**TUGAS : JENIS-JENIS ALGORITMA**

**NAMA : ASRI**

**NIM : D0424506**

**A. PENGERTIAN ALGORITMA**

Algoritma pemrograman adalah **serangkaian langkah atau instruksi untuk memecahkan masalah tertentu**. Di dalam pemrograman, algoritma digunakan untuk menentukan alur logika program. Bisa dibilang bahwa algoritma adalah aspek utama ketika developer membuat program dengan bahasa pemrograman apapun.

Bagi pengguna, algoritma bisa membantu memberikan hasil yang lebih akurat, misalnya rekomendasi produk atau postingan. Sedangkan bagi developer, algoritma berguna untuk memastikan setiap langkah dalam aplikasi berjalan dengan lancar dan efisien.

**B. CARA KERJA ALGORITMA PEMOGRAMAN**

Cara kerja algoritma pemrograman pada dasarnya melibatkan tiga tahap utama, yaitu **input**, **proses**, dan **output**. Berikut penjelasan dan contohnya dalam kasus penggunaan sehari-hari:

1. **Input**. Pada tahap ini, data atau perintah yang perlu diproses oleh algoritma akan dimasukkan. Sebagai contoh, saat Anda menggunakan aplikasi ojek online, inputnya bisa berupa lokasi penjemputan dan tujuan Anda.
2. **Proses**. Data yang diinput kemudian dikirim ke sistem untuk diproses oleh algoritma pemrograman melalui rangkaian langkah yang telah ditetapkan. Di aplikasi ojek online, algoritma memproses input Anda dengan menghitung rute tercepat, tarif, dan ketersediaan pengemudi untuk menghasilkan output.
3. **Output**. Setelah data selesai diproses, algoritma akan memberikan output berupa hasil yang diharapkan. Jadi, output aplikasi ojek online dalam contoh ini adalah nama pengemudi, jenis kendaraan, tarif, dan estimasi waktu perjalanan Anda.

Berkat adanya algoritma, semua proses tersebut bisa dilakukan secara otomatis dalam waktu yang singkat untuk memberikan hasil yang akurat.

Kemudian, algoritma setiap aplikasi atau program juga berbeda-beda tergantung pada tujuan dan masalah yang perlu diselesaikan.

Sebagai contoh, aplikasi yang menyediakan fitur pencarian mungkin

menerapkan algoritma yang difokuskan untuk memilih dan mengurutkan data yang relevan. Sementara itu, aplikasi navigasi seperti Google Maps membutuhkan algoritma yang menghitung rute optimal dengan melihat berbagai kondisi lalu lintas.

Selain itu, jenis data yang diproses dan kompleksitas tugas juga menentukan struktur dan jenis algoritma yang digunakan. Ingin mempelajari jenis-jenis algoritma pemrograman lebih lanjut? Mari simak bagian berikutnya!

**C. JENIS-JENIS ALGORITMA**

Ada beberapa jenis algoritma pemrograman yang saat ini umum digunakan oleh developer. Tiap-tiapnya menerapkan metode yang berbeda-beda untuk menyelesaikan masalah dan mencapai tujuan yang diinginkan.

Berikut adalah 6 jenis algoritma pemrograman yang sering digunakan:

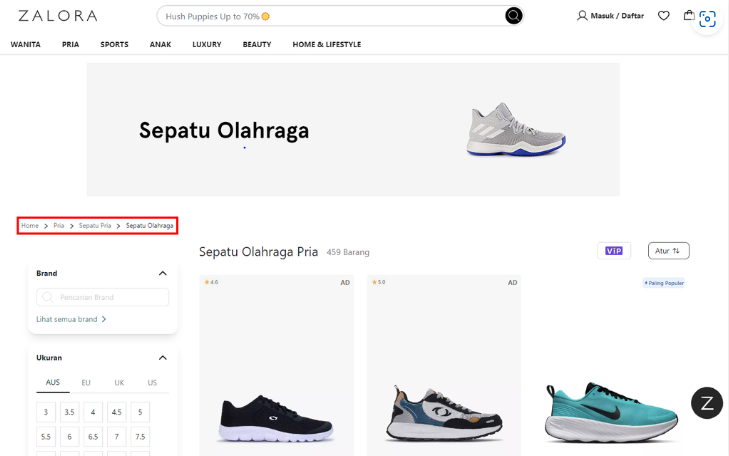
**Algoritma Recursive**

Menggunakan algoritma recursive, suatu fungsi akan memanggil dirinya sendiri untuk menyelesaikan sebuah masalah dengan memecahnya menjadi bagian-bagian yang lebih kecil.

Proses ini akan berlanjut hingga mencapai kondisi dasar, yaitu ketika masalah bisa diselesaikan tanpa adanya perulangan lagi.

Algoritma ini cocok untuk masalah yang memiliki pola berulang, misalnya saat Anda mencari produk sepatu olahraga di [**website e-Commerce**](https://www.hostinger.co.id/tutorial/contoh-website-e-commerce):

1. Anda membuka website e-Commerce lalu memilih kategori utama, misalnya **Pria**.
2. Algoritma kemudian akan mencari subkategori **Sepatu Pria** untuk mengerucutkan hasilnya.
3. Setelah itu, algoritma akan mencari lebih dalam lagi, kali ini di subkategori **Sepatu Olahraga**.
4. Selanjutnya, algoritma akan menampilkan produk yang Anda cari berdasarkan hasil dari setiap langkah tersebut, yaitu sepatu olahraga.



Dengan jenis algoritma ini, aplikasi bisa menelusuri struktur hierarkis seperti kategori produk sehingga pencarian bisa berjalan lebih cepat dan efisien.

**Algoritma Sorting**

Algoritma sorting digunakan untuk mengatur data dalam urutan tertentu, misalnya dari yang terkecil hingga terbesar (ascending) atau sebaliknya (descending). Jenis algoritma ini meningkatkan efisiensi dan kecepatan pemrosesan data, terutama saat menangani data dalam jumlah besar.

Berikut adalah beberapa jenis algoritma sorting yang umum digunakan:

* **Bubble sort**. Elemen dibandingkan secara berpasangan dan ditukar kalau tidak dalam urutan yang benar, lalu diulang sampai semuanya diurutkan.
* **Merge sort**. Data dipecah menjadi bagian-bagian yang lebih kecil, diurutkan, lalu digabung kembali secara terurut.
* **Quick sort**. Satu elemen dipilih sebagai pivot atau patokan, lalu elemen lain di sekitar pivot tersebut dibagi lagi berdasarkan ukurannya.

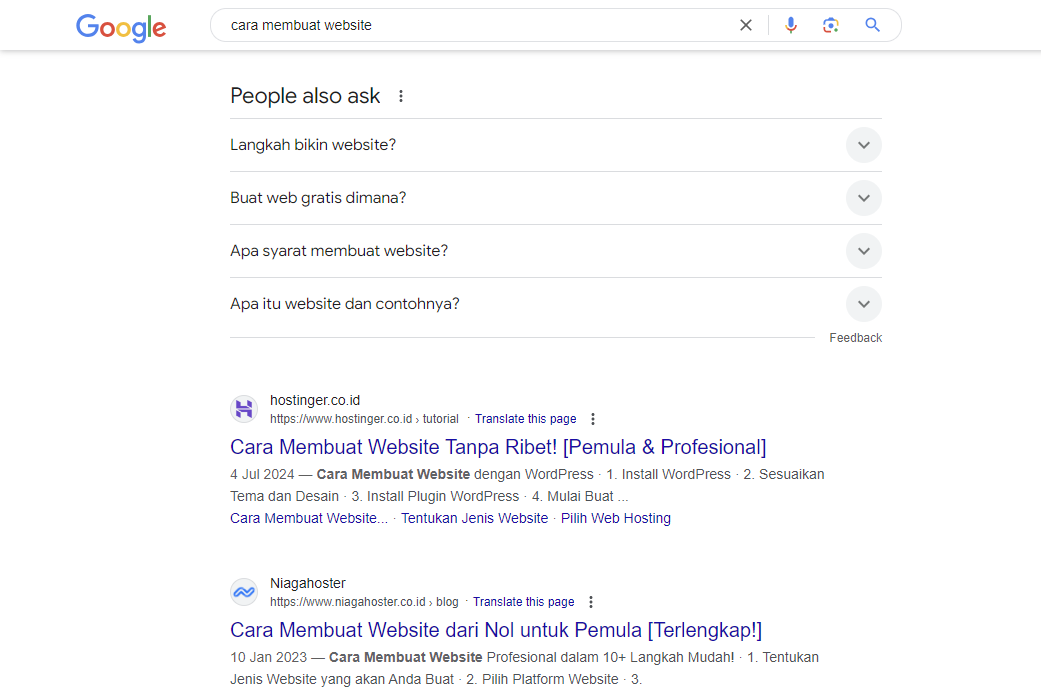
Contoh penggunaannya adalah saat Anda mengurutkan harga produk di toko online dari yang termurah hingga yang termahal. Jenis algoritma ini akan membantu sistem mengatur data secara otomatis sehingga Anda bisa menemukan produk yang relevan dengan cepat dan mudah.

**Algoritma Searching**

Jenis algoritma ini berguna untuk menemukan hasil tertentu dalam kumpulan data, seperti saat mencari kata kunci di mesin pencari atau menemukan produk di website toko online.

Algoritma searching sendiri dibagi menjadi beberapa jenis, yaitu:

* **Linear search**. Bekerja dengan memeriksa setiap elemen satu per satu hingga menemukan kecocokan.
* **Binary search**. Hanya bisa digunakan pada data yang sudah diurutkan. Data akan dipecah menjadi dua bagian, lalu algoritma ini mengecek apakah elemen yang dicari ada di bagian kiri atau kanan, kemudian mengulangi proses ini.
* **Hashing**. Algoritma ini mengubah data menjadi indeks menggunakan fungsi hash untuk pencarian yang lebih cepat. Penggunaan umumnya meliputi struktur data seperti hash table.



**Algoritma Greedy**

Jenis algoritma ini selalu memilih opsi yang dirasa paling optimal dalam setiap tahap pemrosesan untuk mendapatkan solusi terbaik.

Prinsipnya, langkah yang paling menguntungkan pada saat itu akan dipilih, tanpa mempertimbangkan langkah sebelumnya atau kemungkinan yang bisa terjadi di langkah berikutnya.

Contoh penerapannya bisa dilihat pada aplikasi peta, yang menggunakan algoritma greedy untuk menemukan rute tercepat dari satu lokasi ke lokasi lain. Algoritma ini melihat rute yang terlihat paling cepat pada setiap persimpangan hingga akhirnya menemukan jalur terpendek untuk mencapai tujuan.

**Algoritma Backtracking**

Dalam algoritma backtracking, pemecahan masalah dilakukan dengan mencoba berbagai kemungkinan solusi, lalu kembali ke tahap sebelumnya kalau ternyata solusi tersebut tidak berhasil atau berujung buntu.

Metode ini bekerja dengan mencoba semua kemungkinan solusi secara sistematis hingga menemukan solusi yang tepat. Apabila sebuah pilihan tidak memberikan hasil yang diinginkan, pilihan tersebut akan dibatalkan untuk mencoba opsi lain yang belum diuji.

Misalnya, dalam permainan [**Sudoku**](https://id.wikipedia.org/wiki/Sudoku), algoritma backtracking akan menempatkan angka di suatu kotak. Apabila angka tersebut tidak cocok di langkah berikutnya, algoritma akan kembali ke kotak sebelumnya, mengganti angka yang telah dipilih, lalu melanjutkan prosesnya dengan solusi baru.

**Algoritma Randomized**

Algoritma randomized menggunakan keputusan acak untuk menentukan langkah-langkah yang harus diambil saat menyelesaikan masalah. Metode ini sering digunakan untuk menghindari pola tertentu yang bisa memperlambat atau mengganggu efisiensi algoritma.

Jadi, bukannya mengikuti serangkaian langkah yang sudah ditetapkan, algoritma ini akan membuat beberapa pilihan acak untuk mencari solusi yang lebih baik.

Contoh sederhananya adalah saat Anda mengocok kartu secara manual. Dalam proses ini, Anda menggunakan metode pengacakan untuk mengubah urutan kartu. Nah, algoritma randomized juga bekerja dalam cara serupa, yaitu memilih elemen secara acak untuk menentukan langkah berikutnya.

**Fungsi Algoritma Pemrograman**

Dalam penggunaannya, algoritma pemrograman memiliki fungsi yang sangat penting bagi pengguna maupun developer aplikasi. Berikut beberapa yang paling umum:

**Fungsi algoritma pemrograman bagi pengguna:**

* **Efisiensi proses**. Menggunakan algoritma, aplikasi bisa berjalan lebih cepat untuk memberikan hasil yang diinginkan, misalnya ketika mencari produk di toko online.
* **Personalisasi pengalaman pengguna**. Algoritma bisa menganalisis kebiasaan pengguna untuk memberikan rekomendasi yang relevan, misalnya di aplikasi streaming video seperti [**Netflix**](https://www.netflix.com/id-en/).
* **Meningkatkan akurasi hasil pencarian**. Hasil pencarian yang disajikan bisa lebih akurat dan relevan berkat adanya algoritma.
* **Peningkatan keamanan**. Algoritma seperti hashing bisa dimanfaatkan untuk menyimpan kata sandi dengan lebih aman sehingga mencegah kebocoran data dalam aplikasi berbasis web.

**Fungsi algoritma pemrograman bagi developer:**

* **Mempermudah penataan struktur kode**. Algoritma membantu developer menulis kode secara lebih terstruktur dan logis. Hasilnya, kode menjadi lebih mudah dipahami dan dikelola.
* **Mengoptimalkan performa program**. Developer bisa memanfaatkan algoritma untuk meningkatkan performa aplikasi, terutama saat menangani data dalam jumlah yang besar atau proses yang rumit.
* **Meminimalkan duplikasi kode**. Dengan menerapkan algoritma yang efisien, developer bisa menghindari penulisan ulang kode untuk mempermudah proses maintenance aplikasi.
* **Meningkatkan skalabilitas aplikasi**. Seiring dengan bertambahnya jumlah pengguna, algoritma memudahkan developer menyesuaikan ukuran aplikasi tanpa mengorbankan performanya.
* **Mengurangi risiko bug dan error**. Algoritma yang dirancang dengan baik akan memudahkan developer menemukan dan memperbaiki bug sehingga aplikasi berjalan lebih stabil.

Nah, jadi, algoritma tidak hanya membuat aplikasi menjadi lebih cepat dan efisien bagi pengguna, tapi juga membantu developer menulis kode yang lebih optimal dan mudah dikelola, apa pun [**bahasa pemrograman**](https://www.hostinger.co.id/tutorial/bahasa-pemrograman) yang digunakan.

Tanpa adanya algoritma, berbagai proses aplikasi mungkin harus dilakukan secara manual, dan diperlukan waktu yang lama bagi pengguna untuk mendapatkan hasil yang mereka inginkan.

## Kesimpulan

Algoritma pemrograman merupakan langkah-langkah yang dirancang untuk memecahkan masalah dalam aplikasi atau sistem secara efisien dan terstruktur. Saat bekerja, algoritma akan melalui tiga tahap utama, yaitu **input**, **proses**, dan **output**.

Kemudian, algoritma juga dibagi menjadi beberapa jenis tergantung pada masalah yang ingin diselesaikan dan tujuan yang ingin dicapai. Contohnya, algoritma recursive, sorting, dan searching merupakan jenis yang paling sering dijumpai dalam kasus penggunaan sehari-hari.