



# PETIR

JURNAL  
PENGKAJIAN DAN PENERAPAN  
TEKNIK INFORMATIKA

VOLUME 10 - NOMOR 1

MARET 2017

ISSN 1978-9262

MODEL DATA LOGGER UNTUK MENGUKUR ARUS, TEGANGAN, DAN DAYA PADA SIMULASI PEMBANGKIT LISTRIK TENAGA ANGIN DAN SURYA MENGGUNAKAN ANDROID

*Abdurrasyid; Diko Suprayogi*

SISTEM PENGAMBILAN KEPUTUSAN DALAM PEMILIHAN JURUSAN MENGGUNAKAN METODE EKSPONENSIAL (MPE) DI PERGURUAN TINGGI NEGERI DAN SWASTA DI JAWA BARAT

*Andri Sahata Sitanggang*

RANCANG BANGUN ANJUNGAN BIMBINGAN TUGAS AKHIR SECARA ONLINE (STUDI KASUS : JURUSAN TEKNIK INFORMATIKA STT-PLN)

*Dian Hartanti; Wisnu Hendro Martono*

FUZZY CLUSTERING MEANS (FCM) DALAM PENENTUAN LOKASI PENERTIBAN PENYAKIT MASYARAKAT PADA KEGIATAN PEMBINAAN SOSIAL SATPOL-PP WILAYAH SUMATRA-BARAT

*Dine Tiara Kusuma; Rakhmadi Irfansyah Putra*

METODE RANCANG BANGUN PEMAHAMAN PANCASILA PADA MAHASISWA TEKNIK SEKOLAH TINGGI TEKNIK PLN

*Emillia; Intan Ratna Sari Yanti*

PEMANFAATAN AUGMENTED REALITY UNTUK PENGENALAN HARDWARE KOMPUTER PADA SEKOLAH DASAR BERBASIS ANDROID

*Harni Kusniyati; Raka Yusuf; Mohamad Aris Widyartanto*

IMPLEMENTASI AUDIT SISTEM CONTACT CENTER MENGGUNAKAN KERANGKA KERJA COBIT 4.1 DOMAIN DELIVERY AND SUPPORT (STUDI KASUS : PT VISIONET INTERNATIONAL)

*Muhaimin Hasanudin*

DESK CHECK TABLE PADA FLOWCHART OPERASI PERKALIAN MATRIKS

*Rini Nuraini*

APLIKASI MONITORING KEGIATAN PETUGAS PEMELIHARA SARANA DAN PRASARANA UMUM BERBASIS WEBSITE

*Syam Gunawan*

METODE FUZZY SIMPLE ADDITIVE WEIGHTING (SAW) DALAM PEMILIHAN KETUA OSIS

*Adi Supriyatna; Dewanto Ekaputra*

RANCANG BANGUN APLIKASI LOKASI PARIWISATA PROVINSI SUMATERA BARAT BERBASIS ANDROID

*Dwina Kuswardani; Dioreza*

PERANCANGAN APLIKASI MEDIA PEMBELAJARAN KERAGAMAN SUKU BANGSA DAN BUDAYA DI INDONESIA BERBASIS MULTIMEDIA

*Yasni Djamain; Intan Ratna Sari Yanti; Santria Jaula Tama*

ISSN 1978-9262



771978 926272

SEKOLAH TINGGI TEKNIK - PLN (STT-PLN)

PETIR

VOL. 10

NO. 1

HAL. 1 - 90

JAKARTA, MARET 2017

ISSN 1978-9262

# DESK CHECK TABLE PADA FLOWCHART OPERASI PERKALIAN MATRIKS

Rini Nuraini

Universitas Bunda Mulia Jakarta  
rini.nuraini@mercubuana.gmail.com

## ABSTRAK

Untuk menentukan output suatu pseudocode algoritma biasanya mengalami kesulitan, terutama pada logika algoritma repetition control structure atau nested looping, yang biasanya rumit, sehingga sulit untuk ditebak atau ditentukan outputnya. Misalkan, apakah akhir dari suatu looping? Dimanakah batas awal (start) dan batas akhir (stop) dari suatu looping? Mulai dari manakah dan sampai dimanakah batas badan looping? Apakah harus looping atau berhenti? Apakah hasil looping bernilai true atau false? Apakah looping selama hasil boolean atau kondisi bernilai true? Apakah looping sampai kondisi bernilai true? Selain hal tersebut, mengingat keterbatasan nalar manusia untuk mengingat atau melakukan hal tersebut, maka kepentingan akan suatu tools sangat sekali dibutuhkan dalam hal menganalisa ini. Berdasarkan kondisi permasalahan itu, penulis mencari suatu rancangan atau suatu metode, bagaimana cara mudah untuk menganalisa variable baik input atau output beserta logika algoritmanya pada suatu pseudocode algoritma, inilah tujuan penulis merancang suatu tabel, yang penulis bernama desk check table. Dalam hal ini fungsi desk check table adalah mendokumentasikan history dari setiap perubahan sejumlah variabel yang dianalisa, baik variabel input atau output beserta logikanya dengan menggunakan suatu tabel. Artikel ini juga, sekaligus sebagai artikel lanjutan dari artikel penulis sebelumnya, dengan judul "**Algorithm Design Of Operation Multiplication Matrix By Using Flowchart Method**". Berarti tujuan dari artikel ini adalah menganalisa kebenaran jawaban-jawaban yang sudah penulis jawab pada artikel tersebut, serta pembuktian yang akan penulis jadikan contoh soal, yaitu pada soal Aljabar Linier mengenai Perkalian Matriks, sama halnya dengan artikel sebelumnya.

**Kata Kunci:** Algorithm Analysis, Operation Multiplication Matrix, Pseudocode, Desk Check Table.

## I. PENDAHULUAN

Ada beberapa langkah dasar yang perlu untuk diikuti dalam pembuatan suatu algoritma, antara lain adalah: pernyataan masalah; membangun model dari suatu masalah; merancang algoritma dari model; menguji kebenaran algoritma; implementasikan dengan suatu bahasa pemrograman seperti C, Java, dan lain-lain; dokumentasi; dan analisa kompleksitas algoritma seperti analisa output dengan menggunakan desk check table, space complexity, dan time complexity.

Pembuatan algoritma mempunyai banyak keuntungan diantaranya: pembuatan atau penulisan algoritma tidak tergantung pada bahasa pemrograman apapun, artinya penulisan algoritma independen dari bahasa pemrograman dan computer yang melaksanakannya; notasi algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman; apapun bahasa pemrogramannya, output yang akan dikeluarkan sama, karena algoritmanya sama.

Permasalahan atau soal-soal science khususnya kalkulus, bagi sebagian orang dianggap sulit untuk dipahami, seperti bagaimana urutan langkah-langkahnya, urutan logikanya, pengambilan keputusannya, dan proses aritmatikanya. Berdasarkan hal tersebutlah, penulis berkeinginan mengangkat salah satu soal kalkulus, yaitu Integral Tentu untuk dijadikan contoh soal dalam artikel ini, untuk dibuatkan metode flowchart dan pseudocodenya.

Tujuannya adalah untuk mempelajari dan memahami contoh soal tersebut dengan menggambarkan urutan logika, pengambilan

keputusan, dan proses aritmatikanya, dengan menggunakan simbol, sehingga mudah dipahami. Simbol tersebut adalah simbol-simbol dalam flowchart, yang merupakan suatu alat atau sarana yang menunjukkan langkah-langkah yang harus dilaksanakan dalam menyelesaikan suatu permasalahan untuk komputasi dengan cara mengekspresikannya ke dalam serangkaian simbol-simbol grafis khusus.

Pseudocode adalah deskripsi dari algoritma pemrograman komputer yang menggunakan struktur sederhana dari beberapa bahasa pemrograman tetapi bahasa tersebut hanya ditujukan agar dapat mudah dibaca manusia. Biasanya yang ditulis dari pseudocode adalah variabel dan function. Fungsi dari pseudocode sama dengan Flowchart. Perbedaannya terletak pada cara penyampaiannya. Pseudocode menggunakan kata-kata untuk menjelaskan suatu algoritma, sedangkan Flowchart menggunakan gambar.

## II. TINJAUAN PUSTAKA

### 1. Operation Multiplication Matrix

Data seringkali disusun dalam suatu rangkaian di mana setiap elemen diberi indeks dengan satu atau lebih subskrip, susunan ini disebut suatu array. Banyaknya subskrip disebut dimensi array tersebut. (Hal 67 Matematika Dasar Yusuf Yahya, etc).

Menurut Suryadi (1990, 65), bahwa matriks adalah himpunan skalar (bilangan riil atau kompleks) yang disusun/dijajarkan secara empat



persegi panjang (menurut baris-baris dan kolom-kolom). Skalar-skalar itu disebut elemen matriks.

Contoh matriks riil:

$$\begin{array}{|cccc|} \hline 2 & 3 & 1 & 1 \\ 4 & 0 & 0 & -3 \\ 7 & \sqrt{2} & 10 & 1 \\ \hline \end{array} \quad \begin{array}{l} \rightarrow \\ \rightarrow \\ \rightarrow \end{array} \quad \begin{array}{l} \text{2 baris} \\ \\ \end{array}$$

$$\begin{array}{l} \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \end{array} \quad \begin{array}{l} 2 \\ 3 \\ 4 \end{array} \quad \text{kolom}$$

Matriks diberi nama dengan huruf besar A, B, P, C, dan lain-lain. Secara lengkap ditulis matriks A = (a<sub>ij</sub>), artinya suatu matriks A yang elemennya a<sub>ij</sub> di mana indeks i menyatakan baris ke-i matriks j menyatakan kolom ke-j dari elemen tersebut.

Sebuah matriks A = (a<sub>ij</sub>), i = 1, 2, ..., m dan j = 1, 2, ..., n ; yang mana berarti bahwa banyaknya baris = m serta banyaknya kolom = n.

$$\begin{bmatrix} a_{11} & \dots \\ a_{21} & \ddots \\ \vdots & \dots \end{bmatrix}$$

## 2. Algoritma

Algoritma berasal dari nama seorang Ilmuwan Arab yang bernama Abu Ja'far Muhammad Ibnu Musa Al Khuwarizmi penulis buku berjudul Al Jabar Wal Muqabala (Buku Pemugaran dan Pengurangan). Kata Al Khuwarizmi dibaca orang barat menjadi Algorism yang kemudian lambat laun menjadi Algorithm diserap dalam bahasa Indonesia menjadi Algoritma. Algoritma dapat diartikan urutan langkah-langkah (instruksi-instruksi/aksi-aksi) terbatas untuk menyelesaikan suatu masalah.

Syarat-Syarat Algoritma menurut Donald E. Knuth, dalam <http://www.akmi-baturaja.ac.id/wp-content/uploads/2012/07/Logika-dan-Algoritma.pdf>, yaitu:

1. Finiteness (Keterbatasan)  
Algoritma harus berakhir setelah melakukan sejumlah langkah proses.
2. Definiteness (Kepastian)  
Setiap langkah algoritma harus didefinisikan dengan tepat dan tidak menimbulkan makna ganda
3. Input (Masukan)  
Sebuah algoritma memiliki nol atau lebih masukan (input) yang diberikan kepada algoritma sebelum dijalankan.
4. Output (Keluaran)  
Setiap algoritma memberikan satu atau beberapa hasil keluaran.
5. Effectiveness (Efektivitas)  
Langkah-langkah algoritma dikerjakan dalam waktu yang "wajar".

Suatu Algoritma dapat terdiri dari tiga struktur dasar, yaitu runtunan, pemilihan dan pengulangan. Berikut Penjelasan ringkas dari tiga struktur tersebut:

1. Runtunan  
Runtunan yaitu satu atau lebih instruksi yang dikerjakan secara berurutan sesuai dengan urutan penulisannya. Urutan dari instruksi menentukan hasil akhir dari suatu algoritma. Bila urutan penulisan berubah maka mungkin juga hasil akhirnya berubah.

2. Pemilihan  
Pemilihan yaitu instruksi yang dikerjakan dengan kondisi tertentu. Kondisi adalah persyaratan yang dapat bernilai benar atau salah. Instruksi hanya dilaksanakan apabila kondisi bernilai benar, sebaliknya apabila salah maka instruksi tidak akan dilaksanakan. Pernyataan kondisi menggunakan statemen If (jika) dan Then (maka).

3. Pengulangan  
Pengulangan merupakan pengulangan sejumlah aksi yang sama sebanyak jumlah yang ditentukan atau sesuai dengan kondisi yang diinginkan. Beberapa statemen pengulangan yaitu:
  - a) For ... To ... Do / For ... Downto ... Do
  - b) While ... Do
  - c) Repeat ... Until

Algoritma dapat ditulis dengan cara berikut:

1. Menggunakan bahasa natural
2. Menggunakan kode semu (pseudo-code)  
Teknik penulisan yang mendekati bahasa pemrograman tertentu
3. Menggunakan diagram alir (flowchart)  
Teknik penyajian dengan menggunakan simbol-simbol.

## 3. Array

Peubah atau variable hanya dapat menyimpan sebuah nilai saja. Peubah tidak dapat menyimpan beberapa buah nilai yang bertipe sejenis sekaligus. Sementara dalam kebutuhan pemrograman, seringkali kita diharuskan atau dibutuhkan mengolah sekumpulan data yang bertipe sama dalam yang bersamaan, misalnya dalam kasus menampung hasil ujian 100 orang mahasiswa, table harga-harga barang di swalayan, dan lain sebagainya. Dikarenakan setiap elemen data bertipe sama, maka elemen tersebut, cukup diacu dengan satu nama peubah, dan untuk membedakan elemen data yang satu dengan elemen data yang lainnya, maka elemen diacu dengan menggunakan indeks (*subscript*). Misalnya jika data nilai ujian dilambangkan dengan peubah A, maka indeksnya A<sub>i</sub> menyatakan nilai ujian mahasiswa yang ke-i.

Dalam matematika, statistik, atau bidang eksakta lainnya, sering ditemui besaran yang menggunakan nama peubah berindeks seperti: u<sub>1</sub>, u<sub>2</sub>, u<sub>3</sub>, u<sub>4</sub>, u<sub>5</sub>, u<sub>6</sub>, u<sub>7</sub>, u<sub>8</sub>, u<sub>9</sub>, u<sub>10</sub>; a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>, ..., a<sub>n</sub>; v<sub>k</sub> ≥ 0, untuk k = 0, 1, 2, ... , n; dan sebagainya. Besaran-besaran tersebut adalah sekumpulan nilai yang bertipe sama. Nama peubah yang menyatakan kumpulan nilai itu masing-masing adalah u, a, dan v. Nilai tertentu di dalam kumpulan peubah tersebut diacu dengan menggunakan indeksnya, misalnya u<sub>3</sub>, a<sub>8</sub>, a<sub>k</sub>, atau v<sub>j</sub>, dan lain-lain.

Dalam kegiatan pemrograman, sekumpulan data yang bertipe sama perlu disimpan sementara di dalam memori komputer untuk sewaktu-waktu dimanipulasi. Misalnya jika hendak menghitung nilai rata-rata kumpulan data nilai ujian, dengan rumus:

$$\text{Rata-rata} = (a_1 + a_2 + \dots + a_n) / n = \sum_{i=1}^n a_i$$

Sekumpulan data yang bertipe sama disimpan secara berurutan di dalam memori komputer, setiap elemen data diacu dengan menggunakan indeks. Indeks menyatakan posisi data relative di dalam kumpulannya. Struktur penyimpanan data seperti ini dinamakan larik (array). Nama lain untuk larik adalah table, vector, atau peubah majemuk (satu peubah mempunyai banyak elemen).

Menurut Rinaldi Munir (2011, hal.296) larik adalah struktur data yang menyimpan sekumpulan elemen yang bertipe sama, setiap elemen diakses langsung melalui indeksinya. Indeks larik haruslah tipe data yang menyatakan keterurutan, misalnya integer atau karakter.

### III. METODE PENELITIAN

#### 1. Flowchart

Menurut Edhy Sutanta (2004, hal. 28), flowchart dapat diartikan sebagai suatu alat atau sarana yang menunjukkan langkah-langkah yang harus dilaksanakan dalam menyelesaikan suatu permasalahan untuk komputasi dengan cara mengekspresikannya ke dalam serangkaian simbol-simbol grafis khusus. Manfaat yang akan diperoleh bila menggunakan flowchart dalam pemecahan masalah komputasi:

1. Terbiasa berfikir secara sistematis dan terstruktur
2. Mudah mengecek dan menemukan bagian-bagian prosedur yang tidak valid dan bertele-tele
3. Prosedur akan mudah dikembangkan

#### 2. Repetition Control Structures

Robertson, Lesley Anne (2004:54), *outline*:

##### a. Repetition Using The DOWHILE Structure

*The format is:*  
DOWHILE condition *p* is true  
statement block  
ENDDO

##### b. Repetition Using the REPEAT ... UNTIL Structure

*The format of the REPEAT ... UNTIL structure is:*  
REPEAT  
statement  
statement

...  
UNTIL condition is true

##### c. Counted Repetition

*Counted repetition occurs when the exact number of loop iterations is know in advance. The execution of the loop is controlled by a loop index, and instead of using DOWHILE, or REPEAT ... UNTIL, the simple keyword DO is used as follows:*

*DO loop\_index = initial\_value to final\_value  
statement block  
ENDDO*

#### 3. Struktur Kendali Pengulangan

Ngoen, Thompshon Susabda (2009:127-148), bahasa C menyediakan tiga instruksi untuk melakukan proses pengulangan: for, while, dan do while. Ketiga intruksi ini memiliki karakteristik masing-masing.

#### a. Instruksi For

Instruksi for digunakan untuk melakukan proses pengulangan yang frekuensi pengulangannya telah diketahui sebelum proses pengulangan dimulai.

**for ([expression1]; [expression2]; [expression3]) statement;**

Expression1 digunakan untuk melakukan proses awal atau inisialisasi, misalnya pemberian nilai awal kepada pencacah atau counter. Expression2 berupa ekspresi Boolean yang bila dikerjakan akan memberi nilai true (bukan nol) atau false (nol). Expression3 adalah instruksi pasca pengerjaan statement.

#### b. Instruksi While

Instruksi while ialah instruksi untuk melakukan proses pengulangan yang pemeriksaan syarat pengulangannya dilakukan pada awal proses. Instruksi while umumnya digunakan untuk melakukan proses pengulangan yang frekuensi pengulangannya belum diketahui pada saat proses pengulangan dimulai.

**while (expression) statement;**

Expression berupa ekspresi Boolean dan berfungsi sebagai control pengulangan. Selama hasil evaluasi ekspresi ini memberikan nilai bukan nol maka statement dikerjakan berulang kali.

#### c. Instruksi Do While

Instruksi do while ialah instruksi untuk melakukan proses pengulangan yang pemeriksaan syarat pengulangannya dilakukan pada akhir proses. Instruksi do while umumnya digunakan untuk melakukan proses pengulangan yang belum diketahui frekuensi pengulangannya tetapi pasti dikerjakan minimal satu kali.

**do statement while (expression);**

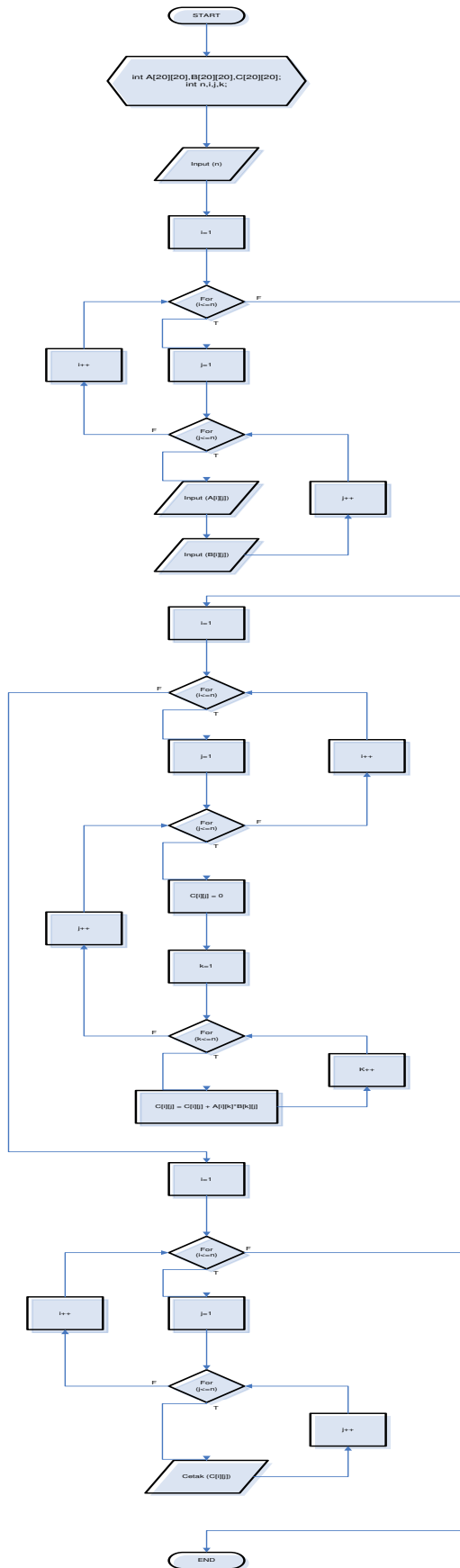
Statement berupa sebuah instruksi atau beberapa instruksi yang dilingkup oleh {}. Expression berupa ekspresi Boolean dan berfungsi sebagai control pengulangan. Selama hasil evaluasi expression ini memberikan nilai bukan nol maka statement dikerjakan berulang kali.

### IV. PEMBAHASAN DAN HASIL

Penerapan metode-metode tersebut adalah dengan menjawab soal-soal dari materi operasi perkalian matriks seperti tertulis di bawah ini:

$$A \begin{vmatrix} 1 & 2 \\ 3 & 1 \end{vmatrix} \times B \begin{vmatrix} 2 & 2 \\ 1 & 3 \end{vmatrix} = C \begin{vmatrix} 4 & 8 \\ 7 & 9 \end{vmatrix}$$

#### A. Flowchart



Gambar 1. Flowchart Perkalian Matriks

**B. Pseudocode**

```

A[2,2], B[2,2], C[2,2] = 0;
n, i, j, k = 0;
write ('Masukkan Ukuran Matriks: ');
read n;
j=1;
for i <= n;
    { j=1;
      for j <= n;
        { write ('Matriks A: '); read A[i,j];
          write ('Matriks B: '); read B[i,j];
        }
      }
i=1;
for i <= n;
    { j=1;
      for j <= n;
        { C[i,j] = 0;
          k=1;
          for k <= n;
            { C[i,j] = C[i,j] + A[i,k] *
              B[k,j];
            }
          }
        }
    }
write ('Hasil Matriks: ');
i=1;
for i <= n;
    { j=1;
      for j <= n;
        { write ('Matriks C: ');
          write C[i,j];
        }
      }
    }

```

**C. Desk Check Table**

Hasil analisa penulis pada flowchart operasi perkalian matriks dengan menggunakan desk check table, sebagai berikut:

1. Tahap 1: Desk Check Table Proses Pengisian Pengisian Elemen Matriks

Tabel 1. Desk Check Table Proses Pengisian Pengisian Elemen Matriks

Input (n)	i=1	For (i<=n)	HB	j=1	For (j<=n)	HB	Input (A[i][j])	Input (B[i][j])	j++	i++
2	i=1	1<=2	T	j=1	1<=2	T	A[1][1]=1	B[1][1]=2	1++	
					2<=2	T	A[1][2]=2	B[1][2]=2	2++	
					3<=2	F	-	-	-	1++
		2<=2	T	j=1	1<=2	T	A[2][1]=3	B[2][1]=1	1++	
					2<=2	T	A[2][2]=1	B[2][2]=3	2++	
					3<=2	F	-	-	-	2++
					3<=2	F	-	-	-	-

Sumber: Hasil Penelitian (2014)

2. Tahap 2: Desk Check Table Proses Perkalian Elemen Matriks

Sumber: Hasil Penelitian (2014)

3. Tahap 3: Desk Check Tabel Proses Pencetakan Hasil Perkalian Matriks

**Tabel 2.** Desk Check Table Proses Perkalian Elemen Matriks

i=1	For (i<=n)	HB	j=1	For (j<=m)	HB	C[i][j]=0	k=1	For (k<=n)	HB	C[i][j]=C[i][j]+A[i][k]*B[k][j]	K++	j++	i++
i=1	1<=2	T	j=1	1<=2	T	C[1][1]=0	k=1	1<=2	T	C[1][1]=0+1*2=2	1++		
								2<=2	T	C[1][1]=2+1*2=4	2++		
								3<=2	F				1++
			2<=2	T	C[1][2]=0	k=1	1<=2	T	C[1][2]=0+1*2=2	1++			
								2<=2	T	C[1][2]=2+2*3=8	2++		
								3<=2	F				2++
			β<=2	-	-	-	-	-	-	-	-	-	1++
2<=2	T	j=1	1<=2	T	C[2][1]=0	k=1	1<=2	T	C[2][1]=0+3*2=6	1++			
								2<=2	T	C[2][1]=6+1*1=7	2++		
								3<=2	F				1++
			2<=2	T	C[2][2]=0	k=1	1<=2	T	C[2][2]=0+3*2=6	1++			
								2<=2	T	C[2][2]=6+1*3=9	2++		
								3<=2	F				2++
3<=2	F	-	-	-	-	-	-	-	-	-	-	-	-

**Tabel 3.** Desk Check Tabel Proses Pencetakan Hasil Perkalian Matriks

input (n)	i=1	For (i<=n)	HB	j=1	For (j<=n)	HB	input (A[i][j])	input (B[i][j])	j++	i++	For (i<=n)	HB	j=1	For (j<=n)	HB	C[i][j]=0	k=1	For (k<=n)	HB	C[i][j]=C[i][j]+A[i][k]*B[k][j]	K++	j++	i++	For (i<=n)	HB	j=1	For (j<=n)	HB	Cetak (C[i][j])	j++	i++
2	i=1	1<=2	T	j=1	1<=2	T	A[1][1]=1	B[1][1]=2	1++											C[1][1]=0+1*2=2	1++										
				2<=2	T	A[1][2]=2	B[1][2]=2	2++													C[1][1]=2+2*1=4	2++									
				3<=2	F	-	-	-	1++												-	-	1++								
	2<=2	T	j=1	1<=2	T	A[2][1]=3	B[2][1]=1	1++													C[2][1]=0+3*2=6	1++									
				2<=2	T	A[2][2]=1	B[2][2]=3	2++													C[2][1]=6+1*1=7	2++									
				3<=2	F	-	-	-	2++												-	-	1++								
	3<=2	F	-	-	-	-	-	-	-												-	-	2++								

sebanyak 2 kali. Perintah for ( $i \leq n$ ) adalah perintah yang mengontrol pengulangan sebanyak 2 kali. Intruksi  $i \leq n$ , dapat berarti  $i \leq 2$ , artinya pengulangan akan dilakukan sebanyak kurang dari 2, dan dimulai dari 1, untuk jumlah baris matriks. Kolom HB, maksudnya adalah Hasil Boolean, yaitu untuk mengetahui kondisi True atau False, yang selanjutnya dapat menentukan, mengulang atau tidak mengulang. Pada kolom HB1 baris 1, tertulis T, ini artinya kondisi True (Benar), berarti dapat dilanjutkan pengulangan. Intruksi  $j=1$ , artinya variable  $j$  diberi nilai awal 1. Sama halnya dengan variable  $i$ , tetapi dalam hal ini bedanya adalah bahwa variable  $j$  untuk menentukan jumlah kolom matriks.

Input ( $A[i][j]$ ), intruksi untuk menampung nilai matrik, berurutan, mulai dari baris 1 kolom 1, berulang sampai baris 2 kolom 2, pada matriks A. Pada kasus ini, secara berurutan, isi matriks A baris 1 kolom 1 adalah nilai 1, isi matriks B baris 1 kolom 1 adalah nilai 2. Instruksi  $j++$  berfungsi untuk menambah nilai pada variabel  $j$ , berarti  $1++$  artinya, variabel  $j$  bertambah 1, sebelumnya bernilai 1 berarti sekarang menjadi 2 pada variable  $j$ . Lihat Gambar 1, selanjutnya variable  $j$  akan dikontrol oleh instruksi for ( $j \leq n$ ), jika hasilnya True maka dapat masuk ke badan looping atau melakukan pengulangan. Pada kasus ini berarti menguji apakah  $2 \leq 2$ , hasilnya adalah T, lihat HB2 kolom 2. Lihat Tabel 1, variabel  $i$  masih 1, sementara variabel  $j$  sudah menjadi 2. Pengisian matrik dapat dilanjutkan, matriks A baris 1 kolom 2 diisi nilai 2 dan matriks B baris 1 kolom 2 adalah nilai 2.

Intruksi  $2++$  pada baris ketiga Tabel 1 mengakibatkan variabel  $j$  bertambah 1, maka sekarang variabel  $j$  menjadi 3, dengan demikian instruksi for ( $j \leq 2$ ) mengakibatkan bernilai False (Lihat HB2 baris 3 Tabel 1), mengakibatkan keluar dari badan looping  $j$  masuk ke intruksi  $i++$ . Instruksi  $i++$  berarti nilai  $i$  bertambah 1, berarti sekarang nilai variable  $i$  adalah 2.

2. Tahap 2: Deskripsi Proses Perkalian Elemen Matriks
3. Tahap 3: Deskripsi Proses Pencetakan Hasil Perkalian Matriks

Kesimpulan penulis dari kasus di atas peneliti dapat membuktikan dengan lebih detail, jelas, dan bertahap pembuktian dari masing-masing variabel dari elemen matriks dari mulai proses pengisian elemen matriks, proses perkalian elemen matriks, sampai pada proses dari pencetakan hasil perkalian tersebut yang dapat dilihat baik pada

tabel pertahap atau pun pada tabel keseluruhan dari operasi perkalian matriks.

Saran penulis kepada pembaca artikel ini, mencari metode yang lebih sederhana untuk menganalisa suatu kebenaran suatu flowchart dengan menggunakan desk check table ini atau metode lainnya yang jauh lebih efisien.

## DAFTAR PUSTAKA

- Munir, Rinaldi. 2011. Algoritma & Pemrograman. Dalam Bahasa Pascal dan C. Bandung: Penerbit Informatika Bandung.
- Ngoen, Thompson Susabda. 2009. ALGORITMA DAN STRUKTUR DATA Bahasa C. Jakarta: Penerbit Mitra Wacana Media. Edisi Pertama.
- Robertson, Lesley Anne. 2004. *Simple Program Design. A Step-by-Step Approach. Fourth Edition*. Hongkong: *Course Technology*.
- Suryadi, H.S., dkk. 1990. Teori Dan Soal Pendahuluan Aljabar Linier. Serial Matematika. Jakarta: Ghalia Indonesia.
- Sutanta, Edhy. 2004. Algoritma Teknik Penyelesaian Permasalahan Untuk Komputasi. Graha Ilmu. Yogyakarta.
- Yahya, Yusuf, etc. 2005. Matematika Dasar untuk Perguruan Tinggi. Bogor : Ghalia Indonesia. Cetakan Keduabelas.
- <http://www.akmi-baturaja.ac.id/wp-content/uploads/2012/07/Logika-dan-Algoritma.pdf>

## Riwayat Penulis:



Rini Nuraini, S.T, M.Kom. Tahun 1998 lulus pendidikan Strata Satu (S1) dengan program studi Teknik Komputer (S.T). Tahun 2010 lulus pendidikan Strata Dua (S2) dengan program studi Ilmu Komputer (M.Kom).

Sejak tahun 1999 menjadi Dosen hingga saat ini, pada program studi Teknik Informatika, Sistem Informasi, Manajemen Informatika, Teknik komputer, dan Akuntansi Komputer, untuk matakuliah yang berkaitan dengan komputasi, baik *software* ataupun *hardware*, diantaranya. TMT 1 Agustus 2004 Jabatan Fungsional Akademik dengan pangkat Lektor. Tahun 2009 sebagai penerima penelitian dosen kategori PDM (Peneitian Dosen Muda) Hibah DIKTI. Tahun 2011 mendapat Sertifikasi Dosen.