Artikel maulidya pmm1

by Maulidya Maulidya

Submission date: 28-Dec-2021 08:03AM (UTC+0700)

Submission ID: 1735990914

File name: Artikel_-_Maulidya_Asrianti.pdf (501.99K)

Word count: 2999

Character count: 18930

Konsep Dasar Algoritma Pemrograman Sebagai Pengantar Pembelajaran Pada Mata Kuliah Algoritma dan Pemrograman

Oleh: Maulidya Asrianti

PMM FITK UIN SU

Email: maulidyaasrianti27@gmail.com

Abstrak

Algoritma merupakan serangkaian instruksi ¹⁰ untuk menyelesaikan masalah. Algoritma menyelesaikan masalah secara sistematis dan logis. Dalam bidang komputer, algoritma sangat diperlukan dalam menyelesaikan berbagai masalah pemrograman. Apabila algoritma tidak dirancang sedemikian rupa, maka proses pemrograman akan menjadi rusak, salah, tidak efisien, dan berakhir lambat. Oleh karenanya, penyelesaian algoritma pada komputer diperlukan bahasa khusus. Algoritma sejatinya tersusun atas 3 struktur dasar, yakni pengulangan, percabangan, dan terurut. Artikel ini bertujuan untuk memberikan pengetahuan dasar sebagai bekal pembelajaran pada materi algoritma dan pemrograman, memudahkan mahasiswa memahami dasar-dasar algoritma pemrograman yang berisikan tentang konsep algoritma, pemrograman, struktur algoritma, dan penyajian algoritma.

Kata kunci: Algoritma, Pemrograman, Bahasa Pemrograman, Struktur Algoritma, Penyajian Algoritma

Pendahuluan

Algoritma bukan merupakan hal yang baru. Algoritma bahkan sudah ¹⁷ ada sejak zaman sebelum masehi. Asal kata algoritma sendiri memiliki sejarah yang unik. Orang-orang menemukan dan mengartikan kata algorism sebagai cara menghitung menggunakan bahasa Arab.

Ahli bahasa mencoba mencari asal kata algorism ini, hanya saja hasil yang didapat tidak memuaskan. Kemudian, ditemukanlah hasil kata ⁶ tersebut oleh para ahli. Algoritma diambil dari nama seorang penulis buku asal Arab, yang bernama Abu Ja'far Muhammad ibnu Musa al-Khuwarizmi.

Peralihan kata algorism menjadi algorithm terjadi karena kata algorism yang sering kali disalahartikan dengan arithmetic,

sehingga akhiran -sm berubah menjadi -thm. Lambat laun kata algorithm diartikan sebagai metode perhitungan (komputasi).

Kemudian kata algorithm diterima ke dalam bahasa Indonesia menjadi algoritma. Matematikawan dari Yunani bernama Euclid (350 M) berpesan di dalam bukunya mengenai dua hal penting. Pertama, algoritma tidak boleh salah. Kedua, algoritma harus berhenti, lalu menghasilkan akhir yang benar.

²⁶ Algoritma adalah rangkaian langkah-langkah pemecahan masalah yang masuk akal yang terstruktur secara berurutan dan independen dari bahasa pemrograman tertentu. Dalam algoritma, kata logis adalah kuncinya. Pada algoritma langkah-langkahnya harus logis yang dalam hal ini artinya adalah masuk akal, mengikuti suatu

urutan, dan tidak boleh menjengkal-jengkali. Algoritma juga harus dapat diklasifikasikan sebagai benar atau salah.

Algoritma merupakan serangkaian instruksi dalam menyelesaikan masalah. Masalahnya bisa apa saja, asalkan setiap masalah memiliki kondisi awal sebelum suatu algoritma dijalankan. Serangkaian instruksi tersebut dikenal dengan nama program.

Untuk menjalankan program tersebut, komputer memerlukan sebuah bahasa yang bisa dipahami. Bahasa pemrograman adalah yang bisa dimengerti oleh komputer. Untuk menuliskan sebuah algoritma, setiap langkahnya harus jelas maknanya dan tidak terikat dengan aturan tertentu (Edy Budiman, 2015).

Permasalahan dalam pemrograman, dapat diselesaikan dengan algoritma. Algoritma sangat diperlukan di komputer. Apabila algoritma tidak dirancang sedemikian rupa, maka proses pemrograman akan menjadi rusak, salah, tidak efisien, dan berakhir lambat. Keahlian berpikir seperti menganalisis, berpikir secara logis, mendesain, merefleksi, dan mengevaluasi sangat diperlukan di bidang pemrograman. (Hadjerrouit, 2007). Dalam ilmu komputer pemrograman adalah keterampilan dasar yang harus dikuasai oleh mahasiswa.

Bagi seorang mahasiswa yang menempuh pendidikan di bidang informatika dan studi komputer, algoritma dan pemrograman adalah mata pelajaran dasar yang sangat penting. Penguasaan yang baik dari konsep algoritma dan pemrograman

mutlak diperlukan. Hal ini sangat membantu demi kelancaran studi mahasiswa.

Studi Literatur

Secara umum algoritma merupakan cara, urutan, atau tahapan tindakan yang diatur dengan sistematis untuk menyelesaikan suatu kegiatan. Sehingga proses yang akan diselesaikan dapat sesuai dengan algoritma atau urutan langkah yang telah disusun (Wahyu Eko Susanto dan Akhmad Svukron, 2020). Dalam algoritma, kata logis adalah kuncinya. Langkah-langkah logis (masuk akal, tidak boleh melompat-lompat, tersusun secara berurutan) serta harus dapat ditentukan bernilai benar atau salah sangat diperlukan dalam algoritma. Algoritma memiliki arti solusi. Ketika pembicaraan orang-orang seputar algoritma ada bidang pemrograman, jalan keluar dalam menyelesaikan masalah dengan komputer adalah yang dimaksud (Edy Budiman, 2015).

Menurut Gun Gun Maulana (2017), algoritma adalah serangkaian perintah berupa langkah untuk memecahkan suatu masalah. Perintah dapat diterjemahkan secara bertahap mulai dari awal hingga akhir, sedangkan masalah dapat berupa apa saja, asalkan ketika belum menjalankan suatu algoritma harus ada kondisi awal yang dipenuhi.

Menurut Donald E. Knuth dalam judul bukunya "The Art of Computer Programming" merumuskan 5 hal utama yang wajib ada dalam algoritma:

1. Finiteness, setelah melakukan serangkaian langkah, algoritma diharuskan berhenti.

2. Definiteness, tidak boleh ambigu dalam tiap langkahnya.
3. Input, setiap algoritma harus ada masukan.
4. Output, setiap algoritma harus ada keluaran.
5. Effectiveness, keefektifan harus ada pada algoritma.

Sehari-hari banyak kita temukan langkah-langkah suatu pengerjaan walaupun tidak kita sebut sebagai algoritma. Contoh mudahnya adalah tahapan-tahapan dalam memasak air, dimulai dari mengambil panci, mengisi panci dengan air, menyiapkan kompor, meletakkan panci berisi air ke atas kompor, menghidupkan kompor. Hampir setiap hari kita melakukan aktifitas ini, dan telah kita lakukan yang disebut dengan algoritma secara tidak sadar.

Contoh algoritma: mengganti isi dari dalam dua buah cangkir, cairan biru di cangkir A, dan cairan merah di cangkir. Maka, penyelesaian masalah tersebut dapat dilakukan dengan sebuah cara, misal:

1
Algoritma 1:

- a. Tuang isi cangkir A ke cangkir B
- b. Tuang isi cangkir B ke cangkir A

Algoritma 2:

- a. Tuang isi cangkir A ke cangkir C
- b. Tuang isi cangkir B ke cangkir A
- c. Tuang isi cangkir C ke cangkir B

Lalu bisa kita analisis kedua jawaban di atas: Algoritma 1 tidak salah, hanya saja pertukaran yang dihasilkan tidak benar serta penyelesaiannya tidak tepat. Apabila

dilakukan dengan cara ini akan terjadi kemungkinan cairannya tercampur.

1
Algoritma 2 sudah tepat. Dengan memakai bantuan sebuah cangkir C, maka kedua isi gelas dapat ditukar tanpa tercampur.

Algoritma umumnya terstruktur atas 3 dasar, yaitu perulangan, percabangan, dan terurut.

1. Perulangan (*looping*)

Di keseharian, kita banyak melakukan pekerjaan yang berulang seperti makan, minum, mencuci dan lain sebagainya. Pekerjaan tersebut akan dilakukan sampai sampai semua pekerjaan selesai, kemudian berhenti. Misalnya mencuci semua baju kotor sampai bersih, maka ketika semua baju telah bersih proses pun akan terhenti.

2. Percabangan

Percabangan adalah kondisi ketika harus menentukan suatu pilihan. Pada algoritma hal seperti ini disebabkan ketika ada proses yang tidak pada tempatnya, sehingga diminta untuk menentukan proses apa yang akan dilakukan selanjutnya apabila keadaan tersebut terjadi.

3. Terurut (*sequential*)

Terurut atau sequential adalah proses di mana tersusun algoritma berdasarkan urutannya. Algoritma tersebut tidak berulang ataupun bercabang, sehingga berjalannya proses akan urut mulai dari proses ke 1 sampai proses ke -n.

Dalam pembuatan algoritma yang baik, kita harus paham bagaimana prosesnya dan mengidentifikasi segala kemungkinan yang dapat terjadi. Apabila dijalankan sebuah

pemroses (processor) maka akan efektif algoritmanya. Yang merupakan pemroses adalah manusia, komputer, robot, mesin, dan sebagainya. Tiap instruksi akan dibaca oleh pemroses kemudian dikerjakan di dalam algoritma (Dewi Yunita Sari, 2020).

Dalam hal ini fokus kita ada pada komputer. Perintah-perintah dalam algoritma yang telah dimasukkan akan dijalankan oleh komputer sebagai alat bantu. Perintah yang dimasukkan ke dalam algoritma harus ditulis dengan bahasa komputer. Dengan demikian, harus ditulis dengan bahasa khusus yakni bahasa komputer. Algoritma yang ditulis dalam bahasa komputer dinamakan program. Bahasa pemrograman adalah bahasa yang digunakan ketika menulis program, programmer adalah orang yang menciptakan program, dan pemrograman adalah kegiatan merancang dan menulis program. (Patmi Kasih dan Umi Mahdiah, 2018).

Fungsi dari diciptakannya bahasa pemrograman adalah agar kita tidak sulit berkomunikasi dengan komputer. Dengan hadirnya bahasa pemrograman ini, ketika manusia ingin berkomunikasi dengan komputer tidak perlu lagi diterjemahkan ke dalam 0 dan 1. Jika manusia melakukan ini, seberapa rumit sebuah program? (Edy Budiman, 2015). Menurut Edy Budiman umumnya bahasa pemrograman terbagi atas empat kelompok:

1. Bahasa Tingkat Rendah (Low Level Language). Bahasa ini yang mengarah pada mesin. Bahasa ini memiliki ciri harus berpikir sesuai dengan logika berpikir otomatis, sehingga kurang

fleksibel dan sulit dipahami. Contoh: bahasa mesin, bahasa assembly (rakitan)

2. Bahasa Tingkat Menengah (Middle Level Language). Saat ingin menuliskan ekspresi atau pernyataan standar yang mudah dipahami manusia dan adanya instruksi khusus yang bisa diakses oleh komputer bahasa inilah yang digunakan. Bahasa yang memiliki aturan. Contoh: Bahasa C

3. Bahasa Tingkat Tinggi (High Level Language). Saat ingin menuliskan ekspresi atau pernyataan standar yang langsung dipahami manusia, dapat menggunakan bahasa ini. Adanya aturan gramatikal dalam penggunaannya. Contoh: Bahasa Pascal, Basic, COBOL

4. Bahasa Berorientasi Objek (Object Oriented Programming). Dengan hanya memasukkan ciri-ciri yang kita maksud, dan kita tidak lagi menuliskan pernyataan maupun ekspresi secara detail. Seperti itulah bahasa ini. Contoh: Delphi, Visual Basic, C++.

Bahasa pemrograman dan algoritma sangat erat kaitannya dengan program. Ketika program menjadi kurang baik, maka algoritma yang dibuat sebelumnya dalam kondisi baik namun bahasa yang digunakan tidak tepat, demikian pula sebaliknya. Algoritma yang dibuat memiliki banyak kelebihan, yaitu: (1) Tidak bergantungnya penulisan algoritma pada bahasa pemrograman apapun, (2) Berbagai bahasa pemrograman yang manapun itu dapat menerjemahkan notasi algoritma, (3) Bahasa pemrograman apa pun yang digunakan, outputnya akan menjadi sama dikarenakan algoritma yang sama pula, (Andika Fisma, 2015).

Kualitatif adalah metode yang dipakai untuk penelitian ini, dengan pengumpulan data adalah studi pustaka. Penulis menghimpun data-data dari banyak sumber seperti buku, jurnal, maupun portal media yang tentunya bisa dijadikan sebagai bahan rujukan yang berkaitan dengan tema yang diangkat. Proses penelitian dilakukan dengan menghimpun data-data yang didapat dari studi literatur. Setelah semua data didapat, penulis akan melakukan kompilasi, kemudian data tersebut dianalisis, dan terakhir disimpulkan.

Pembahasan

Penggambaran algoritma secara umum ditampilkan dengan dua cara, yaitu dengan teks/tulisan dan dengan gambar. Penyajian algoritma dengan tulisan dapat disajikan dengan bahasa tertentu misalnya bahasa Inggris atau bahasa Indonesia maupun dengan *pseudocode*. Sedangkan penyajian dengan gambar dapat disajikan dengan *flowchart*, namun sebenarnya *flowchart* bukan merupakan satu-satunya cara dalam penggambaran algoritma. Penggambaran algoritma dapat juga menggunakan data *flow diagram*, HIPO (Hierarcial Input Process Output) maupun dengan menggunakan *structure chart*. Namun, umumnya dalam menulis algoritma, ada tiga macam cara yang biasa digunakan, yakni:

1. Kalimat deskriptif
2. Pseudocode
3. Flowchart

1. Kalimat deskriptif

Penulisan instruksi yang jelas dalam algoritma adalah dengan kalimat deskriptif.

Notasi algoritma juga harus beriringan dengan bahasa pemrograman yang umum, agar notasinya dapat dengan mudah diterjemahkan ke dalam bahasa pemrograman. Pada umumnya notasi bahasa kalimat deskriptif ini berbahasa Inggris, namun bisa menggunakan bahasa Indonesia juga, atau bahasa yang digunakan sehari-hari.

Penulisan bahasa deskriptif biasanya menggunakan kata kerja, contohnya hitung, tulis, baca, tampilkan, ulangi, bandingkan, dan lain-lain. Algoritma yang pendek sangat cocok menggunakan notasi ini. Begitupun sebaliknya, notasi ini kurang efektif bila digunakan pada algoritma yang panjang. Kalimat deskriptif menyusun algoritma dengan tiga bagian utama, yaitu judul (header), deklarasi (kamus), dan deskripsi.

a. Judul (header)

Pada bagian ini berisikan nama algoritma dan penjelasan (spesifikasi). Spesifikasi algoritma dituliskan dalam kurung kurawal ({}). Berikut contohnya:

```
Algoritma luas_layang-layang <-
judul
{Menghitung luas layang-layang
dengan ukuran sisi tertentu.
Algoritma menerima masukan sisi
layang-layang, menghitung luas
layang-layang, kemudian cetak
luasnya pada piranti keluaran} <-
spesifikasi
```

b. Deklarasi (kamus)

Pada bagian ini nama pada algoritma didefinisikan. Nama tersebut dapat berupa nama variabel, nama tipe, nama konstanta,

nama prosedur atau nama fungsi. Berikut contohnya:

Deklarasi:

```
diagonal = real (tipe data
bilangan pecahan)

Luas = real (tipe data bilangan
pecahan)
```

c. Deskripsi

Bagian ini menjelaskan langkah pemecahan masalah. Berikut contohnya:

Deskripsi:

1. Baca diagonal
2. Hitung luas
3. Tampilkan luas ke layar
4. Selesai

Algoritma luas_layang-layang <-
judul

{Menghitung luas layang-layang dengan ukuran sisi tertentu. Algoritma menerima masukan sisi layang-layang, menghitung luas layang-layang, kemudian cetak luasnya pada piranti keluaran} <- spesifikasi

Deklarasi:

```
diagonal = real (tipe data
bilangan pecahan)

Luas = real (tipe data bilangan
pecahan)
```

Deskripsi:

1. Baca diagonal
2. Hitung luas
3. Tampilkan luas ke layar
4. Selesai

2. Pseudocode

Salah satu solusi penyelesaian masalah dalam algoritma adalah dengan meniru bahasa pemrograman itu sendiri, atau lebih dikenal dengan nama pseudocode. Pseudocode merupakan langkah penulisan algoritma yang hampir serupa dengan bahasa pemrograman. Penggunaan bahasa dalam pseudocode biasanya bersifat universal dan mudah untuk dipahami. Penggunaan bahasanya dirancang lebih ringkas dibanding algoritma. Karena tidak jauh beda dengan bahasa pemrograman, maka penggunaan kata kunci pseudocode sebaiknya menggunakan bahasa pemrograman yang umum, seperti: while, else, then, if, do, for, repeat, dan sebagainya. Berikut contoh pseudocode menghitung luas dan keliling persegi panjang:

```
(1) Start
(2) Baca_panjang (p)
(3) Baca_lebar
(4) Luas = p x l
(5) Keliling = 2p + 2l
(6) cout ("luas persegi panjang: ") <- luas
(7) cout ("keliling persegi panjang: ") <- keliling
(8) End
```

Berikut contoh penentuan ¹bilangan ganjil atau genap pada pseudocode:

```

Read bil
if (bil mod 4 = 0) then
    cout ("Output Genap")
else
    cout ("Output Ganjil" ) End

```

3. Flowchart

Penulisan algoritma yang terakhir adalah flowchat. Penulisan algoritma cara ini adalah dengan penggunaan simbol-simbol. Simbol-simbol tersebut direpresentasikan sebagai gambaran tahap-tahap penyelesaian masalah. Adapun simbol-simbolnya adalah:

Simbol Terminal, menunjukkan awalan atau akhiran sebuah program.
Simbol Input/Output, menunjukkan operasi masukan atau keluaran.
Simbol Proses, menunjukkan untuk proses pengolahan data.
Simbol Keputusan/Decision, untuk memilih kondisi tertentu.

Simbol persiapan/Preparation, memberikan nilai awal pada suatu variabel atau pencacah.
Simbol proses terdefinisi/predefined process, untuk menunjukkan terpisahnya detail yang diproses, contohnya dalam bentuk Subroutine.
Simbol Penghubung menuju halaman berbeda, menghubungkan dengan halaman berbeda pada bagian diagram alir.
Simbol Penghubung ke halaman yang sama, menghubungkan dengan halaman sama pada bagian diagram alir.
Simbol Arah aliran, simbol menandakan arah pada aliran proses.

Berikut contoh penggunaannya dalam proses perhitungan persegi panjang:

Algoritma hakikatnya adalah langkah-langkah menyelesaikan masalah, maka dalam penyelesaiannya haruslah sesuai dengan struktur dasar algoritma itu sendiri:

1. Perulangan (*looping*)

Struktur ini memiliki beberapa bentuk:

a. Struktur while, selagi syarat tersebut masih terpenuhi.

```

1) Start
2) i = 0
3) selagi i <= 8 lakukan langkah 4 sampai langkah 5
4) cetak i
5) i = i + 1
6) selesai
  
```

b. Struktur do ... while ..., hingga syarat tidak terpenuhi.

```

3) Start
2) i = 0
3) i = i + 1
4) cetak i
5) jika i < 8, lakukan langkah 3 sampai langkah 4
6) selesai
  
```

c. Struktur for, sampai jumlah perulangan syarat dipenuhi.

```

1) Start
2) Kerjakan langkah 3 mulai i = 1 sampai i = 8
3) Cetak i
4) Selesai
  
```

d. **Struktur** for bersarang, dengan mengimplementasikan for dalam for.

```

1) Start
2) Lakukan langkah 3 mulai i = 1 sampai i = 8
3) Jika i <= 8 lakukan langkah 4
4) Lakukan langkah 5 mulai dari j=1 sampai j=8
5) Jika j <= 8 lakukan langkah 6
6) i*j
7) cetak (i+j)
8) j = j + 1
9) kembali ke langkah 5, hingga j>8 (langkah 5 bernilai salah) kerjakan langkah 10
10) i=i+1
11) kembali ke langkah 3 sampai i>8 (langkah 3 bernilai salah)
12) Selesai
  
```

2. Percabangan/pemilihan (*selection*)

Pada struktur ini hanya yang memenuhi syarat sajalah yang dikerjakan. Struktur ini dapat dilakukan dengan dua cara:

a. Struktur if

1. If sederhana

```
1) Start
2) Baca usia
3) if (usia>10), maka lakukan
   langkah 4 selain itu selesai
4) Cetak "selamat menonton"
5) selesai
```

2. If ... then ... else

Operasi ini berdasarkan kode operasi. Jumlahkan dua bilangan apabila nilai kode adalah 1, dan apabila nilai kode bukan 1 maka kalikan dua bilangan.

```
1) Start
2) Baca bilangan C
3) Baca bilangan D
4) Baca kode operasi
5) Jika (kode=1), kerjakan
   langkah 6. Jika (kode=2/selain
   1), lakukan langkah 7
6) Hasil = C + D
7) Hasil = C * D
8) Cetak hasil
9) selesai
```

3. Nested if

1 Nested if mempunyai susunan flowchart sebagai berikut:

3. Terurut (*sequential*)

Setiap langkah program dikerjakan berurut seperti urutan penulisannya dimulai dari atas hingga bawah.

- 1) Mulai membaca dan memahami bentuk struktur
- 2) Baca sisi
- 3) Hitung luas = sisi * sisi
- 4) Tampilkan luas
- 5) selesai

Kesimpulan

Algoritma merupakan serangkaian instruksi atau langkah-langkah dalam menyelesaikan masalah. Masalahnya bisa apa saja, asalkan setiap masalah memiliki kondisi awal sebelum suatu algoritma dijalankan. Algoritma menyelesaikan permasalahan secara sistematis dan logis. Tanpa kita sadari, di dalam keseharian kita telah bertemu dengan langkah-langkah pengerjaan sesuatu walaupun kita tidak menyebutnya dengan algoritma. Umumnya algoritma terdiri atas 3 struktur dasar yakni terurut, percabangan, dan perulangan. Agar algoritma dapat berjalan dengan baik, maka kita harus memahami bagaimana prosesnya dan mengidentifikasi segala kemungkinan yang dapat terjadi. Salah satu solusi penyelesaian masalah dalam algoritma

adalah dengan meniru bahasa pemrograman itu sendiri

Dalam hal ini fokus kita ada pada komputer. Perintah-perintah dalam algoritma yang telah dimasukkan akan dijalankan oleh komputer sebagai alat bantu. Agar komputer dapat memahami perintah yang dimasukkan, maka penulisan perintah tersebut haruslah ditulis menggunakan bahasa yang dapat komputer pahami. Algoritma yang ditulis dalam bahasa komputer dinamakan program. Umumnya dalam menulis algoritma, ada tiga macam cara yang biasa digunakan, yakni: Kalimat deskriptif, Pseudocode, dan Flowchart. Dan umumnya Algoritma umumnya terstruktur atas 3 dasar, yaitu perulangan, percabangan, dan terurut.

Daftar Bacaan

- Ardiansyah, H., Resti, Agus B. P. (2019). Algoritma dan Pemrograman 1. Tangerang Selatan: Unpam Press.
- Barakbah, Ali R., Tita K.a, Ahmad S. A. (2014). Logika dan Algoritma. Surabaya.
- Bibi, S. (2015). Efektivitas Model Blended Learning Terhadap Motivasi dan Tingkat Pemahaman Mahasiswa Mata Kuliah

- Algoritma dan Pemrograman. Jurnal Pendidikan Vokasi, Vol 5, Nomor 1, Februari 2015.
- Budiman, E. (2015). Belajar Dasar Algoritma dan Pemrograman. Samarinda: Pemula.
- Budiyanto, A. (2003). Pengantar Algoritma dan Pemrograman. Available: <https://www.academia.edu/>
- Chaiunnas, A. (2017). Penerapan Algoritma Tripod Gait Pada Robot Hexapod Menggunakan Arduino Mega128. Jurnal Penelitian Pos Indonesia Vol 7 No 1, September 2017.
- Dianta, I. A. (2019). Logika dan Algoritma Pemrograman.
- Gunadi, G., Dana I. S. (2012). Penerapan Metode Data Mining Market Basket Analysis Terhadap Data Penjualan Produk Buku Dengan Menggunakan Algoritma Apriori dan Frequent Pattern Growth (FP-Growth): Studi Kasus Percetakan PT Gramedia. Jurnal TELEMATIKA MKOM Vol.4 No.1, Maret 2012.
- ¹⁵ Hadjerrouit, S. (2007). A Blended learning model in java programming: A designbased research approach. Proceeding of the 2007 computer science and IT education conference, 283-308.
- Kasih, Patmi, dan Umi M. (2018). Konsep Dasar Algoritma Pemrograman Menggunakan C++. Kediri: CV Kasih Inovasi Tinggi.
- Layona, Rita, Budi Y., dan Yovita T. (2017). *Authoring Tool for Interactive Video Content for Learning Programming. Scientific Committee of the 2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, 13-14 October 2017*
- ² Maulana, G. (2017). Pembelajaran Dasar Algoritma dan Pemrograman Menggunakan El-Goritma Berbasis Web. Jurnal Teknik Mesin (JTM): Vol. 06, Edisi Spesial 2017.
- ⁸
- ¹⁹

- Nita, S., Sulistyning K. 2020. Analysis Of The Impact Narrative Algorithm Method, Pseudocode And Flowchart Towards Students Understanding Of The Programming Algorithm Courses. IOP Conf. Series: Materials Science and Engineering.
- Sari, Dewi Y. (2020) Algoritma dan Pemrograman Pascal dan C++. Indramayu: K-Media.
- Susanto, Wahyu E., dan Akhmad S. (2020). Logika & Algoritma untuk Pemula. Yogyakarta: Graha Ilmu.
- Yahfizham, ST., M.Cs. *Analisis Waktu Algoritma Quicksort dan Mergesort.*

Artikel maulidya pmm1

ORIGINALITY REPORT

24%

SIMILARITY INDEX

24%

INTERNET SOURCES

5%

PUBLICATIONS

7%

STUDENT PAPERS

PRIMARY SOURCES

1

repository.unpkediri.ac.id

Internet Source

8%

2

docplayer.info

Internet Source

2%

3

eprints.uny.ac.id

Internet Source

2%

4

www.slideshare.net

Internet Source

1%

5

ft.unwir.ac.id

Internet Source

1%

6

www.coursehero.com

Internet Source

1%

7

acidz-ti-poltektegal.blogspot.com

Internet Source

1%

8

eprints.umg.ac.id

Internet Source

1%

9

nyohongo.wordpress.com

Internet Source

1%

10	journal.uny.ac.id Internet Source	1 %
11	123dok.com Internet Source	1 %
12	sukeratayasa.wordpress.com Internet Source	1 %
13	Submitted to Griffth University Student Paper	<1 %
14	archive.org Internet Source	<1 %
15	www.yumpu.com Internet Source	<1 %
16	andichristiandotblog.files.wordpress.com Internet Source	<1 %
17	ilmukomputerstikom.wordpress.com Internet Source	<1 %
18	www.quantumbook.id Internet Source	<1 %
19	jurnal-ppi.kominfo.go.id Internet Source	<1 %
20	makalah18.blogspot.com Internet Source	<1 %
21	Sunarko Sunarko Sunarko, Hanafi Ali. "PERANCANGAN ALGORITMA PADA	<1 %

PENGEMBANGAN OTOMATISASI SISTEM RABBIT PNEUMATIK DI REAKTOR RSG-GAS", Reaktor : Buletin Pengelolaan Reaktor Nuklir, 2020

Publication

22

es.scribd.com

Internet Source

<1 %

23

repo.apmd.ac.id

Internet Source

<1 %

24

belajarlprog.blogspot.com

Internet Source

<1 %

25

ilmukuring.wordpress.com

Internet Source

<1 %

26

www.ngopot.com

Internet Source

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On