

## CONTACT

082318332771  
lg: @esaprmna  
909.555.0100  
Esapermanam124@gmail.com  
www.esapermana.com

## COMMUNICATION

I have received several awards for my outstanding communication skills, including recognition for providing exceptional patient education and counseling.

## LEADERSHIP

I received the "Outstanding Informatics Student" award during my time at Informatics school, and I have been recognized for my contributions to the development of the campus and company while I was working.

# ESA PERMANA

## REGISTERED NURSE

## EDUCATION

SDN 1 GEREBA  
SMPN 1 KAWALI  
SMKN 1 KAWALI  
D3 MANAGEMENT INFORMATIKA POLITEKNIK LP3I TASIKMALAYA

Relevant subjects: programming, website design, IT consulting, management science and data analysis.

## EXPERIENCE

2027-2028

Data Analyst | Mostar Studio

2028-2030

Programer | Swala Technology

2030-2034

Manager | Swala Technology

I have a good track record in the company by always contributing greatly to company projects

## REFERENCES

[Available upon request]

# FULL TIME



## REAL MADRID 0 VS 4 BARCELONA

R. Lewandowski 54' 46'

Lamine Yamal 77'

Raphinha 84'

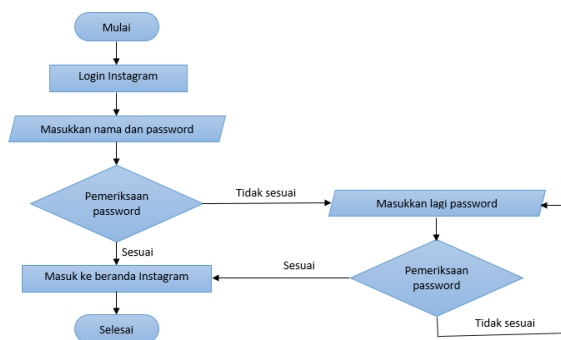
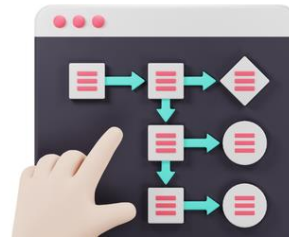
Santiago Bernabeu • Madrid

Okt 27 2024 • 02:00 AM

## Algoritma: Pengertian, Sejarah, Jenis, Fungsi, dan Contohnya

Pengertian Algoritma – Saat ini, Anda mungkin telah mendengar banyak tentang apa itu algoritma. Ya, algoritma sangat penting dalam dunia pemrograman. Misalnya, ini adalah trik yang digunakan YouTube untuk membantu Anda menonton video yang direkomendasikan YouTube. Cukup menarik bukan?

### Algoritma dalam Pemrograman Komputer?



Bagi Anda yang sudah lama berkecimpung di dunia teknik komputer pasti mengenal istilah ini. Namun, bagi yang baru menggunakan perangkat ini, tentu tidak jelas mengenai komponen, fitur, dan tujuan penggunaan perangkat ini.

Algoritma tidak hanya digunakan untuk menyelesaikan masalah pada komputer, tetapi juga dapat digunakan untuk menyelesaikan masalah yang ada dalam kehidupan sehari-hari. Selain itu, masalah apa pun yang terkait dengan proses atau langkah prosedural biasanya memerlukan algoritma.

Masalah mudah dipecahkan dengan menggunakan algoritma karena kita tahu langkah-langkah logis dan sistematis yang harus diambil terlebih dahulu. Oleh karena itu, banyak orang telah memecahkan masalah menggunakan algoritma, apakah Anda salah satunya?.

Banyak istilah yang ada di dunia komputasi seperti *AI*, *supercomputer*, *machine learning*, *cloud*, *quantum computing* dan masih banyak lagi. Terlepas dari ini, ada kata yang digunakan terutama dalam ilmu komputer dan itu adalah algoritma.

Dalam pengertian yang paling umum, algoritma adalah serangkaian instruksi yang memberitahu komputer bagaimana mengubah satu set fakta tentang dunia menjadi informasi yang berguna.

Perkembangan ilmu pengetahuan dan teknologi memungkinkan manusia menghasilkan karya yang semakin canggih dan kompleks. Meskipun komputer dapat melakukan perhitungan lebih cepat daripada manusia pada umumnya, komputer tidak dapat memecahkan masalah tanpa diajarkan oleh manusia melalui urutan langkah (algoritma) yang telah ditentukan sebelumnya.

Selain digunakan untuk menyelesaikan masalah oleh komputer, algoritma juga dapat diterapkan untuk menyelesaikan masalah sehari-hari yang membutuhkan serangkaian proses atau langkah-langkah prosedural. Untuk lebih memahami apa itu algoritma, mari kita pelajari apa artinya dari beberapa sumber.

Pengertian umum dari suatu algoritma adalah urutan dari sejumlah langkah logis dan sistematis untuk memecahkan suatu masalah tertentu.

Beberapa ahli menganggap algoritma sebagai urutan langkah-langkah yang harus diikuti dalam matematika atau perhitungan untuk memecahkan masalah lain, terutama komputer. Artinya semua susunan logis yang disusun dalam urutan sistem pembelajaran tertentu dan digunakan untuk memecahkan suatu masalah tertentu, dapat kita katakan seperti sebuah algoritma.

Umumnya algoritma akan digunakan untuk dapat melakukan perhitungan, melakukan inferensi otomatis, dan mengolah data pada komputer dengan menggunakan *software* tertentu. Selain itu, Algoritma juga berisi serangkaian instruksi terbatas serta instruksi khusus untuk menghitung fungsi yang, ketika dieksekusi dan diproses, menghasilkan output tertentu dan kemudian berhenti pada kondisi terminasi yang ditentukan.

Algoritma adalah suatu langkah atau metode yang telah direncanakan secara matang sehingga terurut dan terorganisir dengan baik dan biasanya digunakan untuk memecahkan suatu masalah dengan memberikan suatu instruksi sehingga menjadi suatu tindakan.

Sedangkan dalam Kamus Besar Bahasa Indonesia (KBBI), algoritma adalah suatu prosedur sistematis untuk menyelesaikan masalah matematika dalam langkah-langkah terbatas atau urutan pengambilan keputusan yang logis untuk memecahkan masalah tersebut.

Dari pengertian tersebut dapat dikatakan bahwa algoritma ini digunakan untuk menyelesaikan atau menyelesaikan suatu masalah dengan langkah-langkah logis yang terurut.

Inilah sebabnya mengapa algoritma harus digunakan pada elektronik komputer karena dengan algoritma komputer akan dapat memproses data, melakukan perhitungan, membuat inferensi otomatis dan memecahkan masalah di komputer.

Ketika algoritma tersebut digunakan pada komputer, maka akan menghasilkan output yang kemudian akan berhenti pada keadaan awal. Tidak hanya pada komputer atau dalam

kehidupan sehari-hari, algoritma juga digunakan oleh banyak perusahaan terutama yang bergerak di bidang keuangan.

Hampir semua bidang perusahaan keuangan menggunakan algoritma, mulai dari perdagangan saham dan aset hingga manajemen utang dan penetapan harga pinjaman. *Trading* menggunakan algoritma sering disebut sebagai *trading* otomatis.

Perdagangan otomatis sering menggunakan program komputer untuk menjual atau membeli sekuritas dengan cepat, mengapa menggunakan program komputer? Hal ini karena transaksi bisnis terjadi sangat cepat, sehingga kecepatan tidak mungkin dihitung sama manusia.

Secara umum, perdagangan yang sangat cepat terjadi pada saat naik turunnya harga saham, komoditas, dan obligasi. Sederhananya, algoritma ini memfasilitasi perdagangan otomatis di mana transaksi perdagangan berlangsung sangat cepat.

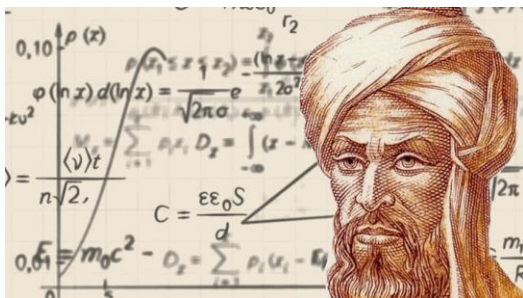
Jadi algoritma jelas merupakan salah satu metode dimana suatu masalah dapat diselesaikan dengan mudah, sehingga suatu operasi atau proses produksi dapat berlanjut. Selanjutnya, algoritma merupakan metode yang dapat dipelajari oleh semua orang, termasuk Anda sendiri.

Apalagi di era teknologi ini, mempelajari algoritma adalah sesuatu yang bisa dipelajari, seperti menulis artikel agar muncul di halaman pertama mesin pencari, Anda harus mempelajari algoritma. Tidak hanya itu, pasti banyak sekali orang yang menonton youtube, kemunculan video di halaman tersebut adalah hasil dari algoritma. Munculnya video di halaman tersebut adalah hasil dari algoritma youtube.

## Sejarah Algoritma

Saat itu, Al-Khawarizmi menulis sebuah buku dengan judul *Al Jabar wal-Muqabala* yang artinya Buku Pemulihan dan Pengurangan (*The Book of Restoration and Reduction*). Dari judul buku kita juga mendapatkan kata “aljabar” atau lebih dikenal dengan istilah algebra.

Abu Abdullah Ibn Musa al-Khawarizmi (770- 840M) lahir di Khawarizm (Kheva), sebuah kota di selatan Sungai Oxus (sekarang disebut Uzbekistan) pada 770 M. Al Khawarizmi adalah salah satu ilmuwan terkenal pada masanya. Ada beberapa cabang matematika yang ditemukannya antara lain yang disebut astronom dan ahli geografi. Awalnya, algoritma adalah istilah



yang mengacu pada aturan aritmatika yang berguna untuk memecahkan masalah menggunakan angka Arab.

Pada tahun 1950, kata algoritma pertama kali digunakan dalam “*Algoritma Euclidean*”. Euclid, seorang matematikawan Yunani (lahir pada 350 M). Dalam bukunya *Elements* menulis langkah-langkah untuk menemukan pembagi persekutuan terbesar (*common greatest divisor* atau *gcd*), dari dua bilangan bulat,  $m$  dan  $n$  [KNU 73] (tentu saja Euclid tidak menyebut metodenya sebagai algoritma, itu hanya di zaman modern orang menyebut metodenya (“*algoritma Euclidean*”).

Pembagi persekutuan terbesar dari dua bilangan bulat non-negatif adalah bilangan bulat positif terbesar yang membagi dua bilangan sama rata.

*Misalnya,  $m = 80$  dan  $n = 12$ . Semua faktor pembagi 80 adalah 1, 2, 4, 5, 8, 10, 16, 20, 40, 80 dan semua faktor pembagi 12 adalah 1, 2, 3, 4, 6, 12, maka  $\gcd(80, 12) = 4$ .*

*Langkah-langkah mencari  $\gcd(80, 12)$  dengan algoritma Euclidean sebagai berikut:*

*80 dibagi 12 hasilnya = 6, sisa 8 (atau:  $80 = 6 \cdot 12 + 8$ )*

*12 dibagi 8 hasilnya = 1, sisa = 4 (atau:  $12 = 1 \cdot 8 + 4$ )*

*8 dibagi 4 hasilnya = 2, sisa = 0 (atau:  $8 = 4 \cdot 2 + 0$ )*

*karena pembagian yang terakhir menghasilkan 0, maka sisa pembagian terakhir sebelum 0, yaitu 4, menjadi  $\gcd(80, 12)$ . Jadi,  $\gcd(80, 12) = \gcd(12, 8) = \gcd(8, 4) = \gcd(4, 0) = 4$ .*

## Jenis-Jenis Algoritma

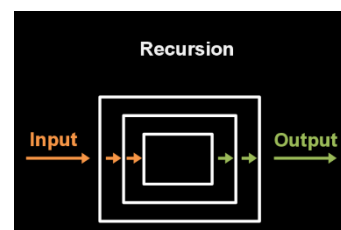
Seorang matematikawan dan ilmuwan komputer bernama Dr. Christoph Koutschan mengatakan bahwa setidaknya ada 32 algoritma dalam ilmu komputer. Namun jika dilihat dari fungsinya, hanya ada enam algoritma dasar, yaitu:

### 1. Rekursi

Sebuah algoritma rekursi adalah sesuatu yang akan memanggil dirinya sendiri berulang-ulang sehingga masalah dapat diselesaikan dengan benar. Berikut adalah beberapa kode yang dapat menemukan faktorial menggunakan algoritma rekursi.

*Fact(y)*

*If y is 0*

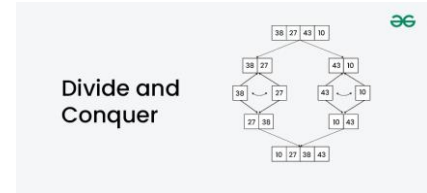


*return l*

*return (y\*Fact(y-1)) /\* this is where the recursion happens \*/*

## 2. Divide and Conquer

*Divide and Conquer* akan membagi masalah besar menjadi banyak menjadi masalah kecil. Jenis algoritma ini sendiri sebenarnya terdiri dari dua bagian utama, yaitu:



- Memecah masalah menjadi submasalah yang lebih independen dan lebih kecil dari masalah lain yang sejenis.
- Memecahkan masalah asli setelah dapat menyelesaikan masalah yang lebih kecil secara terpisah.

Berikut ini adalah contoh kode semu dari algoritma *divide and conquer*:

*MergeSorting(ar[], l, r)*

*If  $r > l$*

*Find the mid-point to divide the given array into two halves:*

*middle  $m = (l+r)/2$*

*Call mergeSorting for the first half:*

*Call mergeSorting(ar, l, m)*

*Call mergeSorting for the second half:*

*Call mergeSorting(ar, m+1, r)*

*Merge the halves sorted in step 2 and 3:*

*Call merge(ar, l, m, r)*

## 3. Dynamic Programming

*Dynamic Programming* akan bekerja dengan mengingat hasil dari proses masa lalu dan juga menggunakannya untuk menemukan hasil baru.

Dari penjelasan diatas, berarti bahwa *Dynamic Programming* memecahkan masalah kompleks dengan memecahnya menjadi banyak submasalah sederhana, kemudian menyelesaikannya satu per satu, lalu menyimpannya untuk digunakan di masa mendatang.

Contoh *Dynamic Programming* adalah terdiri dari urutan Fibonacci, berikut adalah contohnya:

*Fibonacci(N) = 0 (for  $n=0$ )*

*= 0 (for  $n=1$ )*

$$= \text{Fibonacci}(N-1) + \text{Fibonacci}(N-2)$$

#### 4. Greedy

Tipe ini dapat digunakan untuk menyelesaikan masalah optimasi. Dalam algoritma ini, kita akan menemukan solusi yang lebih optimal secara lokal tanpa mengkhawatirkan konsekuensi yang akan terjadi di masa depan dan kita dapat menemukan solusi yang lebih optimal secara global.

Namun, cara ini tidak menjamin Anda bisa mendapatkan solusi yang optimal. Ada lima komponen yang terdapat dalam algoritma ini, yaitu:

- Himpunan kandidat yang solusinya akan dicari kemudian.
- Fungsi seleksi terakhir akan membantu memilih kandidat terbaik.
- Fitur kelayakan dapat membantu Anda mengidentifikasi kandidat mana yang dapat digunakan untuk menemukan solusi.
- Fungsi tujuan mampu memberikan nilai ke solusi yang mungkin atau solusi parsial.
- Fungsi solusi dapat memberitahu Anda tentang waktu yang diperlukan untuk menemukan solusi untuk masalah tersebut.

#### 5. Brute Force

Konsep dari algoritma ini sebenarnya sangat sederhana. Selama proses ini, *brute force* mengintegrasikan semua solusi yang mungkin untuk menemukan satu atau lebih solusi yang mungkin untuk memecahkan masalah.

Berikut adalah contoh pencarian sekuensial yang dilakukan menggunakan *brute force*:

*Algorithm S\_Search (A[0..n], X)*

$A[n] \leftarrow X$

$i \leftarrow 0$

*While*  $A[i] \neq X$  *do*

$i \leftarrow i + 1$

*if*  $i < n$  *return*  $i$

*else return*  $-1$

#### 6. Algoritma Backtracking

Backtracking adalah teknik yang dapat memecahkan masalah yang berbeda secara rekursi dan mencoba mereka untuk menemukan solusi dengan memecahkan satu bagian



dari masalah pada waktu yang sama. Jika solusi gagal, kita dapat menghapusnya dan kembali mencari solusi lain.

Artinya algoritma ini akan menyelesaikan sub masalah dan jika gagal maka algoritma akan membatalkan langkah terakhir dan memulai kembali untuk mencari solusi dari masalah tersebut.

## Fungsi Algoritma

Setelah mengetahui definisi dan karakteristik algoritma, Anda harus mengetahui fungsi dari algoritma tersebut. Benarkah algoritma hanya dibutuhkan dalam [pemrograman](#)?

Seperti disebutkan sebelumnya, algoritma dapat diterapkan ke banyak jalur kehidupan. Hanya saja penerapan algoritma yang lebih sering dibahas dalam pemrograman komputer.

Dalam hal pemrograman, algoritma pemrograman membantu tim pemrograman untuk memecahkan masalah. Serangkaian solusi untuk masalah ini yang dapat dikembangkan oleh seorang programmer menjadi sebuah program.

Selain itu, beberapa fungsi lain dari algoritma adalah sebagai berikut:

- Membantu Anda memecahkan masalah tertentu dengan cara yang logis dan sistematis. Algoritma dapat digunakan lebih dari satu kali untuk menyelesaikan masalah yang sama.
- Algoritma juga dapat digunakan untuk membantu memecahkan masalah yang kompleks dengan lebih sederhana.
- Bagi programmer, algoritma akan memudahkan mereka untuk melacak setiap kesalahan yang muncul.
- Memungkinkan pemrogram untuk dengan mudah memodifikasi program tanpa mengubah total algoritma dan tanpa harus mengulang dari awal.
- Pemrogram hanya perlu melakukan perubahan pada waktu tertentu, dan program dapat diperbarui untuk memperbaiki masalah yang muncul.

## 5 Karakteristik Algoritma

Menurut Donald E. Knuth, algoritma harus memiliki lima karakteristik penting yang saling berhubungan. Kriteria untuk algoritma ini meliputi:

### 1. *Finiteness* (Keterbatasan)

Algoritma harus berhenti setelah mengambil sejumlah langkah yang terbatas, yaitu ada tujuan akhir yang tercapai, sehingga program akan berhenti ketika tujuan akhir telah

tercapai. Program yang tidak pernah berhenti menunjukkan bahwa program tersebut mengandung algoritma yang salah.

## 2. **Definiteness (Kepastian)**

Setiap langkah harus didefinisikan dengan tepat dan tidak ambigu. Ada instruksi yang jelas dan tidak ambigu, sehingga tidak ada kesalahan dalam menghasilkan *output*.

## 3. **Input (Masukan)**

Input ini adalah masalah yang diketahui dan solusi akan diselidiki. Algoritma ini tidak memiliki atau lebih input, yang merupakan jumlah yang disediakan untuk algoritma untuk diproses.

## 4. **Output (Keluar)**

Algoritma tidak memiliki atau lebih nilai *output*. *Output* ini tentunya harus menjadi solusi atau solusi dari suatu masalah. *Output* dapat berupa pesan atau kuantitas yang terkait dengan input.

## 5. **Effectiveness (keefektivitasan)**

Algoritma harus efisien, setiap urutan atau langkah harus sesederhana mungkin, sehingga dapat diimplementasikan dalam waktu yang wajar.

## **Contoh Algoritma**

Untuk lebih memahami algoritma, perlu diketahui bahwa algoritma tidak hanya berlaku dalam pemrograman tetapi dapat diterapkan dalam kehidupan sehari-hari.

### **Contoh Soal:**

Data yang diberikan berupa nama pegawai dan hasil evaluasi kinerja. Jika hasil evaluasi kinerja lebih besar atau sama dengan 75, maka pegawai tersebut dinyatakan telah mencapai KPI. Sedangkan jika nilainya kurang dari 75, pegawai tersebut dinyatakan gagal mencapai KPI.

**Contoh penulisan algoritmanya adalah seperti berikut ini:**

### **Contoh 1**

*Algoritmanya akan sebagai berikut:*

*baca nama dan nilai karyawan.*

*jika nilai  $\geq 75$  maka*

*keterangan = berhasil*

*tetapi jika  $\leq 75$*

*keterangan = tidak berhasil.*

*tulis nama dan keterangan*

**Contoh diatas merupakan algoritma narasi**

## Contoh 2

Deklarasi dari tipe datanya akan seperti berikut:

Nama = string

Nilai = integer

Keterangan = string

Algoritmanya akan seperti berikut:

read (nama, nilai)

if nilai  $\geq$  75 then

keterangan = 'berhasil'

else

keterangan = 'tidak berhasil'

write(nama, keterangan)

**Contoh diatas merupakan algoritma Pseudo Code**

## Contoh 3

Sementara itu, contoh lain dari algoritma juga dapat dilihat pada teknologi khususnya pada mesin pencari yang penggunaannya membutuhkan internet. Untuk lebih memahami contoh algoritma dalam teknologi, Anda dapat melihat contoh di bawah ini.

1. *Tentukan informasi yang ingin Anda temukan*
2. *Masukkan informasi yang ingin Anda cari ke mesin pencari*
3. *Tunggu sebentar*
4. *Informasi muncul sebagai artikel yang dicari secara berurutan*
5. *Pilih saja artikel yang berisi informasi yang tepat*

Pada umumnya *search engine* semua memiliki algoritma masing-masing, jadi artikel yang akan muncul di urutan pertama adalah artikel yang layak untuk diurutkan di urutan pertama, kemudian artikel kedua dan seterusnya.

Oleh karena itu, jika Anda ingin membuat artikel yang muncul di halaman pertama atau urutan pertama mesin pencari, Anda harus menentukan algoritma mesin pencari terlebih dahulu.

Di atas adalah 3 contoh algoritma yang ada dalam kehidupan sehari-hari dan algoritma yang berhubungan dengan teknologi mesin pencari. Pada dasarnya, masih banyak contoh algoritma, mungkin Anda bisa memberikan contoh algoritma. Setelah mengenal definisi dan contoh algoritma, sepertinya Anda sudah mulai memahami algoritma atau bahkan ingin menerapkannya dalam kehidupan sehari-hari.