

**ALGORITMA DAN FLOWCHART
DALAM MENYELESAIKAN
SUATU MASALAH**



DISUSUN OLEH

**Johan Reza Fauzi
(20330044)**

**FAKULTAS TEKNIK
TEKNIK INFORMATIKA**

**UNIVERSITAS JANABADRA
YOGYAKARTA
2020**

ABSTRAK

Bahasa pemrograman semakin banyak dipelajari oleh banyak orang. Hal ini terkait dengan kemajuan zaman yang menjadikan teknologi sebagai hal penting untuk menunjang kemajuan. Dalam matematika dan ilmu komputer, algoritma adalah urutan atau langkah-langkah untuk penghitungan atau untuk menyelesaikan suatu masalah yang ditulis secara berurutan. Sehingga algoritma pemrograman adalah urutan atau langkah-langkah untuk menyelesaikan masalah pemrograman komputer. Algoritma juga mempunyai alat bantu untuk menyelesaikan suatu masalah agar lebih mudah yaitu menggunakan *flowchart*, *flowchart* adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol.

Dalam pemrograman hal yang penting untuk dipahami adalah logika kita dalam berpikir bagaimana cara untuk memecahkan masalah pemrograman yang akan dibuat. Dalam hal ini algoritma dan *flowchart* sangat penting dalam pemecahan masalah

Kata kunci : algoritma, *flowchart*, pemrograman, *pseudocode*

DAFTAR ISI

JUDUL	i
ABSTRAK	ii
DAFTAR ISI	iii
KATA PENGANTAR	iv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	1
1.3 Tujuan	2
BAB II ISI	3
2.1 Pengertian Algoritma dan <i>Flowchart</i>	3
A. Pengertian Algoritma.....	3
B. Pengertian <i>Flowchart</i>	3
2.2 Notasi Algoritma.....	4
A. Algoritma Dalam Bentuk Kalimat Deskriptif	4
B. Algoritma Dalam Bentuk Kalimat <i>flowchart</i>	5
C. Algoritma Dalam Bentuk Kalimat <i>pseudocode</i>	6
2.3 Penerapan Algoritma dan <i>Flowchart</i> dalam menyelesaikan masalah	7
A. Algoritma	7
B. <i>Flowchart</i>	9
BAB III PENUTUP	11
3.1 Kesimpulan	11
3.2 Saran	11
DAFTAR PUSTAKA	12

KATA PENGANTAR

Puji syukur saya panjatkan kehadiran Allah SWT Tuhan Yang Maha Esa karena berkat limpahan Rahmat dan Karunia-Nya sehingga saya dapat menyusun makalah “Algoritma dan *Flowchart* Dalam Menyelesaikan Suatu Masalah” ini dengan tepat waktu.

Tidak lupa juga saya ucapkan terima kasih kepada Dosen mata kuliah “Bahasa Indonesia” saya, Ibu Zulisih Maryani, M.A. yang telah membimbing saya dalam mata kuliah yang bersangkutan.

Dalam tugas ini saya dapat menyelesaikan makalah dengan judul “Algoritma dan *Flowchart* Dalam Menyelesaikan Suatu Masalah” ini dibuat dalam rangka memenuhi tugas mata kuliah Bahasa Indonesia. Semoga makalah yang saya buat ini dapat bermanfaat bagi saya dan semua pihak yang membacanya.

Demikian kata pengantar ini saya buat. Saya menyadari bahwa makalah ini masih sangat jauh dari kata sempurna, untuk itu saya mohon maaf bila ada kesalahan kata dalam pembuatan makalah ini maupun kata pengantar ini, juga saya meminta kritik dan saran yang membangun agar dapat dibuatnya makalah yang lebih baik. Semoga makalah ini bermanfaat bagi para pembaca dan dapat menambah cakrawala pengetahuan kita.

Yogyakarta, 3 November 2020

penulis

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi dari tahun ke tahun, terutama teknologi komputer sudah tak bisa dipungkiri lagi, karena sudah banyak bisa kita lihat dari sekeliling kita perkembangan apa yang sudah terjadi. Dibalik semua itu, ada banyak tahapan saat seorang *programmer* merancang program. Salah satu yang paling penting adalah tahap perancangan program. Tahapan ini dikatakan sebagai tahapan yang paling penting karena pada tahapan ini dasar dari program akan dibuat (rancangan program).

Permasalahan dapat diselesaikan dengan langkah-langkah penyelesaian yang dibuat berisi urutan langkah-langkah penyelesaian masalah yang ditulis ke dalam bahasa komputer atau yang biasa disebut bahasa pemrograman, lalu komputer menjalankan perintah-perintah yang tertulis dalam program tersebut. Langkah-langkah penyelesaian masalah yang dibuat oleh disebut sebagai algoritma. Algoritma yang ditulis ke dalam bahasa komputer disebut program komputer.

Masalah yang ada dibuat rancangan program bisa juga menggunakan alat bantu, yaitu *flowchart*. *Flowchart* dalam Bahasa Indonesia diterjemahkan sebagai Diagram Alir. Dari dua kata ini, maka dapat kita bayangkan bahwa *flowchart* itu berbentuk diagram yang bentuknya dapat mengalirkan sesuatu. Hal ini memang benar, *flowchart* memang melukiskan suatu aliran kegiatan dari awal hingga akhir mengenai suatu langkah-langkah dalam penyelesaian suatu masalah.

1.2 Rumusan Masalah

Adapun rumusan masalah adalah sebagai berikut :

1. Apa itu algoritma dan *flowchart*?
2. Bagaimana contoh menyelesaikan suatu masalah menggunakan algoritma dan *flowchart*?
3. Apakah algoritma dan *flowchart* saling berhubungan?

1.3 Tujuan

Tujuan dari penulisan makalah ini antara lain :

1. Untuk memenuhi tugas mata kuliah Bahasa Indonesia
2. Menjelaskan penyelesaian suatu masalah menggunakan algoritma dalam kehidupan sehari-hari
3. Menjelaskan pengertian algoritma dan *flowchart*
4. Untuk belajar bagaimana menyelesaikan suatu masalah menggunakan *flowchart*

BAB II

ISI

2.1 Pengertian Algoritma dan *Flowchart*

A. Pengertian Algoritma

Menurut Tominanto dan Subinarto dalam buku *Algoritma dan Pemrograman* (2018:3) ditinjau dari asal usul katanya kata Algoritma sendiri mempunyai sejarah yang aneh. Orang hanya menemukan kata *Algorism* yang berarti proses menghitung dengan angka arab. Anda dikatakan *Algorist* jika anda menghitung menggunakan Angka Arab. Para ahli bahasa berusaha menemukan asal kata ini namun hasilnya kurang memuaskan. Akhirnya para ahli sejarah matematika menemukan asal kata tersebut yang berasal dari nama penulis buku arab yang terkenal yaitu Abu Ja'far Muhammad Ibnu Musa Al-Khuwarizmi.

Al-Khuwarizmi dibaca orang barat menjadi *Algorism*. Al-Khuwarizmi menulis buku yang berjudul Kitab Al Jabar Wal-Muqabala yang artinya “Buku pemugaran dan pengurangan” (*The book of restoration and reduction*). Dari judul buku itu kita juga memperoleh akar kata “Aljabar” (*Algebra*). Perubahan kata dari *Algorism* menjadi *Algorithm* muncul karena kata *Algorism* sering dikelirukan dengan *Arithmetic*, sehingga akhiran *-sm* berubah menjadi *-thm*. Karena perhitungan dengan angka Arab sudah menjadi hal yang biasa. Maka lambat laun kata *Algorithm* berangsur-angsur dipakai sebagai metode perhitungan (komputasi) secara umum, sehingga kehilangan makna kata aslinya. Dalam Bahasa Indonesia, kata *Algorithm* diserap menjadi Algoritma.

B. Pengertian *Flowchart*

Flowchart adalah cara penulisan algoritma dengan menggunakan notasi grafis. *Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan antara proses digambarkan dengan garis penghubung. Dengan menggunakan flowchart akan memudahkan untuk melakukan pengecekan bagian-bagian yang terlupakan

dalam analisis masalah. *Flowchart* dapat membantu analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan membantu dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

Flowchart didefinisikan juga sebagai bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* dapat juga merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut (Tominanto dan Subinarto, 2018:80)

2.2 Notasi Algoritma

A. Algoritma Dalam Bentuk Kalimat Deskriptif

“Algoritma dengan uraian kalimat deskriptif adalah notasi algoritma yang paling sederhana karena algoritma ini menggunakan bahasa sehari-hari” (Suprpto dkk, 2008:53). Untuk permasalahan yang sederhana penggunaan notasi ini sangatlah mudah, akan tetapi untuk permasalahan yang lebih kompleks dan rumit, penggunaan notasi ini akan lebih sulit dan sering kali terjadi ambigu dalam langkah-langkah penyelesaian masalah. Oleh karena itulah untuk kasus-kasus yang lebih kompleks, penggunaan notasi ini jarang sekali bahkan tidak digunakan.

Permasalahan di atas, yaitu mencari bilangan terbesar dari tiga buah bilangan yang dimasukkan, tergolong permasalahan yang sederhana, jadi algoritmanya masih mudah dan dapat dijelaskan dengan uraian kalimat deskriptif sebagai berikut:

1. Masukkan sembarang bilangan sebanyak tiga buah.
2. Ambil bilangan pertama dan set maksimum-nya sama dengan bilangan pertama.
3. Ambil bilangan kedua dan bandingkan dengan maksimum.
4. Apabila bilangan kedua lebih besar dari maksimum maka ubah maksimum-nya menjadi sama dengan bilangan kedua.
5. Ambil bilangan ketiga dan bandingkan dengan maksimum.
6. Apabila bilangan ketiga lebih besar dari maksimum maka ubah lagi maksimum-nya menjadi sama dengan bilangan ketiga.
7. Variabel maksimum akan berisi bilangan yang terbesar dan tampilkan hasilnya

Algoritma dengan uraian kalimat deskriptif seperti di atas sudah jarang sekali kita temukan karena kadang kala agak sulit untuk memahaminya. Yang paling banyak kita temukan adalah algoritma (dengan uraian kalimat deskriptif) yang ditulis secara lebih sistematis dan efisien sehingga lebih mudah untuk memahaminya. Algoritma tersebut adalah sebagai berikut:

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

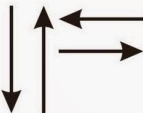











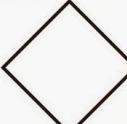


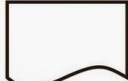
B. Algoritma Dalam Bentuk Kalimat *flowchart*

Seperti yang di tuangkan oleh Fadilah di dalam makalah *Flowchart* (2020:3) membuat algoritma dalam bentuk *flowchart* ada beberapa petunjuk yang harus diperhatikan, seperti :

1. Flowchart digambarkan dari halaman atas ke bawah dan dari kiri ke kanan
2. Aktivitas yang digambarkan harus didefinisikan secara hati-hati dan definisi ini harus dapat dimengerti oleh pembacanya.
3. Kapan aktivitas dimulai dan berakhir harus ditentukan secara jelas
4. Setiap langkah dari aktivitas harus diuraikan dengan menggunakan deskripsi kata kerja, misalkan MENGHITUNG PAJAK PENJUALAN.
5. Setiap langkah dari aktivitas harus berada pada urutan yang benar.
6. Lingkup dan range dari aktifitas yang sedang digambarkan harus ditelusuri dengan hati-hati. Percabangan-percabangan yang memotong aktivitas yang sedang digambarkan tidak perlu digambarkan pada flowchart yang sama. Simbol konektor harus digunakan dan percabangannya diletakan pada halaman yang terpisah atau hilangkan seluruhnya bila percabangannya tidak berkaitan dengan sistem.

7. Gunakan simbol-simbol flowchart yang standar

Perlu diperhatikan juga simbol-simbol yang ada di dalam *flowchart* beserta fungsinya, sebagai berikut :

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

C. Algoritma Dalam Bentuk Kalimat *pseudocode*

Pseudocode adalah algoritma yang bentuknya (strukturnya) sangat mirip dengan bahasa pemrograman khususnya bahasa pemrograman terstruktur seperti *pascal*. Kemiripan ini merupakan keuntungan dari *pseudocode* karena implementasi atau penerjemahan algoritma ke dalam *source code* suatu bahasa pemrograman sangatlah mudah meskipun penggunaannya tidak sepopuler *flowchart*.

Dalam penulisannya, *pseudocode* harus terdiri dari tiga bagian, yaitu:

1. Judul algoritma
Bagian yang terdiri atas nama algoritma dan penjelasan (spesifikasi) dari algoritma tersebut. Nama sebaiknya singkat dan menggambarkan apa yang dapat dilakukan oleh algoritma tersebut.
2. Deklarasi
Bagian untuk mendefinisikan semua nama yang digunakan di dalam program. Nama tersebut dapat berupa nama tetapan, peubah atau variabel, tipe, prosedur, dan fungsi
3. Deskripsi
Bagian ini berisi uraian langkah-langkah penyelesaian masalah yang ditulis dengan menggunakan aturan-aturan yang akan dijelaskan selanjutnya (Suprpto dkk, 2008:60-61).

2.3 Penerapan Algoritma dan *Flowchart* dalam menyelesaikan masalah

A. Algoritma

Dalam kehidupan sehari-hari, terdapat banyak proses yang dinyatakan dalam suatu algoritma.

Jika kita buat algoritma memasak mie akan seperti di bawah ini:

1. Siapkan 1 bungkus mie instan, 400 ml air (2 gelas), panci, mangkok, sendok, dan garpu
2. Masukkan 400 ml air kedalam panci
3. Masak air
4. Tunggu hingga mendidih
5. Masukkan mie kedalam panci yang sudah berisi air mendidih
6. Tunggu dan aduk hingga 3 menit
7. Jika sudah matang masukkan bumbu
8. Aduk hingga rata
9. Sajikan mie

Algoritma mencuci piring :

1. Siapkan spons
2. Apakah spons ada?
3. Jika tidak ada, beli diwarung
4. Siapkan sabun pencuci piring

5. Apakah sabun ada?
6. Jika tidak ada, beli diwarung
7. Taruh piring kotor kedalam wastafel
8. Buka keran air
9. Basahi piring kotor dengan air
10. Tutup keran air
11. Tuang sabun ke spons
12. Usapkan spons ke piring yang kotor
13. Buka kran air
14. Bilas piring yang telah disabun dengan air sampai bersih
15. Tutup keran air
16. Jika sudah bersih, taruh disamping wastafel
17. Tunggu hingga airnya tiris dan kering
18. Jika sudah kering, pindahkan piring kedalam rak piring
19. Piring siap digunakan

Algoritma mengganti ban mobil :

1. Lihat ban bagian mana yang bocor
2. Siapkan dongkrak dan peralatan
3. Apakah dongkrak dan peralatannya ada?
4. Jika tidak ada, pinjam dongkrak
5. Pasang dongkrak
6. Lepaskan ban yang pecah
7. Siapkan ban baru
8. Apakah ban baru ada?
9. Jika tidak ada, cari tukang tambal ban
10. Berikan ban yang pecah until ditambal
11. Tunggu ban selesai ditambal
12. Kembali ke mobil
13. Pasang ban yang baru
14. Kencangkan baut
15. Lepaskan dongkrak
16. Masukkan dongkrak dan peralatan lainnya kedalam bagasi mobil
17. Masuk ke mobil
18. Mobil bisa digunakan seperti semula

Dan masih banyak lagi algoritma-algoritma dalam kehidupan sehari-hari. Dalam dunia IT seperti ini juga sangat dibutuhkan untuk menyelesaikan suatu masalah yang sedang terjadi dengan mengerti urutan langkah-langkah logis yang disusun secara sistematis dan logis yang biasa disebut dengan algoritma (Setiawan, 2018)

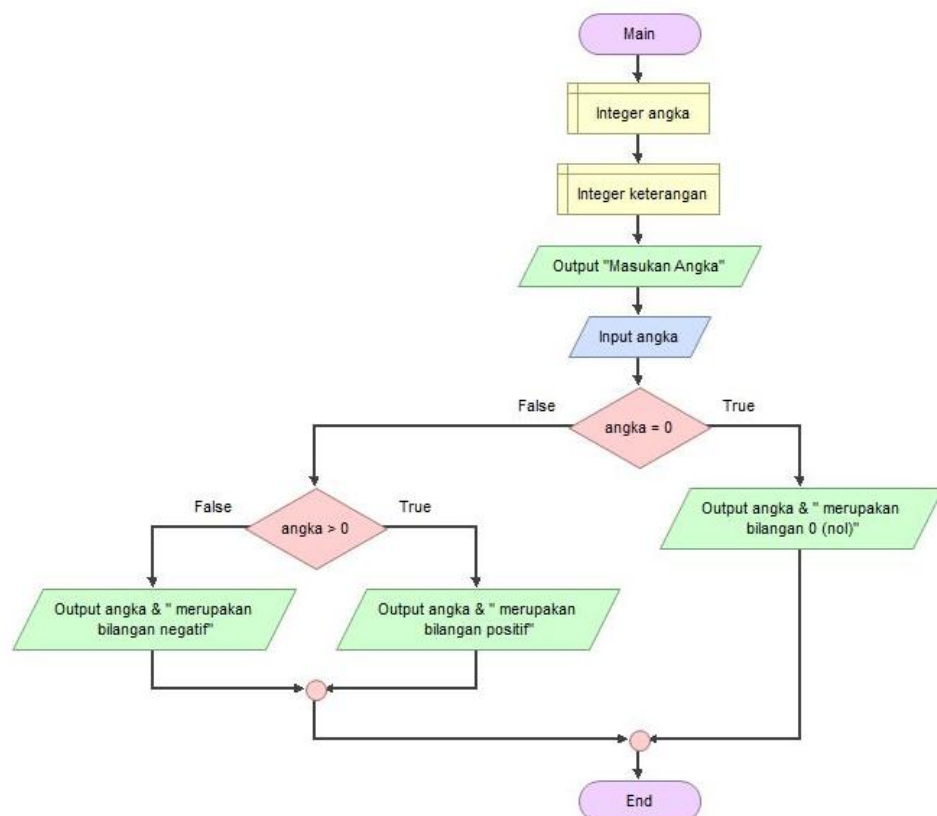
B. Flowchart

Sebelum membuat *flowchart* ada petunjuk yang harus diperhatikan dan di atas sudah di jelaskan oleh Fadilah di dalam makalah *Flowchart* (2020:3).

Algoritma sendiri mempunyai 3 (tiga) notasi algoritma yaitu deskripsi, *flowchart*, dan *pseudocode*. *Flowchart* bisa disebut dengan alat bantu untuk membuat sebuah algoritma.

Contoh notasi algoritma dalam bentuk *flowchart* :

Menentukan sebuah bilangan apakah menjadi bilangan positif atau bilangan negatif



Notasinya dalam bentuk deskriptif :

Algoritma menentukan sebuah bilangan apakah menjadi bilangan positif atau bilangan negatif.
{Menentukan sebuah bilangan tersebut menjadi bilangan positif atau bilangan negatif. Dengan menggunakan program percabangan if... then...else}

Deklarasi :

angka : integer;

keterangan : integer;

Deskripsi :

1. Masukan dan inputnya
 output "Masukan angka"
 input : angka
2. Mulai program percabangan.
 if angka = 0 then
 output angka & " merupakan bilangan 0 (nol)"
 else
 if angka > 0 then
 output angka & " merupakan bilangan positif"
 else
 output angka & " merupakan bilangan negatif"
3. Tampilan hasil pada layar
4. Selesai

Notasinya dalam bentuk *pseudocode* :

Function Main

 Declare integer angka

 Declare integer keterangan

 output "Masukan angka"

 input angka

 if angka = 0 then

 output angka & " merupakan bilangan 0 (nol)"

 else

 if angka > 0 then

 output angka & " merupakan bilangan positif"

 else

 output angka & " merupakan bilangan negatif"

 endif

 endif

end

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan teori yang telah dijabarkan, serta melihat fakta yang ada, maka dapat ditarik sebuah kesimpulan. Diantaranya:

1. Algoritma adalah sebuah urutan cara penyelesaian sebuah masalah dan juga pengambilan keputusan yang dapat dimanfaatkan untuk menjalankan suatu program.
2. Algoritma tidak dapat di buat secara tidak pasti, karena dibutuhkan algoritma yang baik dan sistematis untuk menghasilkan output yang tepat.
3. *Flowchart* didefinisikan juga sebagai bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah.
4. *Flowchart* merupakan gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya.
5. Tujuan utama *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas dengan menggunakan simbol-simbol yang standar.

Jadi hubungan antara algoritma dengan *flowchart* yaitu Flowchart sebagai gambar dari penjelasan atau diskripsi dari algoritma

3.2 Saran

Berdasarkan pembahasan diatas dapat diketahui bahwa pemahaman tentang algoritma maupun *flowchart* sangat penting. Oleh karena itu,beberapa hal yang dapat dilakukan:

1. Memahami dengan baik tentang cara pembuatan algoritma dan *flowchart*
2. Mempelajari pengaplikasian algoritma dan *flowchart* dengan sebaik-baiknya dengan cara membaca artikel, buku dan lain-lain
3. Mempraktekannya ke dalam sebuah bahasa pemrograman dengan masalah-masalah lain, misalnya membuat aplikasi, membuat *website*, dan lain-lain

DAFTAR PUSTAKA

Suprpto, Adi, dkk. 2008. *Bahasa Pemrograman*. Jakarta : Direktorat Pembinaan SMK

Tominanto dan Subinarto. 2018. *Algoritma dan Pemrograman*. Jakarta : Kementrian Kesehatan Republik Indonesia

Fadilah, Moh Fikri. 2020. "*Flowchart*". Makalah

Karma, Indrawan. 2015. "Algoritma dan Pemrograman". Makalah

Setiawan, Hendri Sheza. 2018. "Algoritma Dalam Kehidupan Kita".
<https://hendrisheza.blogspot.com/2018/02/contoh-algoritma-dalam-kehidupan-kita.html/>. Diakses pada 4 November 2020.