencarian.
3. Menghitung elemen tengah pada rentang pencarian.
4. Membandingkan elemen tengah dengan elemen yang dicari.
5. Jika elemen yang dicari sama dengan elemen tengah, pencarian berhasil.
6. Jika elemen yang dicari lebih besar dari elemen tengah, lanjutkan pencarian di bagian kanan.
7. Jika elemen yang dicari lebih kecil dari elemen tengah, lanjutkan pencarian di bagian kiri.
8. Ulangi langkah 2 hingga elemen ditemukan atau rentang pencarian menjadi kosong.

Contoh 3: Implementasi Struktur Data *Stack* pada Aplikasi Kalkulator

Tujuan : Membuat kalkulator sederhana

Struktur Data : Stack (tumpukan)

Implementasi:

1. Membaca ekspresi matematika dari pengguna.
2. Membaca karakter per karakter dalam ekspresi.
3. Jika karakter adalah operand (angka), masukkan ke dalam stack.
4. Jika karakter adalah operator (+, -, *, /), ambil dua operand teratas dari stack, lakukan operasi, dan hasilnya dimasukkan kembali ke stack.
5. Ulangi langkah 2-4 sampai semua karakter dievaluasi.
6. Hasil akhir berada di puncak stack dan dapat ditampilkan kepada pengguna.

Contoh 4: Implementasi Queue

Tujuan : Membuat sistem antrian untuk layanan pelanggan.

Struktur Data : *Queue* (antrian)

Implementasi:

1. Membuat aplikasi untuk menerima permintaan layanan dari pelanggan.
2. Ketika pelanggan baru datang, tambahkan data pelanggan ke dalam antrian.
3. Layanan diberikan kepada pelanggan berdasarkan urutan antrian (FIFO).
4. Ketika layanan selesai, pelanggan pertama dalam antrian dihapus.

Studi Kasus: Aplikasi Pencatatan Daftar Belanjaan

Tujuan:
Menggunakan algoritma dan struktur data untuk mengimplementasikan aplikasi pencatatan daftar belanjaan.

1. Menyimpan Daftar Belanjaan

Deskripsi:

Pengguna ingin mencatat daftar barang yang perlu dibeli.

Implementasi:

- Setiap barang dalam daftar belanjaan diwakili oleh elemen dalam struktur data (contoh: array atau linked list).
- Pengguna dapat menambahkan barang baru ke dalam struktur data.

2. Menampilkan Daftar Belanjaan

Deskripsi
Pengguna ingin melihat seluruh daftar belanjaan.

Implementasi:

- Aplikasi menampilkan semua barang dalam daftar belanjaan dalam urutan yang ditambahkan.
- Pengguna dapat melihat daftar lengkap dengan barang-barang yang perlu dibeli.

3. Menghapus Barang dari Daftar Belanjaan

Deskripsi

Pengguna ingin menghapus barang yang sudah dibeli atau tidak lagi diperlukan.

Implementasi:

- Pengguna memilih barang yang ingin dihapus.
- Aplikasi menghapus barang tersebut dari struktur data.

4. Menghitung Total Belanjaan

Deskripsi:

Pengguna ingin tahu total belanjaan yang harus dibayar.

Implementasi:

- Aplikasi menjumlahkan harga dari semua barang dalam daftar belanjaan.
- Total belanjaan ditampilkan kepada pengguna.

5. Pengurutan Daftar Belanjaan

Deskripsi:
Pengguna ingin mengurutkan daftar belanjaan berdasarkan nama atau harga.

Implementasi:

- Pengguna memilih kriteria pengurutan (nama atau harga).
- Aplikasi menggunakan algoritma sorting (misal: Bubble Sort) untuk mengurutkan barang dalam struktur data.

Latihan-Latihan

1

Buatlah algoritma untuk menentukan apakah suatu bilangan Positif, Negatif, atau Nol

2

Buatlah algoritma untuk menentukan input angka genap atau ganjil: output dari angka: genap/ganjil

3

Buatlah algoritma untuk menentukan suatu bilangan adalah bilangan prima atau bukan

Contoh Menjawab Soal Nomor 1

Algoritma:
Menentukan Bilangan Positif, Negatif, atau Nol

Input:
Sebuah bilangan bulat

Langkah-langkah:

- Masukkan bilangan bulat.
- Jika bilangan sama dengan 0, tampilkan output "nol".
- Jika bilangan lebih besar dari 0, tampilkan output "positif".
- Jika bilangan kurang dari 0, tampilkan output "negatif".

… Contoh Menjawab Soal Nomor 1

Input Bilangan = -5

Output = negatif