**Algortima & Pemrograman**



Identitas :

Nama : Fadil Biyan Eka Satria

NIM : 048634082

Prodi : Sistem Informasi

UPBJJ : Semarang

Tanggal : 3 Mei 2024

**Universitas Terbuka**

**Tahun 2024**

**Pengertian Algoritma**

Algoritma adalah serangkaian langkah-langkah komputasi yang dirancang untuk memecahkan masalah atau menjalankan suatu tugas dengan logis dan sistematis. Munir (2007) menekankan bahwa algoritma merupakan prosedur atau langkah-langkah komputasi yang mengubah data masukan menjadi keluaran yang diinginkan. Algoritma haruslah benar dan logis sehingga dapat memberikan hasil yang diharapkan. Menurut Microsoft Press Computer and Internet Dictionary (1998), algoritma adalah serangkaian tindakan tertentu yang logis dalam menyelesaikan suatu persoalan. Logis di sini mengacu pada urutan yang tepat dan tersusun secara runtut, serta langkah-langkah tersebut harus diketahui dengan pasti untuk menjalankan algoritma secara efektif. Pendapat dari Sjukani (2005) menyatakan bahwa algoritma adalah jalan pikiran yang digunakan saat menyelesaikan sebuah operasi, yang dijabarkan dalam bentuk tertulis seperti kalimat, tabel, atau gambar. Hal ini menunjukkan bahwa algoritma dapat berbeda-beda tergantung pada cara berpikir individu. Gunadarma (1988) mendefinisikan algoritma sebagai sekumpulan perintah yang menjelaskan langkah-langkah yang jelas dalam memecahkan suatu masalah dengan tuntutan yang bersifat mekanik. Ini menekankan bahwa algoritma harus memiliki langkah-langkah yang terstruktur secara jelas dan dapat dilaksanakan secara mekanik. Secara keseluruhan, algoritma adalah cara sistematis untuk menyelesaikan masalah atau menjalankan suatu tugas dengan menggunakan langkah-langkah yang terdefinisi secara jelas dan logis. Algoritma merupakan inti dari skema kerja komputer dan penting untuk memastikan kinerja yang optimal dalam memproses informasi.

Contoh: Algoritma Pencarian dalam Daftar Nama

Deskripsi Masalah: Anda memiliki daftar nama-nama mahasiswa di sebuah kelas dan ingin mencari apakah nama tertentu ada dalam daftar tersebut.

Langkah-langkah Algoritma,

Langkah 1: Mulai dari awal daftar.

Langkah 2: Periksa nama pada posisi saat ini.

Langkah 3: Jika nama yang Anda cari cocok dengan nama pada posisi saat ini, berhenti dan tampilkan pesan bahwa nama ditemukan.

Langkah 4: Jika nama tidak cocok, lanjutkan ke nama berikutnya dalam daftar.

Langkah 5: Ulangi langkah 2-4 sampai Anda menemukan nama yang cocok atau sampai Anda mencapai akhir daftar.

Langkah 6: Jika Anda telah mencapai akhir daftar dan nama masih belum ditemukan, tampilkan pesan bahwa nama tidak ditemukan.

Contoh Penerapan: Misalkan Anda memiliki daftar nama-nama mahasiswa seperti berikut: ["John", "Jane", "Doe", "Alice", "Bob"]. Anda ingin mencari apakah nama "Alice" ada dalam daftar tersebut.

Implementasi Algoritma,

Anda mulai dari awal daftar, yaitu nama "John".

Anda memeriksa nama pada posisi ini. Karena "John" bukan "Alice", Anda lanjutkan ke nama berikutnya.

Anda memeriksa nama "Jane". Karena "Jane" bukan "Alice", Anda lanjutkan lagi.

Anda memeriksa nama "Doe". Karena "Doe" bukan "Alice", Anda lanjutkan lagi.

Anda memeriksa nama "Alice". Karena "Alice" cocok dengan nama yang Anda cari, Anda berhenti dan tampilkan pesan bahwa nama ditemukan.

Dengan menggunakan algoritma pencarian seperti ini, kita dapat menemukan apakah nama tertentu ada dalam daftar dengan cara yang sistematis dan efisien. Algoritma ini memastikan bahwa memeriksa setiap nama dalam daftar sampai menemukan yang dicari atau sampai akhir daftar.

**Pengertian Pemrograman**

Pemrograman adalah proses pembuatan program menggunakan komputer, yang melibatkan penulisan instruksi-instruksi untuk melakukan tugas tertentu. Program merupakan implementasi dari algoritma yang ditulis dalam bahasa pemrograman tertentu agar bisa dieksekusi oleh komputer. Pemrograman membutuhkan keahlian dalam algoritma, bahasa pemrograman, matematika, dan logika untuk menciptakan program yang sesuai dengan kebutuhan pengguna. Bahasa pemrograman merupakan alat komunikasi antara programmer dan komputer, yang memungkinkan penentuan data yang akan diproses dan langkah-langkah yang harus diambil dalam setiap kondisi.

Standar perencanaan pemrograman yang baik harus memperhatikan kemampuan dalam memecahkan masalah dan pemrograman secara efisien. Teknik top-down sering digunakan dalam pemecahan masalah kompleks, di mana masalah dibagi menjadi subkelompok yang lebih kecil dan kemudian disusun langkah-langkah rinci dalam algoritma. Dengan mengikuti algoritma yang telah disusun, programmer dapat membuat program yang efektif dan efisien untuk menyelesaikan masalah yang dihadapi. Pemrograman tidak hanya tentang menulis kode, tetapi juga tentang merencanakan secara sistematis dan logis bagaimana suatu program harus berfungsi untuk mencapai tujuan yang diinginkan.

Unsur-unsur pemrograman terdiri dari input, proses, dan output. Input adalah data yang dimasukkan melalui perangkat seperti keyboard, mouse, atau scanner. Proses adalah tahapan di mana data yang telah diinput diolah menggunakan algoritma dan teknik tertentu untuk menghasilkan output. Output kemudian ditampilkan atau dilaporkan kepada pengguna melalui perangkat seperti monitor atau printer.

Prinsip-prinsip desain perangkat lunak, dikenalkan oleh Robert C. Martin, bertujuan untuk membantu pengembang merancang perangkat lunak dengan lebih baik. Salah satu prinsip yang dikenal sebagai SOLID, terdiri dari lima prinsip: single responsibility principle, open closed principle, liskov substitution principle, interface segregation principle, dan dependency inversion principle. Single responsibility principle menekankan bahwa setiap kelas harus bertanggung jawab atas satu fungsionalitas tertentu, sementara open closed principle mengizinkan penambahan fitur tanpa memodifikasi kode yang sudah ada. Liskov substitution principle membahas penggantian objek dengan objek yang memiliki perilaku yang sama, sementara interface segregation principle mengusulkan pembagian interface ke dalam unit yang lebih kecil. Terakhir, dependency inversion principle mempromosikan ketergantungan pada abstraksi daripada implementasi konkrit, untuk mengurangi keterikatan dan meningkatkan fleksibilitas sistem.

Paradigma pemrograman merupakan sudut pandang atau pendekatan yang digunakan dalam mengklasifikasikan suatu bahasa pemrograman. Beberapa paradigma pemrograman yang umum digunakan meliputi:

1. Pemrograman fungsional: Paradigma ini menggunakan fungsi sebagai alat utama untuk memetakan masalah ke solusi. Fungsi dianggap sebagai objek utama dan dapat dibentuk seperti variabel atau data. Pemrograman fungsional sering menggunakan rekursi untuk melakukan pengulangan atau looping. Contoh bahasa pemrograman yang menggunakan paradigma ini antara lain LISP, Haskell, Scala, dan Scheme.

2. Pemrograman prosedural: Paradigma ini didasarkan pada konsep panggilan prosedur, di mana pernyataan disusun menjadi prosedur atau fungsi. Instruksi-instruksi diberikan langkah demi langkah, dan bahasa pemrograman prosedural sering dikenal sebagai top-down. Contoh bahasa yang menggunakan paradigma ini adalah Fortran dan Cobol.

3. Pemrograman logika: Paradigma ini didasarkan pada pendefinisian relasi antara individu yang dinyatakan sebagai predikat. Programmer harus menguraikan fakta dan aturan yang menjadi panduan saat program dieksekusi. Contoh bahasa pemrograman yang menggunakan paradigma ini adalah Prolog.

4. Paradigma berorientasi objek (OOP): Paradigma ini menggunakan konsep kelas dan objek untuk menyelesaikan permasalahan. Objek adalah instansi dari kelas, di mana setiap objek memiliki atribut dan metode. Objek dari kelas yang berbeda dapat berinteraksi satu sama lain. Contoh bahasa pemrograman yang menggunakan paradigma ini antara lain Delphi, Java, C++, PHP, dan Python.

Algoritma pemrograman memiliki banyak manfaat penting dalam pengembangan perangkat lunak. Pertama, algoritma memungkinkan pemrogram untuk memecahkan masalah yang kompleks dengan cara yang terstruktur dan logis. Hal ini membantu dalam menghindari kesalahan yang mungkin terjadi dalam pengembangan program. Kedua, algoritma digunakan untuk menyempurnakan program dari proyek yang besar hingga proyek yang lebih sederhana, dengan metode seperti hierarki atau pendekatan membagi dan menaklukkan. Ketiga, algoritma memudahkan dalam mengatasi masalah yang muncul pada program secara berurutan, sehingga proses debugging menjadi lebih efisien. Algoritma juga membantu mengubah program yang besar menjadi lebih sederhana dan efisien. Hal ini memungkinkan penggunaan ulang atau penggunaan berulang dari fungsi-fungsi tertentu, meminimalisir penulisan program yang berulang, dan memudahkan dalam membuat program secara sistematis. Kedua, algoritma memperkuat cara berpikir untuk memecahkan masalah, melatih berpikir jangka panjang, dan memperkuat analisis dalam pembuatan program. Ketiga, algoritma memungkinkan deskripsi langkah-langkah penyelesaian suatu masalah dalam bentuk teks yang dapat dibuat oleh siapa pun. Namun, untuk dapat dieksekusi oleh komputer, algoritma perlu diterjemahkan ke dalam bahasa pemrograman tertentu. Secara keseluruhan, algoritma pemrograman menjadi aspek penting dalam pengembangan perangkat lunak karena membantu dalam memecahkan masalah, menyederhanakan program, memperkuat cara berpikir, dan memungkinkan penggunaan ulang dan penggunaan berulang dari fungsi-fungsi tertentu. Selain itu, algoritma memungkinkan deskripsi langkah-langkah penyelesaian masalah dalam bentuk teks yang dapat dimengerti oleh siapa pun, meskipun untuk dieksekusi oleh komputer, algoritma perlu diterjemahkan ke dalam bahasa pemrograman yang dipilih.

Algoritma dan pemrograman memiliki hubungan yang erat karena algoritma merupakan langkah-langkah atau prosedur sistematis untuk menyelesaikan suatu masalah, sedangkan pemrograman adalah implementasi dari algoritma tersebut dalam bahasa pemrograman tertentu. Algoritma digunakan untuk merancang cara-cara atau langkah-langkah yang logis dan terurut untuk menyelesaikan suatu masalah atau tugas tertentu. Algoritma membantu programmer dalam merencanakan dan merancang solusi yang efisien sebelum melakukan implementasi dalam kode program. Dengan algoritma yang baik, programmer dapat memahami secara jelas langkah-langkah yang diperlukan untuk menyelesaikan masalah yang dihadapi.

Setelah algoritma dirancang, langkah berikutnya adalah mengimplementasikannya dalam bahasa pemrograman tertentu. Pemrograman melibatkan penulisan kode-kode instruksi yang sesuai dengan algoritma yang telah dirancang sebelumnya. Proses pemrograman mengubah algoritma menjadi bentuk yang dapat dipahami oleh komputer, sehingga komputer dapat menjalankan instruksi-instruksi tersebut untuk menyelesaikan masalah yang diberikan. Dengan demikian, algoritma dan pemrograman saling melengkapi dalam proses pengembangan perangkat lunak. Algoritma memberikan panduan tentang bagaimana menyelesaikan suatu masalah secara logis, sementara pemrograman merupakan implementasi konkret dari algoritma tersebut dalam bahasa yang dapat dipahami oleh komputer.

**Sumber**

<https://files.osf.io/v1/resources/agnzm/providers/osfstorage/61d862ac2962ce13efb02dd3?action=download&direct&version=1>

<https://journal.widyakarya.ac.id/index.php/konstanta-widyakarya/article/download/1875/1814/6583>

<http://eprints.poltekkesjogja.ac.id/11279/1/Modul%20Algoritma%20dan%20Pemograman.pdf>

<https://repository.unikom.ac.id/37773/1/00%20Definisi.pdf>

<https://medium.com/@mashidicky/5-contoh-algoritma-dan-flowchart-dalam-kehidupan-sehari-hari-869033068f78>