PENGANTAR ALGORITMA DAN PEMROGRAMAN

Oleh: Intana Purnama Sari
PMM FITK UINSU
Intanapurnamasari@gmail.com

ABSTRAK

“Algoritma” merupakan sebuah kata yang dikutip dari nama seorang ilmuwan dari Persia, yaitu Abu Ja'fzr Mohammad bin Musa al-Khawarizmi. Masyarakat Barat menyebut namanya Al-khwarizmi sebagai algorism dan kemudian dimasukkan ke dalam algoritma dalam bahasa Indonesia. Mengenai bahasa “pemrograman” yang dibuat oleh Korad zuse. Sejak ia menciptakan kalkulator papan bahasa pemrograman komputer pertama, itu telah dianggap sebagai alat untuk komputer virtual (digital). Algoritma pemrograman adalah langkah-langkah atau urutan untuk memecahkan masalah dalam pemrograman komputer.

KATA KUNCI : *Algoritma pemrograman, flowchart, pseudocode, komputer, matematika*

I. PENDAHULUAN

Algoritma merupakan metode efisien yang dinyatakan menjadi deret sampai. Algoritma pula ialah deretan perintah buat memecahkan suatu persoalan. Tugas-tugas ini bisa diterjemahkan langkah demi langkah dari awal sampai akhir. Masalahnya bisa apa saja, asalkan untuk setiap persoalan terdapat kriteria syarat awal yang harus dipenuhi sebelum suatu algoritma dapat dieksekusi. Algoritma juga mempunyai proses perulangan serta jua mempunyai keputusan hingga keputusan tadi terselesaikan.

Pada cabang bidang ini, algoritma dipelajari secara abstrak, terlepas dari sistem komputer maupun bahasa pemrograman yang digunakan. Algoritma yang berbeda dapat diterapkan untuk masalah dengan kriteria yang sama. Kompleksitas suatu algoritma adalah ukuran jumlah komputasi yang diterapkan pada algoritma untuk menyelesaikan suatu masalah. Secara informal, algoritma yang dapat menyelesaikan masalah dalam waktu

yang relatif singkat memiliki kompleksitas yang rendah, sedangkan algoritma yang dapat menyelesaikan masalah dalam waktu yang lebih lama memiliki kompleksitas yang lebih tinggi.

Dengan mempelajari informasi dan algoritma, kita belajar tentang algoritma dan penggunaannya dalam program komputer. Kompleksitas masalah ini terletak pada kenyataan bahwa kita sulit memahami algoritma dan solusi dari masalah yang muncul, dan juga sulit membayangkan struktur database yang akan digunakan. Memahami solusi masalah akan membantu kita mengingat dan memahami apakah masalah dapat disajikan secara visual dan imajinatif untuk membuat pertunjukan lebih menarik.

Pada permasalahan di atas, penulis ingin membantu menyederhanakan algoritma untuk mempermudah penyelesaian matematika dengan membuat perangkat komputer untuk informasi dan algoritme.

II. PEMBAHASAN HASIL STUDI LITERATUR

A. Sejarah Algoritma Pemrograman

Kata “Algoritma” berasal dari ilmuwan Arab Abu Jafar Mohammed bin Musa al-Khawarizmi, sarjana Persia yang menulis buku “Al-Jabar wal Mukabala” (rules of restoration and eduction). Dia menulis pada tahun 825 M. Orang-orang membaca Guarism sebagai algoritma Barat yang kemudian secara bertahap disesuaikan dengan algoritma Indonesia.

Kata algoritma sangat erat kaitannya dengan kata logika, yang berarti kemampuan manusia untuk berpikir kritis terhadap suatu masalah yang akan menghasilkan suatu fakta yang realistis, dapat diverifikasi, dan dapat diterima oleh akal. Pikiran sering dikaitkan dengan kecerdasan. Seseorang yang dapat berpikir jernih sering disebut sebagai orang yang cerdas. Ketika suatu masalah dipecahkan, sangat penting untuk menghitung. Pikiran identik dengan pikiran dan intelek. Mindfulness adalah salah satu bentuk berpikir. Asumsi pengetahuan tidak langsung didasarkan pada pernyataan pemikiran langsung yang mungkin benar atau tidak. Pengertian informasi sangatlah sederhana, ilmulah yang memberikan prinsip-prinsip yang harus diikuti agar dapat berpikir dengan benar sesuai dengan kaidah-kaidah yang relevan. Oleh karena itu, secara umum algoritma dapat didefinisikan sebagai serangkaian langkah pemecahan masalah, yang secara khusus disusun dengan menggunakan metode atau bahasa yang sesuai.

Sedangkan untuk bahasa “pemrograman” dibuat oleh Korad Zuse, seorang ilmuwan dari Jerman. Dia memberikan pengembangan untuk komputer Z-1 pada tahun 1935, sehingga

komputer tersebut menggunakan metode biner dalam sistemnya. Karena dia adalah orang pertama yang menjadi pencipta bahasa pemrograman atau kalkulator papan, dia dianggap sebagai alat dalam komputer digital.

B. Pengertian Pemrograman

Algoritma didefinisikan sebagai urutan yang digunakan untuk memecahkan masalah tertentu. Dalam bahasa pemrograman, algoritma sering disebut sebagai design wizard untuk menjalankan sebuah program.

Maksud dari pernyataan di atas adalah untuk lebih memperjelas bahwa jika program perlu menyelesaikan masalah menggunakan perkiraan, mereka harus diselesaikan dengan perbandingan. Namun, jika Anda hanya membutuhkan objek mental untuk menyelesaikan masalah, ini cukup untuk menyelesaikannya dengan menggunakan teknik sederhana. Oleh karena itu, algoritma aplikasi program ini harus disesuaikan dengan mata pelajaran yang terkait dengan program tersebut.

Sebenarnya ada dua cara untuk menentukan penggunaan algoritma komputer sebagai langkah awal sebuah program, seperti:

- algoritma disajikan menggunakan gambar atau grafik
- Algoritme disediakan menggunakan skrip

Desain algoritma dokumen ini biasanya melibatkan pseudocode, sedangkan algoritma untuk menggambar gambar sering kali menyertakan gambar (flowchart).

Tantangan bagi seorang programmer, bagaimanapun, adalah untuk mengimplementasikan sebuah algoritme yang merupakan kode atau skrip yang

telah ditentukan sebelumnya untuk dijalankan sesuai keinginan. Untuk alasan ini, penting bagi pengembang untuk mengetahui lebih banyak tentang algoritme perangkat lunak agar berhasil membangun program atau aplikasi tingkat perangkat lunak.

C. Fungsi Algoritma Pemrograman

Tujuan utama dari algoritma program adalah untuk memecahkan masalah. Algoritme perangkat lunak memberikan efisiensi dan keandalan yang tinggi saat merancang sebuah program. Berikut ini adalah beberapa kegiatan dari program tersebut, antara lain:

- a. Perangkat lunak dapat memecahkan masalah yang kompleks, dan perangkat lunak dapat memecahkan masalah perangkat lunak yang menggunakan perbandingan tingkat lanjut. Biasanya, program dengan perbandingan tinggi sering memiliki kesalahan dalam programnya, dan masalah ini dapat diatasi dengan menggunakan algoritma dalam program.
- b. Algoritma pemrograman dapat menyederhanakan pemrograman, program yang sebelumnya kompleks dan sulit akan disederhanakan dan disederhanakan menjadi algoritma pemrograman, sehingga program menjadi lebih efisien dan efektif. Juga, algoritmenya bercabang dua, yang merupakan cara top down dan cara divide and conquer.
- c. Algoritma ini dapat digunakan kembali, yaitu tidak menggunakan satu saja, yang berguna untuk mengurangi pengulangan. Oleh karena itu, program yang sama tidak boleh ditulis, yang akan

menyederhanakan program setelah program selesai.

- d. Saat membuat program, sering terjadi kesalahan. Kesalahan dalam suatu program akan mudah dideteksi dengan menggunakan fungsi algoritmik. Selain penggunaan fungsi algoritma, kesalahan ini juga dapat diperbaiki dengan cepat.
- e. Algoritma program memiliki alur yang jelas, ketika ada kesalahan, kita dapat dengan mudah menemukan kesalahan program. Hal ini karena perangkat lunak ini terorganisir dengan baik dan terorganisir dengan baik, sehingga memudahkan kita untuk mengikuti masalah yang dihadapi dalam program di komputer kita.

D. Perbedaan Algoritma dan Pemrograman

Niklaus Wirth mengatakan: $\text{program} = \text{Algoritma} + \text{program bahasa}$. Karena bahasa pemrograman dan algoritma sangat erat kaitannya dengan program. Algoritma yang baik tanpa memilih format data yang tepat akan mengurangi efisiensi program, begitu juga sebaliknya. Sehubungan dengan program, algoritma harus mengikuti tiga aturan, sebagai berikut:

1. Pembuatan atau penulisan algoritme tidak bergantung pada bahasa umum apa pun, yang berarti bahwa penulisan algoritme tidak bergantung pada bahasa program dan komputer yang digunakan
2. Notasi algoritma dapat diterjemahkan ke dalam beberapa bahasa pemrograman.
3. Terlepas dari bahasa pemrograman, prosesnya sama karena algoritmanya sama.

E. Kriteria Algoritma

Menurut Donald E. Knuth, algoritma yang baik memiliki kriteria sebagai berikut:

1. Input

Suatu algoritma harus memiliki 0 (nol) atau lebih (digunakan) parameter. Ini berarti bahwa suatu algoritma tidak dapat memiliki input langsung dari pengguna, tetapi juga dapat memiliki beberapa komponen. Algoritma tidak memiliki data langsung dari pengguna, sehingga semua data dapat dikompresi atau dibangkitkan dalam algoritma.

2. Output

Suatu algoritma harus memiliki satu atau lebih algoritma. Algoritma tanpa tindakan adalah algoritma yang tidak berguna, yang tidak perlu dilakukan. Algoritma dibuat dengan tujuan menghasilkan persyaratan, seperti status proyek

3. Keterbatasan

Semua pekerjaan yang harus dilakukan harus dihentikan. Demikian pula, harus dibuktikan bahwa algoritma berhenti setelah beberapa langkah proses dilakukan.

4. Kepastian

Itu tidak menyebabkan perilaku ambigu. Setiap baris tindakan / informasi dalam suatu algoritma harus aman, yang berarti tidak menghasilkan informasi tambahan untuk semua pembaca algoritma, untuk mendapatkan fungsi yang diharapkan dari pengguna.

5. Efektivitas

Langkah-langkah algoritma dilakukan dalam waktu yang wajar. Algoritma tidak mengandung tindakan yang tidak perlu dilakukan. Tindakan apa pun akan membutuhkan waktu untuk dieksekusi, bahkan jika tindakan itu jelas tidak berpengaruh atau tidak berguna. Misalnya: jika tindakan $X \leftarrow X + 0$. Jadi tindakan ini tentu saja tidak berpengaruh dan tidak

berguna karena $X + 0$ juga akan menghasilkan angka X yang tidak berguna. Oleh karena itu tidak perlu dilakukan karena tidak ada gunanya.

F. Struktur Dasar Algoritma

Struktur dasar dari algoritma adalah sebagai berikut:

a. Urutan

Dalam urutan ini, langkah-langkah yang dilakukan dalam algoritma secara berurutan. Mulailah dengan yang pertama, kedua, dan seterusnya. Faktanya, sebuah program menjalankan proses dari fondasi seperti struktur ini. Misalnya, perhatikan urutan angka di bawah ini, $(4 + 3) \times 7 - 49$, tetapi jika urutan tindakan berubah, tindakan akan berbeda dari $4 + (3 \times 7) = 25$.

Contoh lain ialah urutan operasi algoritma perbandingan dua-algoritma, yaitu:

1. Deklarasikan A, B, dan C sebagai bilangan bulat
2. Tambahkan nilai A dan B
3. Tambahkan nilai A ke bagian C
4. Tambahkan nilai B ke bagian A
5. Tambah nilai C ke bagian B

a. Struktur seleksi

Struktur seleksi menentukan pilihan langkah berdasarkan situasi atau keputusan. Struktur ini selalu diwakili dengan membuat serangkaian bidang (flowchart yang berbentuk belah ketupat)). Ada banyak contoh yang dapat kita gunakan dalam jenis struktur ini dalam hal

keputusan, termasuk: potongan yang berbeda berdasarkan jumlah barang yang akan dibeli dan contoh model pemilihan untuk menghitung jumlah yang terdaftar. di bawah:

1. Tambahkan bilangan seperti suatu bilangan bulat
2. Buat bilangan menggunakan angka 2, simpan nilai sisa pembagian dalam variabel sisa
3. Jika nilai sisa adalah 0, lakukan langkah 4:
4. Tunjukkan "ganjil" di layar.
5. Jika nilai residu tidak 0, lakukan langkah 6
6. Tunjukkan "Ganjil" di layar.
7. Sudah berakhir.

b. Struktur perulangan

Struktur ini menyediakan instruksi atau tindakan yang akan dieksekusi beberapa kali. Misalnya, jika seorang teman ingin menulis kata "belajar c" sepuluh kali. Akan lebih efektif jika teman Anda menggunakan struktur ini daripada buku sepuluh baris. Contohnya adalah menampilkan huruf khusus n kali di layar sebagai berikut:

1. Jelaskan perubahan font untuk menyimpan font yang ditampilkan.
2. Identifikasi bakat n untuk menyimpan nomor seri
3. Tunjukkan keberhasilan menggunakan rak sebagai rak prefabrikasi.
4. Masukkan sesuatu karakter dan simpan di bawah karakter khusus

5. Masukkan nomor pengulangan yang diinginkan dan simpan sesuai kebutuhan
6. Atur nilai default ke 0
7. Tunjukkan huruf-huruf itu ke matamu
8. Tambahkan baki 1
9. Jika nilai tumpukan $< n$, lakukan Langkah 6
10. Jika persamaan $= n$ selesai

G. Flowchart

Flowchart dapat didefinisikan sebagai alat atau metode yang menunjukkan langkah-langkah yang perlu diambil untuk memecahkan suatu masalah dengan menampilkan serangkaian indikator tertentu. Manfaat yang diperoleh dengan menggunakan diagram alur untuk memecahkan masalah teknis:

- a. Terbiasa berfikir secara sistematis dan terstruktur
- b. Menjadi mudah mengecek dan menemukan bagian-bagian prosedur yang tidak valid dan bertele-tele
- c. Prosedur akan sangat mudah dikembangkan

Flowchart membantu memahami urutan-urutan logika yang rumit dan panjang. Flowchart membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrograman) akan lebih mudah.

Simbol Flowchart

SIMBOL	NAMA	FUNGSI
Terminator	Permulaan / akhir program
Garis alir (flow line)	Arah aliaran program
Preparation	Proses inisialisasi (pemberian harga awal)
Proses	Proses Perhitungan (pengolahan data)
Input / Output Data	Proses input/output data, parameter, informasi
Predefined Proses (sub program)	Permulaan sub program/proses menjalankan sub program
Decision	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
On Page Connector	Penghubung bagian" flowchart yang berada pada satu halaman
Off Page Connector	Penghubung bagian" flowchart yang berada pada halaman berbeda

Flowchart di bagi menjadi 5 jenis flowchart, yaitu ada *system flowchart*, *document flowchart*, *schematic flowchart*, *program flowchart*, *process flowchart*.
System Flowchart

a. System Flowchart (Diagram alir sistem)

Diagram alir sistem dapat didefinisikan sebagai diagram yang menunjukkan alur kerja sistem secara keseluruhan. Diagram ini menggambarkan alur prosedur dalam sistem. Diagram alir sistem menunjukkan apa yang dilakukan dalam sistem.

b. Document Flowchart (Diagram alir dokumen)

Bagan alir dokumen (document flowchart) atau disebut juga bagan alir formulir (form flowchart) atau paperwork flowchart merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusannya.

c. Schematic Flowchart (Diagram alir skematik)

Bagan alir skema adalah bagan alir, mirip dengan bagan alir sistem, yang

dimaksudkan untuk menggambarkan operasi suatu sistem. Perbedaannya adalah diagram alir skema, selain menggunakan ikon diagram alur sistem, juga menggunakan gambar komputer dan perangkat lain yang digunakan. Menggunakan gambar-gambar ini mudah dimengerti tetapi sulit dan memakan waktu untuk menggambar.

d. Flowchart Program (Diagram alir program)

Diagram alir program adalah diagram yang menguraikan langkah-langkah dalam proses program. Flowchart program diturunkan dari flowchart sistem. Bagan alir program dapat terdiri dari dua jenis, yaitu bagan alur logika program dan bagan alur program komputer terperinci. Diagram alur logika program digunakan untuk menggambarkan secara logis setiap langkah dalam program komputer. Diagram instrumen logika program ini dibuat oleh analis sistem. Bagan alir program komputer yang terperinci sekarang digunakan untuk menggambarkan instruksi program komputer yang terperinci. Flowchart ini dibuat oleh sistem program.

e. Process Flowchart (Diagram alir proses)

Diagram alir proses adalah diagram alir yang banyak digunakan dalam industri rekayasa. Flowchart ini juga berguna bagi seorang analis sistem untuk menggambarkan proses dalam suatu prosedur.

Berikut ini digunakan notasi atau simbol dari diagram alir proses, yang dapat dibagi menjadi 3 (tiga) kelompok sebagai berikut:

1) Flow direction symbol (simbol penghubung/alur)

Simbol digunakan untuk menghubungkan satu simbol dengan simbol lainnya. Simbol

ini disebut juga garis penghubung, simbol-simbolnya adalah sebagai berikut:

Flow direction symbol

	Simbol arus / flow, yaitu menyatakan jalannya arus suatu proses
	Simbol communication link, yaitu menyatakan transmisi data dari satu lokasi ke lokasi lain
	Simbol connector, berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
	Simbol offline connector, menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda

2) Processing Symbol (simbol proses)
Simbol ini menunjukkan jenis pemrosesan dalam proses/prosedur. Simbol-simbolnya adalah sebagai berikut:

Processing Symbols

	Simbol process, yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
	Simbol manual, yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
	Simbol decision, yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak
	Simbol predefined process, yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
	Simbol terminal, yaitu menyatakan permulaan atau akhir suatu program
	Simbol keying operation, Menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
	Simbol offline-storage, menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
	Simbol manual input, memasukkan data secara manual dengan menggunakan online keyboard

3) Input/ Output symbols (simbol masukan/ pengeluaran)
Ikon/ simbol yang menunjukkan jenis perangkat yang digunakan sebagai media input atau output. Simbol Adapun adalah:

Input-Output Symbols

	Simbol input/output, menyatakan proses input atau output tanpa tergantung jenis peralatannya
	Simbol punched card, menyatakan input berasal dari kartu atau output ditulis ke kartu
	Simbol magnetic tape, menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis
	Simbol disk storage, menyatakan input berasal dari disk atau output disimpan ke disk
	Simbol document, mencetak keluaran dalam bentuk dokumen (melalui printer)
	Simbol display, mencetak keluaran dalam layar monitor

H. Pseudocode (kodesemu)

Pseudocode adalah pengembangan dari algoritma dimana pseudocode, seperti namanya, menggunakan spesifik atau pseudocode untuk menjelaskan operasi rinci atau solusi dari suatu masalah. Singkatnya, pseudocode dapat dilihat sebagai algoritma yang diubah menjadi pseudocode tertentu.

Kelebihan dari pseudocode ini adalah memiliki struktur yang jauh lebih kompak dibandingkan dengan algoritma, namun tetap mudah dipahami oleh manusia dan memiliki struktur yang hampir mirip dengan pemrograman. Pseudocode menggunakan perintah yang sangat umum yang masih banyak kita pelajari, misalnya IF, ELSE, THEN dan penggunaan kode atau karakter tertentu, seperti koma (,) dan tanda tanya (?), Tanda seru, tanda kutip, kiri dan panah kanan, tanda pagar dan tanda lainnya.

Untuk dapat membuat pseudocode, setidaknya kita harus mengetahui terlebih dahulu algoritma dari suatu masalah. Setelah itu diubah ke bentuk yang lebih ringkas dan detail dengan menambahkan kode dan karakter tertentu.

Ciri-ciri Pseudocode:

1. Pseudocode adalah kode/notasi yang menyerupai suatu masalah atau memberikan penjelasan tentang cara menyelesaikan suatu masalah
2. Pseudocode sering digunakan oleh seseorang untuk menulis suatu algoritma masalah.
3. Pseudocode berisi langkah-langkah untuk menyelesaikan masalah (hampir identik dengan algoritma), hanya bentuknya yang sedikit berbeda dengan algoritma.
4. Pseudocode menggunakan bahasa yang mirip dengan bahasa pemrograman. Selain itu, pseudocode biasanya menggunakan bahasa yang umumnya mudah dipahami dan juga lebih ringkas daripada algoritma.
5. Tidak ada aturan ketat untuk menulis pseudocode.

Notasi Pseudocode

Merumuskan langkah-langkah di tingkat tertinggi terlepas dari bahasa pemrograman.

Contoh:

Temukan yang lebih besar dari dua angka yang dimasukkan. Solusi kode semu:

1. Tambahkan bilangan pertama
2. Tambahkan bilangan kedua
3. Kalau bilangan pertama > bilangan kedua jadi tindakan ke 4, jika tidak maka lakukan tindakan ke 5.
4. Paparkan bilangan pertama
5. Paparkan bilangan kedua.

Solusi Algoritma:

1. Tambahkan bilangan pertama atau A
2. Tambahkan bilangan kedua atau B

3. Jika $a > b$ maka lakukan langkah ke 4
4. Print bagian a
5. Print bagian b

algoritma	pseudocode
Nilai A ditambah dengan 5	$A \leftarrow A + 5$
Cetak Nilai A bilalebih besar dari 10	IF >10 THEN PRINT A
Dari dua buah bilangan A dan B, cari bilangan yang terbesar	IF A > THEN PRINT A ELSE PRINT B

Gambar Contoh pseudocode dan algoritma

Struktur algoritma pseudocode

Pseudo berarti tidak nyata, palsu atau samar. Kode semu adalah notasi surealis kode yang mewakili logika algoritma dan bahasa pemrograman. Tidak seperti bahasa pemrograman, kode semu tidak memiliki aturan penulisan khusus. Petunjuk penulisan algoritma pseudocode adalah sebagai berikut:

1. Memiliki bagian header yang menunjukkan nama algoritma, komentar dan deklarasi.
2. Memiliki bagian utama dari algoritma yang menunjukkan esensi dari proses.
3. Melakukan bagian terakhir dari proses algoritma.
4. Deklarasi pada header tidak berbeda secara signifikan dengan notasi algoritma.
5. Penugasan (perintah atau penugasan) saat menetapkan nilai ke variabel melalui format $\leftarrow$
6. Banyak penggunaan bahasa Inggris seperti IF, Read, Write dll.
7. Komentar ditulis dalam tanda kurung {... Komentar}

I. Cara Kerja Algoritma pemrograman

Komputer hanyalah alat untuk beroperasi. Penting bagi kita untuk menulis algoritma dalam bahasa pemrograman sehingga dapat digunakan di komputer, seperti pemrograman. Dari definisi tersebut dapat didefinisikan sebagai program sebagai suatu fungsi atau metode dari suatu urutan algoritmik yang ditulis dalam bahasa pemrograman tertentu sehingga dapat dieksekusi di komputer. Saat digunakan, kata "algoritma" dan "program" sering digunakan secara bergantian.

Misalnya, ketika seseorang mengatakan "program klasifikasi data menggunakan algoritma klasifikasi selektif." Ketika seseorang bertanya "bagaimana algoritma dan program menginterpretasikan grafik?". Perbedaan antara arti kata "algoritma" dan kata "program" akan kita ketahui jika kita memahami arti dari algoritma dan program, seperti yang telah disebutkan. Program adalah implementasi dari algoritma dalam bahasa pemrograman dan algoritma adalah metode atau sarana untuk memecahkan masalah.

Ketika sebuah program dijalankan dalam bahasa pemrograman, program adalah istilah lain untuk program atau program. Pemrograman adalah nama panggilan untuk programmer. Sebuah instruksi atau instruksi adalah representasi untuk setiap metode yang terdapat dalam suatu sistem. Jika sebuah instruksi dieksekusi, komputer melakukan proses yang sesuai seperti yang ditunjukkan. Secara umum, komputer terdiri dari empat elemen penting, yang disebut perangkat keras, perangkat lunak, memori, dan unit produksi.

Otak komputer adalah unit produksi utama (CPU), unit produksi utama komputer. CPU bekerja pada implementasi proses awal, penjadwalan proses,

perbandingan proses, elaborasi dan elaborasi. Elemen komputer berguna untuk penyimpanan dan penyimpanan data memori. Memori menyimpan program yang berisi penggunaan CPU dan file yang akan diproses.

Perangkat I/O diproduksi dan digunakan, fungsi dari perangkat tersebut adalah untuk memasukkan data ke dalam suatu aplikasi. Perangkat ini juga digunakan oleh komputer untuk mengkomunikasikan hasil tindakan mereka. Contoh perangkat adalah keyboard, pemindai, dan drive disket. Sementara pola tampilan antarmuka perangkat adalah (Monitor), printer adalah (Printer), dan disket adalah (Disk). Berikut ini menjelaskan proses untuk keempat item di atas.

Awalnya, memori komputer mengingat dan mengelola sebuah program. Segera setelah program pemantauan dijalankan, perintah pemantauan dikeluarkan ke CPU. Setelah selesai, CPU melakukan proses sesuai penugasan semula. Ketika suatu proses membutuhkan file, file perangkat pengguna dikenali dan kemudian disimpan dalam memori setelah dikirim ke CPU oleh proses jika diperlukan. Saat game memberikan aksi dengan catatan, aksi tersebut disimpan sebagai memori, kemudian memori menulis aksi ke mesin produksi dan kemudian menampilkannya di layar monitor.

J. Mekanisme Penerapan Algoritma

Beberapa alat langkah demi langkah yang membangun dan mengintegrasikan algoritma dalam fase pengembangan perangkat lunak dibagi menjadi beberapa bagian, termasuk:

1. Informasi variabel

2. Informasi variabel diperlukan ketika bahasa pemrograman membutuhkannya dan tidak diperlukan ketika bahasa pemrograman tidak memerlukannya.
3. Pilih tipe data program di sini diperlukan. Jika data yang diusulkan relevan, ini harus diperhitungkan saat memilih tipe data.
4. Petunjuk penggunaan, Beberapa aturan yang ada bekerja dengan cara yang sama, tetapi setiap perintah memiliki kelebihan atau kekurangan yang berbeda.
5. Aturan prosedural, Saat menulis sebuah program, programmer bergantung pada struktur sintaksis yang disediakan dalam bahasa pemrograman yang ingin digunakan oleh programmer.
6. Tampilkan hasilnya, Saat membuat algoritma, programmer tidak tahu bagaimana program akan muncul. Faktor-faktor tersebut harus diperhitungkan ketika pindah ke program.
7. Langkah-langkah dalam penerjemahan atau translation.

III. KESIMPULAN

Algoritma adalah langkah tertulis dan berurutan untuk memecahkan masalah dan algoritma pemrograman adalah langkah tertulis berurutan untuk menyelesaikan masalah pemrograman. Kita dapat menggunakan algoritma untuk memecahkan masalah apa pun dari yang sederhana hingga yang kompleks. Namun, pengguna harus dapat membuat program menggunakan bahasa yang dimengerti komputer. Sebelum disajikan dalam bentuk bahasa pemrograman, ada baiknya

kita membuat diagram alir (flow chart) dan pseudocode. Hal ini untuk memudahkan kinerja atau untuk menyederhanakan program. Selain itu, algoritma dapat menyelesaikan masalah logika dan masalah matematika secara berurutan, tetapi terkadang algoritma tidak selalu dapat berurutan, yang dapat disebut sebagai proses percabangan.

Dalam algoritma, kita harus bisa menguasai dasar-dasar algoritma dan menyelesaikan masalah lain secara efektif dan cepat. Ini membuatnya sangat mudah untuk mengimplementasikan algoritma yang telah kita kuasai dalam program.

DAFTAR PUSTAKA

- Nuraini, Rini. 2015. *Desain algoritma operasi perkalian matriks menggunakan metode flowchart*. Jurnal teknik komputer amik bsi. Vol.1. No. 1. 144-151.
- Maulana, Gun Gun. 2017. *Pembelajaran dasar algoritma dan pemrograman menggunakan el-goritma berbasis web*. Jurnal teknik mesin. Vol. 06, ISSN : 2549-2888
- Bagus, Kadek, and Udayana, Feritsha. 2018. *Penerapan komponen dan struktur algoritma pada algoritma dan pemrograman dasar*. Jurnal bisnis & teknologi politeknik NSC surabaya. Vol 5, no 1, ISSN : 2355-8865
- Resmawan. 2019. Pengantar algoritma dan pemrograman . repository.ung.ac.id . 1-21.
- Bibi, Sarah, and Handaru Jati. 2015. *Efektivitas model blended learning terhadap motivasi dan tingkat pemahaman mahasiswa mata kuliah algoritma dan pemrograman*. Jurnal pendidikan, Vol 5, no 1. 74-75.
- Munir. 2016. *Kontribusi Matematika dalam TIK*. [Online]. Tersedia :

- <https://file.upi.edu/Direktori/FPMIPA/P RODI. ILMU KOMPUTER/19660325 2001121-MUNIR/Presentasi TIK/Kontribusi Matematika dalam TIK.pdf>. [22 september 2016].
- Ridho, dkk. 2013. *Logika dan Algoritma*. [Online]. Tersedia: <http://entin.lecturer.pens.ac.id/Logika%20Algoritma/Buku%20Logika%20Algoritma.pdf>. [23 mei 2017].
- Retta, Allen Marga dkk. Bulan Tahun. *Pengaruh penerapan algortima terhadap pembelajaran pemrograman komputer*. INDIKTIKA (Jurnal Inovasi Pendidikan Matematika). Vol 2, no 2 ISSN : 2655-2752
- Yuniansyah. 2020. *Algoritma dan Pemrograman Menggunakan Bahasa Pemrograman Jawa (Teori dan Aplikasinya)*. Leuwiliang, Bogor. Lindan Bestari.
- Sitorus, Lamhot. 2015. *Algoritma dan Pemrograman*. CV. ANDI OFFSET, Yogyakarta.
- Hilmi, Novian Dwi Syahrizal & Rosyid, Harits Ar. 2020. *Pengembangan Sistem Kuis Algortima Pemrograman Berbasis Web*. Belantika pendidikan, Vol. 3 no. 1, pp 66-74