

## Pengenalan Algoritma Pemrograman Dasar Dalam Konteks Pembelajaran Pemrograman Awal

**Armina Rangkuti**

Mahasiswa Prodi Pend. Matematika, FITK, UIN Sumtera Utara

Email : [arminaray23@gmail.com](mailto:arminaray23@gmail.com)

**Yahfizham Yahfizham**

Dosen Prodi Pend. Matematika, FITK, UIN Sumtera Utara

Email : [yahfizham@uinsu.ac.id](mailto:yahfizham@uinsu.ac.id)

Korespondensi penulis : [arminaray23@gmail.com](mailto:arminaray23@gmail.com)

**ABSTRACT.** Algorithms are steps that are arranged logically and systematically to solve a problem. In the world of simple programming, the algorithm is the initial stage and is written before the program is written. One of the problems that can be solved with computer programming is problems that involve mathematical calculations. This research aims to develop understanding of algorithms and develop programming skills, especially for beginners. The method used in this research is the literature study or literature review method, namely analyzing several studies related to the topic discussed, especially introducing the basic concepts of programming algorithms, characteristics of programming algorithms, grouping programming languages based on level and algorithm notation, which are some of the basic things that are mandatory. known by everyone who wants to create a program. This understanding will be very helpful in the development of more complex programming projects. Therefore, learning basic programming algorithms is very crucial or important for anyone who wants to start in the world of programming.

**Keywords:** Algorithms, Programming, Basic Concepts, Literature Study

**ABSTRAK.** Algoritma ialah langkah-langkah yang disusun secara logis serta sistematis buat menuntaskan suatu persoalan. Dalam dunia pemrograman sederhana, algoritma merupakan tahap awal dan ditulis sebelum program ditulis. Salah satu permasalahan yang dapat diselesaikan dengan pemrograman komputer adalah permasalahan yang melibatkan perhitungan matematis. Penelitian ini bertujuan untuk mengembangkan pemahaman algoritma serta mengembangkan keterampilan pemrograman, terutama untuk pemula. Adapun metode yang digunakan pada penelitian ini ialah metode studi literatur atau tinjauan pustaka yaitu menganalisis beberapa kajian yang berkaitan dengan topik yang dibahas, khususnya memperkenalkan konsep dasar algoritma pemrograman, karakteristik algoritma pemrograman, pengelompokan bahasa pemrograman berdasarkan tingkatannya dan notasi algoritma ialah beberapa hal dasar yg wajib diketahui oleh setiap orang yg ingin menciptakan suatu program. Pemahaman ini akan sangat membantu dalam pengembangan proyek pemrograman yang lebih kompleks. Oleh karena itu, pembelajaran algoritma pemrograman dasar sangat krusial atau penting bagi siapa pun yang ingin memulai dalam dunia pemrograman.

**Kata Kunci:** Algoritma, Pemrograman, Konsep Dasar, Studi Literatur

### PENDAHULUAN

Dengan berkembangnya ilmu pengetahuan, manusia kini bisa menghasilkan karya yang semakin canggih. Seiring perkembangan teknologi, mereka terus mencari perumusan terbaik untuk algoritma yg semakin sistematis, logis, dan optimal. Zaman sekarang, manusia sudah berhasil menghasilkan komputer yang canggih memakai kecerdasan buatan (AI). Komputer tak bisa merampungkan persoalan secara otomatis. Proses ini terdiri dari sejumlah

langkah, dimulai dengan analisis persoalan serta pembuatan algoritma penyelesaian oleh analis. Selanjutnya, programmer menerjemahkan atau menulis algoritma ke dalam bahasa program komputer tertentu, kemudian langkah-langkah ini lalu dihimpun serta dijalankan untuk membentuk hasil yg diinginkan. Pada bidang pemrograman, desain program yang sempurna bahkan menyampaikan komentar untuk setiap prosedur membantu programmer lain untuk mempertahankan serta memperbaiki fungsi program pada masa depan.

Misalkan teman Anda mengatakan kepada Anda tiga angka berbeda. Dapatkan Anda menemukan angka mana yang paling besar di antara tiga angka tersebut? Anda pasti dapat menemukan angka terbesar dengan cepat, akan tetapi apabila teman Anda bertanya bagaimana Anda menemukan angka terbesar sebelumnya, dapatkah Anda menjelaskan dengan urut? Mungkin sulit untuk menjelaskan. Dalam hal ini, kemampuan dalam pemahaman algoritma memiliki peran yang sangat penting (Fayeldi et al., 2016).

Pada prinsipnya fungsi utama suatu algoritma adalah menyelesaikan masalah. Algoritma pemrograman memberikan manfaat dan fungsi penting dalam pemrograman. Di bawah ini adalah fungsi algoritma:

1. Pemrogram dapat memecahkan masalah pemrograman kompleks yang mungkin memerlukan perhitungan berkualitas tinggi. Hal tersebut dapat dikurangi menggunakan algoritma pada pemrograman, karena perhitungan sering kali menyebabkan program berkinerja buruk karena kesalahan.
2. Algoritma pemrograman bisa melugaskan suatu program mulai program yang besar menjadi program yang lebih lugas kemudian dapat digunakan dengan lebih efisien.
3. Yang unik dari algoritme adalah algoritme ini tidak dapat digunakan hanya sekali, yang berarti dapat digunakan berkali-kali, sehingga memberi Anda keuntungan dalam meminimalkan penulisan ulang. Jadi Anda tidak perlu khawatir untuk menulis ulang program yang sama di kemudian hari, sehingga membuat pemrograman menjadi lebih mudah.
4. Saat membuat suatu program pasti terjadi error, hal ini merupakan fenomena yang wajar. Menggunakan fungsi algoritma akan memudahkan Anda menemukan kesalahan dan memperbaikinya dengan lebih cepat.
5. Dengan alur algoritma pemrograman yang jelas, kita bisa dengan mudah menemukan kesalahan jika terjadi error. Karena programnya terstruktur atau terorganisir, memudahkan Anda menemukan dan mendeteksi problem yang terjadi. (Mardiah, 2021)

Sebelum kita membahas lebih lanjut tentang algoritma dan pemrograman, kemampuan seseorang dalam melogika dan membuat tahapan untuk membereskan berbagai persoalan dengan cara yang efektif dan efisien akan sangat memengaruhi kualitas program yang dibuatnya (Eko Susanto, 2019).

Algoritma adalah tahapan-tahapan yang tersusun dengan logis dan sistematis untuk menyelesaikan suatu problem. Dalam dunia pemrograman sederhana, algoritma merupakan tahap awal dan ditulis sebelum program ditulis. Salah satu permasalahan yang dapat diselesaikan dengan pemrograman komputer adalah permasalahan yang melibatkan perhitungan matematis (Umam, 2021).

Algoritma merupakan inti dari ilmu computer atau komputasi. Berbagai bidang ilmu computer berarah pada istilah algoritma itu sendiri. Akan tetapi, jangan berasumsi bahwa algoritma harus identik dengan komputasi belaka. Pada kehidupan sehari-hari manusia, kita juga memiliki proses algoritmik. Contoh langkah atau metode pembuatan donat atau masakan yg dijelaskan pada resep disebut jua dengan algoritma. Berbagai resep harus memiliki tahapan-tahapan untuk menyiapkan hidangan. Jika langkah atau cara memasaknya tidak logis atau masuk akal, hidangan yang diinginkan tidak akan membuahkan hasil. Kita akan mencoba resepnya, jadi pertama-tama kita akan membaca resepnya dengan langkah-langkah persiapan langkah demi langkah. Kemudian akan melalui proses memasak berdasarkan apa yang dibaca. Secara umum benda yang membantu kita dalam melakukan proses pengolahan pangan disebut dengan prosesor. Prosesor bisa berupa manusia, robot, computer, atau perangkat elektronik yang lain. Prosesor akan mengeksekusi suatu langkah proses dengan menerapkan algoritma secara teratur dan tepat sesuai target (Umam, 2021).

Program dan Bahasa Pemrograman ialah sebuah kata yang tidak mampu dipisahkan. program ialah set intruksi dan bahasa pemrograman merupakan intruksi baku pada pembuatan program. program merupakan satu set intruksi yg berkode yang bisa dimengerti sang computer buat memecahkan persoalan atau membentuk akibat yang diinginkan. Programmng Language merupakan bahasa formal yg terdiri set intruksi buat komputer yg membentuk keluaran. Bahasa Pemrograman dipergunakan dalam pemrograman personal komputer untuk mengimplementasikan algoritma. Bahasa pemrograman memungkinkan pemrogram untuk memilih secara absolut data apa yang akan diproses sang komputer. Hanya ada satu bahasa pemrograman yang bisa dipahami dan dijalankan sang personal komputer mana pun, yaitu kode biner orisinal “0” serta “1”(Kani, 2020).

Untuk menyelesaikan suatu masalah, algoritma terdiri dari tahap-tahap yang tersusun dengan sistematis dan logis. Algoritma memiliki lima ciri, yaitu finiteness (keterbatasan),

definiteness (kepastian), input (masukan), output (keluaran), dan effectiveness (efektivitas). mengelompokkan bahasa pemrograman menurut tingkat keterampilannya yaitu bahasa pemrograman tingkat rendah, tingkat menengah, & tingkat tinggi (Umam, 2021).

Pemrograman komputer adalah keterampilan yang semakin penting di dunia modern. Dengan pesatnya perkembangan teknologi informasi, pemahaman dasar tentang algoritma pemrograman menjadi kunci untuk memahami dan menciptakan perangkat lunak yang inovatif. Di zaman di mana komputasi telah menjadi bagian integral dari hampir setiap aspek kehidupan sehari-hari, kemampuan merancang, mengembangkan, dan memahami perangkat lunak menjadi hal yang penting. Mempelajari pemrograman adalah langkah awal yang sangat penting untuk memahami dasar-dasar ilmu komputer. Melalui pembelajaran pemrograman, suatu individu dapat memahami algoritma, logika pemrograman, dan struktur data dasar. Artikel ini dimaksudkan sebagai pengenalan dasar algoritma pemrograman dalam rangka pembelajaran pemrograman, dengan penekanan pada pemahaman dasar yang diperlukan bagi pemula.

Dalam pembelajaran pemrograman, pemahaman konsep algoritmik menjadi dasar untuk menciptakan program komputer yang efektif dan efisien. Artikel ini akan menjelaskan dasar-dasar pengembangan algoritma, termasuk pemahaman aliran kontrol, pengambilan keputusan, dan iterasi. Selain itu, artikel ini juga akan mengilustrasikan konsep-konsep tersebut melalui contoh sederhana. Pembelajaran pemrograman perlu memberikan landasan yang kokoh bagi siswa, baik pemula maupun ingin memperdalam ilmunya. Oleh karena itu, artikel ini juga akan merinci konsep algoritma, notasi, ciri, sifat, dan struktur dasar pembelajaran algoritma, serta prosedur pemrograman untuk menjalankan algoritma. Penilitain ini juga akan menjelaskan perbedaan algoritma dengan pemrograman dan bagaimana algoritma berhubungan satu sama lain. Di era komputasi yang terus berubah, memahami algoritma pemrograman dasar adalah kunci untuk mengembangkan solusi yang efektif dan inovatif. Nah, artikel ini bertujuan untuk memberikan panduan awal bagi mereka yang ingin memulai pemrograman komputer.

## **METODE**

Dalam penelitian ini penulis menggunakan metode studi literatur atau tinjauan pustaka dengan meninjau beberapa kajian yang berkenaan dengan tema yang dibahas, khususnya memperkenalkan konsep dasar algoritma pemrograman. Berbagai sumber rujukan yang menjadi pokok bahasan berasal dari buku, jurnal, majalah dan sumber yang lain. Meliputi

ulasan mengenai konsep algoritmik, notasi algoritma, ciri-ciri, sifat-sifat, dan struktur dasar pembelajaran algoritma, serta langkah-langkah implementasi algoritma pemrograman.

Pengambilan data dilakukan sejak 15 Oktober 2023, kemudian dilakukan pengolahan data dan didapatkan beberapa buku dan artikel dengan rentang 5 tahun terakhir (2018-2023). Berikut alur atau langkah-langkah metode pengumpulan dan pengolahan data yang dilakukan :



**Gambar 1.1 Metode pengumpulan dan pengolahan data**

Berikut penjelasan dari tahapan-tahapan atau langkah-langkah metode pengumpulan dan pengolahan data pada gambar 1.1 :

1. Pengumpulan data yaitu peneliti mengidentifikasi topik utama, yaitu "Pengenalan Algoritma Pemrograman Dasar," serta tujuan penelitian, yang mungkin mencakup evaluasi metode pembelajaran yang digunakan dalam konteks pembelajaran pemrograman awal. Kemudian peneliti melakukan pengumpulan data dengan melakukan pencarian sumber literatur melalui basis data jurnal ilmiah, perpustakaan online, dan sumber-sumber akademik terkait lainnya.
2. Menyeleksi dan menganalisis sumber yaitu pada tahap ini, peneliti melakukan seleksi sumber-sumber literatur yang paling relevan dan berkualitas tinggi mengenai topik utama dengan mempertimbangkan faktor seperti tahun publikasi dengan rentang 5 tahun terakhir, metode penelitian yang digunakan, dan relevansi dengan topik. Kemudian Setiap sumber literatur yang dipilih akan dianalisis dengan cermat.
3. Sintesis Informasi yaitu dalam tahap ini peneliti menggabungkan temuan dari berbagai sumber literatur untuk memahami tren, perbandingan, atau perbedaan dalam metode pembelajaran algoritma pemrograman dasar dalam konteks pemrograman awal.
4. Kesimpulan yaitu artikel ini akan menyajikan kesimpulan berdasarkan temuan dari studi literatur, serta mungkin memberikan rekomendasi atau implikasi untuk pengajaran algoritma pemrograman dasar.

Dengan menggunakan metode penelitian studi literatur, artikel ini akan memberikan wawasan yang mendalam tentang berbagai metode pembelajaran algoritma pemrograman dasar dalam konteks pembelajaran pemrograman awal.

## PEMBAHASAN

### Konsep Algoritma

Secara umum, algoritma adalah cara dan urutan langkah atau tahapan yang diatur secara sistematis untuk menyelesaikan suatu kegiatan sehingga proses yang akan diselesaikan sesuai dengan algoritma atau urutan langkah yang telah disusun. Pada dasarnya, algoritma sebuah proses terdiri dari tiga tahapan, seperti yang ditunjukkan pada Gambar 1.2. Proses ini menerima inputan, dan kemudian menghasilkan *output* atau keluaran (Eko Susanto, 2019)



**Gambar 1.2 Prinsip Algoritma**

### Kriteria Algoritma

Ada beberapa kriteria atau ciri-ciri yang harus dipenuhi oleh suatu algoritma untuk dianggap sebagai algoritma yang baik:

1. *Finiteness* (keterbatasan) : Algoritma harus beroperasi dalam waktu terbatas, yang berarti harus menyelesaikan tugas dalam jumlah langkah yang wajar atau dalam waktu yang dapat diterima. Kualitas algoritma sangat dipengaruhi oleh efisiensi.
2. *Definiteness* (kepastian) : Karena algoritma harus bersifat deterministik, mereka selalu menghasilkan output yang sama dengan input yang sama. Ini penting dalam pemrograman dan komputasi karena kita perlu dapat memprediksi hasil algoritma.
3. *Input* (masukan) : Dalam menentukan jenis dan format input, algoritma harus jelas. Ini mencakup jenis data yang dapat diolah (seperti bilangan bulat, teks, atau gambar), serta batasan atau syarat khusus untuk input, seperti rentang nilai atau format yang valid.
4. *Output* (keluaran) : Algoritma harus memberikan hasil yang diharapkan dan jelas, yang mencakup jenis data output, format output, dan cara output dibuat dari input yang diberikan. Hasil harus benar dan relevan untuk masalah yang diselesaikan.
5. *Effectiveness* (efektivitas) : Algoritma harus dirancang untuk mencapai tujuan dan memberikan solusi yang efisien atau tepat. Selain itu, algoritma harus mempertimbangkan penggunaan sumber daya, seperti memori dan waktu, dengan efisien. (Saputro & Pradana, 2022).

## Struktur Algoritma

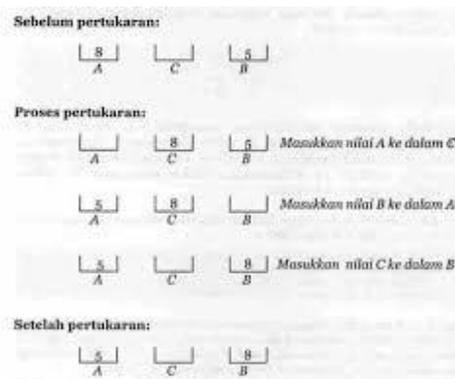
Algoritma Secara umum memiliki 3 struktur dasar yaitu Perulangan, Percabangan dan terurut atau sequential:

### 1. Runtutan (*sequence*)

Pola yang diterapkan untuk menjalankan program dengan pernyataan berurutan (*sequential*). Pola ini memuali prosedur yg diberikan dengan berurutan atau beruntun dari permulaan sampai akhir. Pola yang berurut tidak mengandung pengulangan atau lonjakan. Runtutan memiliki ciri-ciri sebagai berikut:

- a. Setiap arahan dilaksanakan satu demi satu sebanyak sekali
- b. Pelaksanaan arahan dilaksanakan dengan terurut.
- c. Algoritma berakhir dengan arahan akhir.
- d. Beberapa hasil dapat dihasilkan jika urutan diubah.

Salah satu contoh penerapan algoritma penukaran 2 bilangan bulat adalah:



**Gambar 1.3 Algoritma penukaran 2 bilangan bulat**

### 2. Pemilahan (*selection*)

Dalam algoritma, seleksi adalah peristiwa baru yang akan terjadi ketika suatu persyaratan dapat dipenuhi. Salah satu definisi kondisi adalah bahwa suatu instruksi hanya dapat dijalankan jika suatu kondisi dapat dipenuhi, dan suatu instruksi tidak dapat dijalankan jika suatu kondisi tidak dapat dipenuhi. Salah satu contoh suatu kondisi adalah menentukan apakah suatu bilangan ganjil atau genap. (Zein, Afrizal, 2020).

- a. Masukkan bilangan
- b. kemudian masukkan bilangan (-1).
- c. Bagi dengan dua
- d. Angka tersebut genap jika hasilnya nol.

e. Print keterangan ganjil atau genap

```
{ Judul }
#Meriksa_angka_genap_atau_ganjil

{ Kasus }
angka: Integer
keterangan: String

{ Algoritma }
Input(angka);

{ Pemilihan - 1 Kasus }
keterangan <- 'Angka ganjil' // Inisialisasi

If angka mod 2 == 0 Then
    keterangan <- 'Angka genap'

Output(keterangan)
```

**Gambar 1.4** Bilangan ganjil dan genap

3. Perulangan (repetition)

Pola pengulangan adalah pola yg mengulangi satu baris atau satu blok baris program berulang kali. Perulangan akan dilakukan sesuai yang dibutuhkan.. Ada contoh pengulangan untuk mencari kategori usia dengan menerapkan pengulangan if... else... (Retta et al., 2019)

- a. Masukkan bilangan usia
- b. Apabila umurnya kurang dari 45 tahun, maka "masih muda"
- c. Apabila umurnya kurang dari 45 tahun, maka "sudah tua"

```
umur = input("masukan umur: ")
if int(umur) < 45:
    keterangan = 'masih muda'
else:
    keterangan = 'sudah tua'

print('anda memasukan umur {},
artinya {}'.format(umur,
keterangan))
```

**Gambar 1.5** Penghitungan Umur

### Bahasa Pemrograman

Bahasa pemrograman merupakan sekumpulan pedoman sintaksis dan semantik yang dipergunakan untuk membentuk program komputer. Ini adalah pedoman baku yang digunakan untuk mengarahkan komputer dalam menjalankan tugas-tugas tertentu. Dengan bahasa ini, programmer memiliki kemampuan untuk menentukan jenis data yang akan diperlakukan oleh komputer, cara data tersebut disimpan atau dikomunikasikan, dan langkah-langkah apa yang harus diambil dalam berbagai situasi yang mungkin timbul. (Saragih, 2018).

Bahasa pemrograman, yang sering disebut sebagai "bahasa komputer" atau "bahasa pemrograman komputer," adalah standar umum yang digunakan untuk mengendalikan komputer. Ini terdiri dari sekumpulan aturan sintaks dan semantik yang digunakan untuk menciptakan program komputer. (Premana et al., 2022).

### **Tingkatan Bahasa Pemrograman**

Klasifikasi bahasa pemrograman tidak baku. Secara umum, ada 3 (tiga) cara pengklasifikasian bahasa pemrograman, yaitu bahasa pemrograman tingkat rendah, tingkat menengah, dan tingkat tinggi. (Ain et al., 2022)

#### **1. Bahasa Pemrograman Tingkat Rendah**

Bahasa ini berkaitan dengan instruksi perangkat keras komputer dan memberikan kontrol tinggi terhadap perangkat keras. Meskipun kuat, bahasa ini sering sulit untuk dimengerti dan digunakan.

*Contoh:* Bahasa rakitan (Assembly language) adalah contoh bahasa pemrograman tingkat rendah yang memerintah langsung perangkat keras dengan kode biner atau heksadesimal.

#### **2. Bahasa Pemrograman Tingkat Menengah**

Bahasa ini mencapai keseimbangan antara bahasa rendah dan tinggi. Mereka memungkinkan kontrol perangkat keras yang lebih besar daripada bahasa tingkat tinggi sambil menjaga kemudahan penggunaan.

*Contoh:* Bahasa C dan C++ adalah bahasa pemrograman tingkat menengah yang memungkinkan kontrol lebih langsung atas perangkat keras, tetapi dengan sintaks yang lebih manusiawi.

#### **3. Bahasa pemrograman Tingkat Tinggi**

Bahasa ini memberikan abstraksi tinggi dari perangkat keras komputer, memungkinkan programmer fokus pada pemecahan masalah dan algoritma tanpa perlu terlalu memikirkan detail perangkat keras.

*Contoh:* Python, Java, C#, dan JavaScript adalah contoh bahasa pemrograman tingkat tinggi yang mudah digunakan, dengan sintaks yang lebih jelas dan banyak pustaka serta alat bantu pengembangan.

### **Penulisan Algoritma dalam Pemrograman**

Karena notasi algoritma berisi kerangka dasar yang akan digunakan untuk membangun suatu program, sangat penting bagi setiap orang yang membangun program untuk memahaminya. Penjelasan tahapan algoritma tidak bergantung pada sintaks pemrograman atau spesifikasi komputer yang menjalankannya. Dalam menulis algoritma, belum ada aturan

mutlak, yg penting adalah algoritma mesti bisa dipahami dan dibaca dengan mudah. Namun, notasi algoritma harus diperhatikan untuk mencegah kekeliruan dan salah pemahaman. Secara umum, algoritma ditulis dalam tiga cara yaitu *deskriptif*, *pseudocode*, dan *flowchart*. (Nuraini, 2018)

### 1. Algoritma Deskriptif

Notasi algoritma yang diberikan melalui kalimat deskriptif dikenal sebagai notasi alami. Notasi algoritma deskriptif melibatkan penulisan instruksi menggunakan kalimat deskriptif yang mudah dimengerti. Ini adalah pendekatan yang lebih cocok untuk algoritma yang tidak terlalu panjang. Secara umum, notasi deskriptif terdiri dari tiga komponen utama.

- a. Bagian judul (*header*) adalah bagian yang berisi spesifikasi atau penjelasan tentang algoritma tersebut. Pada bagian ini, Anda akan menemukan penjelasan teks terkait algoritma, yang bisa mencakup program, prosedur, atau fungsi. Nama algoritma sebaiknya singkat tetapi informatif.

```

Algoritma Luas_Lingkaran ← Judul Algoritma
(Menghitung luas lingkaran untuk ukuran jari-jari tertentu. Algoritma
menerima masukan jari-jari lingkaran, menghitung luasnya, lalu cetak
luasnya ke piranti keluaran) ← Spesifikasi
    
```

Gambar 2.1 Judul

- b. Bagian deklarasi (kamus) yaitu semua nama yang digunakan dalam algoritma dimasukkan ke dalam bagian kamus, yang dapat mencakup variable, konstanta, tipe, prosedur, atau nama fungsi.

```

Deklarasi :
{Nama Type variabel yang bukan tipe data dasar}
type waktu:<hh:mm:ss: integer> {Type waktu terdiri dari 3 data
masukan yaitu "hh" sebagai jam, "mm" sebagai menit, dan "ss"
sebagai detik}

{Nama konstanta, harus menyebutkan tipe data dan nilai}
constant PHI: real = 3.141592653589793
constant E: real = 2.718281828459045

{Nama variabel yang menggunakan tipe data dasar}
nama      : String   {suatu nilai yang merupakan kumpulan karakter}
ketemu    : boolean  {suatu nilai logika (true atau false)}
beratBadan : real    {suatu nilai bilangan pecahan}
jumlahAnak : integer {suatu nilai bilangan bulat}
    
```

Gambar 2. deklarasi

```

function tambah(x:int, y:int): int
{proses: menambahkan dua nilai data dan hasil penambahan sebagai
nilai keluaran fungsi}
    
```

- c. Bagian deskripsi merupakan komponen utama dari struktur algoritma dan mencakup penjelasan tentang langkah-langkah yang diperlukan untuk menyelesaikan masalah. Setiap bagian di atas disertai dengan informasi tambahan untuk memperjelas makna teks.

```

Algoritma Luas_Lingkaran
{Menghitung luas lingkaran untuk ukuran jari-jari tertentu.
Algoritma menerima masukan jari-jari lingkaran, menghitung
luasnya, lalu cetak luasnya ke piranti keluaran}

Deklarasi :
jari_jari = real {tipe data bilangan pecahan}
luas = real {tipe data bilangan pecahan}
PHI = 3.14

Deskripsi:
1. Baca jari_jari
2. Hitung luas = PHI*jari_jari * jari_jari
3. Tampilkan luas ke layar
4. Selesai

```

**Gambar 2.3 Deskripsi**

Contoh berikut adalah algoritma deskriptif yang disusun secara sistematis dan efisien. Penyertaan tanda-tanda bertujuan untuk meningkatkan efektivitas penulisan dan menjadikan tujuan algoritma lebih mudah dipahami.

1. Masukkan a, b, dan c.
2.  $mak \leftarrow a$ .
3. Jika  $b > mak$ , kerjakan langkah ke-4. Jika tidak, kerjakan langkah ke-5.
4.  $mak \leftarrow b$ .
5. Jika  $c > mak$ , kerjakan langkah ke-6. Jika tidak, kerjakan langkah ke-7.
6.  $mak \leftarrow c$ .
7. Tulis mak.

**Gambar 2.4 Contoh algoritma deskriptif**

## 2. Flowchart

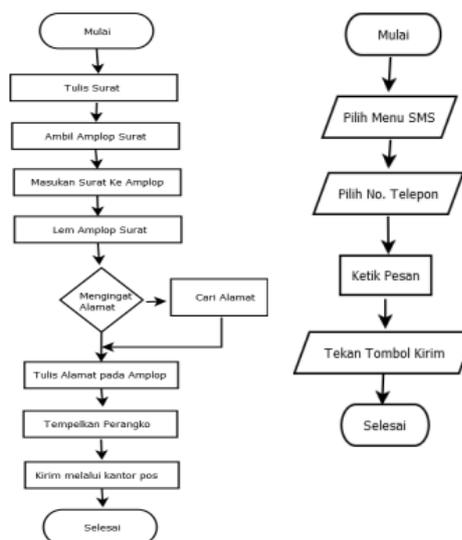
Flowchart atau diagram alir yang menunjukkan arah alur program adalah ilustrasi algoritma dalam suatu program. digunakan untuk dokumentasi, analisis proses, dan bantuan komunikasi. Flowchart adalah representasi visual pseudocode yang lebih sederhana. Bagan flowchart program menunjukkan susunan logika proses program sebagai pemecahan masalah. (Munazilin, 2021). Adapun simbol-simbol flowchart yaitu :

SIMBOL	NAMA	FUNGSI
	<b>TERMINATOR</b>	Permulaan/akhir program
	<b>GARIS ALIR (FLOW LINE)</b>	Arah aliran program
	<b>PREPARATION</b>	Proses inialisasi/pemberian harga awal
	<b>PROSES</b>	Proses perhitungan/proses pengolahan data
	<b>INPUT/OUTPUT DATA</b>	Proses input/output data, parameter, informasi
	<b>DECISION</b>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<b>ON PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	<b>OFF PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

**Gambar 2.5** Simbol-simbol flowchart

Keuntungan dari metode penulisan algoritma dengan flowchart adalah sebagai berikut: Memudahkan pembaca algoritma untuk membaca dan memahaminya, sangat cocok untuk digunakan dalam studi kasus algoritma dengan kerumitan rendah, dan aliran proses programnya sangat rinci karena dimodelkan secara visual dengan simbol. Namun, flowchart tidak hanya mudah digunakan, tetapi juga memiliki kekurangan. Di antaranya adalah bahwa mereka tidak cocok untuk menampilkan alur program yang kompleks. Mereka juga membutuhkan alat khusus untuk merancang algoritmanya. Selain itu, penjelasan alur proses tidak detail karena keterbatasan ruang. (Budiman, 2021)

Contoh algoritma yang di tulis dengan *flowchart*:  
**Algoritma mengirimkan surat:**      **Algoritma mengirimkan SMS:**



**Gambar 2.6** Contoh Algoritma flowchart

### 3. Pseudocode

Pseudocode adalah suatu metode penulisan algoritma yang meniru gaya bahasa pemrograman tingkat tinggi. Secara umum, notasi pseudocode menggunakan bahasa yang lebih mudah dimengerti dan ringkas jika dibandingkan dengan bahasa pemrograman. Pseudocode berisi deskripsi algoritma pemrograman komputer yang menggunakan elemen-elemen dasar dari beberapa bahasa pemrograman, tetapi bahasa tersebut hanya dimaksudkan untuk dibaca dan dipahami oleh manusia. Oleh karena itu, pseudocode tidak dapat dipahami oleh komputer. (Vivin Ayu Lestari et al., 2022)

Notasi pseudocode harus diterjemahkan ke dalam sintaks bahasa pemrograman tertentu agar komputer dapat memahaminya. Tidak ada aturan resmi yang berlaku untuk notasi pseudocode. Sangat dianjurkan dengan menerapkan *key word* yang sering digunakan, seperti *if, then, else, while, do, for, repeat*, dan sebagainya. Pedoman untuk penulisan pseudo kode adalah sebagai berikut: Memiliki bagian header atau kepala yang menunjukkan judul, komentar, dan deklarasi algoritma kemudian memiliki badan algoritma lalu memiliki bagian akhir algoritma yang menunjukkan bagian akhir algoritma dan yang terakhir komentar dimulai dengan karakter "{" dan berakhir dengan karakter "}". (Syamsiah, 2019)

Berikut contoh dari pseudocode :

<pre> program hitung_luas_segitiga </pre>	} Judul
<pre> deklarasi var luas, alas, tinggi : integer; </pre>	} Deklarasi
<pre> algoritma:   alas &lt;-- 5;   tinggi &lt;-- 10;   luas &lt;-- (alas * tinggi)/2;   write(luas); </pre>	} Deskripsi / Algoritma

```

program hitung_luas_segi_panjang

deklarasi
  var panjang,lebar,luas:integer;

algoritma:
  read(panjang);
  read(lebar);

luas <-- panjang * lebar;

print(luas);

```

**Gambar 2.7** Contoh pseudocode

Apabila dibandingkan dengan algoritma berbahasa alami, pseudocode cenderung lebih sederhana untuk dipahami. Algoritma bahasa alami, di sisi lain, lebih praktis untuk digunakan karena mudah untuk mengatur hubungan antar bloknnya dan mudah dipahami. (Maria et al., 2023)

## KESIMPULAN

Algoritma adalah panduan langkah demi langkah yang membantu pemula merancang program dan memecahkan masalah. Seseorang dapat belajar pemrograman dengan pemahaman yang kuat tentang algoritma dasar. Algoritma adalah serangkaian langkah yang diatur secara terstruktur dan logis dengan tujuan menyelesaikan suatu permasalahan. Algoritma memiliki lima aspek kunci yang mencakup Keterbatasan (*Finiteness*), Kepastian (*Definiteness*), Masukan (*Input*), Keluaran (*Output*), dan Efektivitas (*Effectiveness*). Bahasa pemrograman dikelompokkan ke dalam tiga tingkatan, yaitu tingkat rendah, menengah, dan tinggi. Karena notasi algoritma berisi kerangka dasar yang akan digunakan untuk membangun suatu program, sangat penting bagi setiap orang yang membangun program untuk memahaminya. Tiga cara umum untuk menulis algoritma adalah deskriptif, pseudocode, dan flowchart. Pemahaman ini akan sangat membantu dalam pengembangan proyek pemrograman yang lebih kompleks. Oleh karena itu, kesimpulan dari artikel ini adalah bahwa pembelajaran algoritma pemrograman dasar sangat perlu bagi setiap orang yang ingin memulai dalam dunia pemrograman.

## DAFTAR PUSTAKA

- Ain, K., Ukrowiyah, N., & Yunata, E. E. (2022). Pengenalan Bahasa Pemrograman Komputer Pada Siswa Sekolah Menengah Atas ( Sma / Smk ). *Prosiding PEPADU*, 4, 115–120.
- Budiman, I. (2021). ANALISIS PENGENDALIAN MUTU DI BIDANG INDUSTRI MAKANAN. *Jurnal Inovasi Penelitian*, 1(0.1101/2021.02.25.432866), 1–15.
- Eko Susanto, W. dan A. S. (2019). *Logika & Algoritma untuk Pemula*. 1–76. [https://repository.bsi.ac.id/index.php/unduh/item/285644/Bidang-A\\_Buku-Ajar-Logika-Algoritma-a.n-Wahyu-Eko-Susanto.pdf](https://repository.bsi.ac.id/index.php/unduh/item/285644/Bidang-A_Buku-Ajar-Logika-Algoritma-a.n-Wahyu-Eko-Susanto.pdf)
- Fayeldi, T., Murniasih, T. R., & Yunus, A. (2016). *Dasar-Dasar Pemrograman Komputer Dengan Menggunakan MATLAB*.
- Kani. (2020). Modul Pengantar Algoritma dan Pemrograman. *Algoritma Dan Bahasa Pemrograman*, 1, 1–36. <https://pustaka.ut.ac.id/lib/wp-content/uploads/pdfmk/MSIM4203-M1.pdf>
- Mardiah, S. K. A. (2021). MENGANALISIS KONSEP DASAR PADA ALGORITMA PEMOGGRAMAN. *Algoritma Dan Bahasa Pemrograman*, 282.
- Maria, E., Winarso Martyas Edi, S., Suharyadi, Christianto, E., & Purwanto. (2023). Peningkatan Kompetensi Algoritma Pemrograman untuk Guru SMK Negeri 1 Salatiga. *Warta LPM*, 26(3), 245–255. <https://doi.org/10.23917/warta.v26i3.1701>

- Munazilin, A. (2021). *LOGIKA DAN ALGORITMA PEMROGRAMAN* (Cetakan Pe). CV.AA RIZKY.
- Nuraini, R. (2018). *Analisis Algoritma Definite Integration Menggunakan Metode Desk Check*. 50–55.
- Premana, A., Wijaya, A. P., Yono, R. R., & Hayati, S. N. (2022). Media Pembelajaran Pengenalan Bahasa Pemrograman Pada Anak Usia Dini Berbasis Game. *Tekinfo: Jurnal Bidang Teknik Industri Dan Teknik Informatika*, 23(2), 66–75. <https://doi.org/10.37817/tekinfo.v23i2.2597>
- Retta, A. M., Isroqmi, A., & Nopriyanti, T. D. (2019). INDIKTIKA (Jurnal Inovasi Pendidikan Matematika). *Jurnal Inovasi Pendidikan Matematika*, 2(2), 126–135.
- Saputro, N. T., & Pradana, A. E. (2022). *Modul Algoritma Dan Pemograman*. 2–23.
- Saragih, R. R. (2018). Pemrograman dan bahasa Pemrograman. *STMIK-STIE Mikroskil, December*, 1–91.
- Syamsiah, S. (2019). Perancangan Flowchart dan Pseudocode Pembelajaran Mengenal Angka dengan Animasi untuk Anak PAUD Rambutan. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 4(1), 86. <https://doi.org/10.30998/string.v4i1.3623>
- Umam, K. (2021). *Pemrograman Komputer Dengan Python*.
- Vivin Ayu Lestari, Rakhmat Arianto, Baskoro Singgih Anindito, Yuni Kurnia Taramita, Eka Larasati Amalia, & Odhitya Desta Triswidrananta. (2022). Aplikasi Pembelajaran Rekonstruksi Algoritma Pseudocode Dengan Pendekatan Element Fill-in-Blank Problems Di Pemrograman Java. *Jurnal Teknik Ilmu Dan Aplikasi*, 3(2), 153–161. <https://doi.org/10.33795/jtia.v3i1.108>
- Zein, Afrizal, & E. S. E. (2020). Algoritma dan Struktur Data. In *Fakultas Komputer* (Vol. 2, Issue 1).