Pengimplementasian Rekursif dalam Menentukan Determinan dan Hasil Perkalian Matriks

Kristianto Karim - 13514075
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514075@std.stei.itb.ac.id

***Abstrak*— Di dalam tulisan ini akan dibahas mengenai penerapan rekursif dalam perhitungan determinan dan perkalian antara dua matriks. Rekursif adalah seuatu yang berulang dan di dalam pengulangannya mengandung dirinya sendiri. Dalam tulisan ini akan dibahas dua penerapan algoritma rekursif yaitu dalam mencari determinan matriks dan matriks hasil perkalian dua matriks serta implementasinya.**

***Kata kunci*—determinan matriks, implementasi rekursif, matriks, perkalian matriks, rekursif.**

I. Latar Belakang

Pada kehidupan sehari – hari banyak hal yang terjadi berulang – ulang. Tetapi proses tersebut pada suatu saat akan berhenti. Bahasa matematika dari proses berulang tersebut adalah rekursif. Jadi segala sesuatu dalam kehidupan ini, berlangsung, berulang terus – menerus sampai kegiatan tersebut mencapai suatu saat yang kita sebut sebagai basis. Sebagai contoh, kehidupan kuliah kita, kita akan terus kuliah terus – menerus, sampai suatu saat kita mencapai suatu saat dimana kita tidak kuliah lagi yaitu lulus (atau mungkin DO).

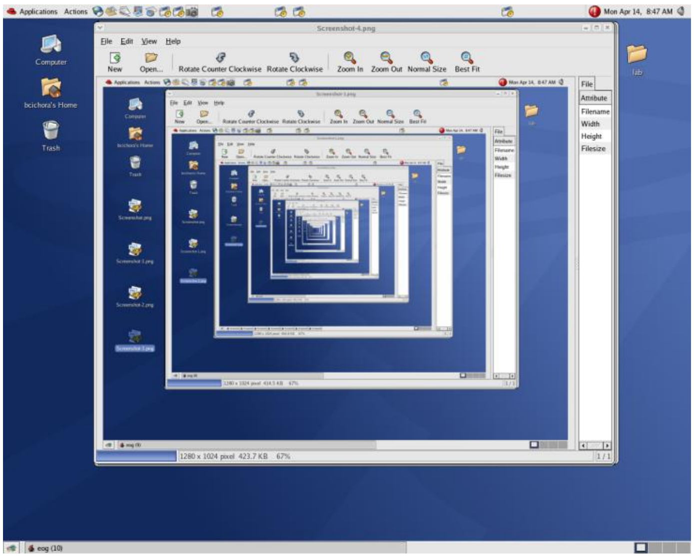
Hal yang berulang itu kemudian diteliti oleh para ahli dan digeneralisasi. Kemudian hal tersebut disebut rekursif, dan mereka menerapkan rekursif dalam penyelesaian masalah dibidang apapun. Salah satu bidang yang digunakan yaitu programming dan matematika.

Menghitung determinan dan hasil dari perkalian matriks akanlah sangat merepotkan jika kita menghitung secara manual terus menerus. Oleh karena itu, kita mengimplementasikannya ke dalam sebuah program secara rekursif untuk memudahkan penyelesaian persoalan matriks.

II. Dasar Teori

A. Rekursif

Rekursif adalah sesuatu yang memiliki sifat pengulangan yang di dalam pengulangan itu mengandung definisi dari dirinya sendiri. Contohnya : membuka kotak dialog pada windows yang membuka kotak dialog



Sumber : https://www.google.co.id/imgres?imgurl=http://logos.cs.uic.edu/examples%252520and%252520notes/notes/Java/Recursion/RecursionScreenshot.png&imgrefurl=http://logos.cs.uic.edu/examples%2520and%2520notes/notes/Java/Recursion/Recursion.htm&h=1024&w=1280&tbnid=kdYXa0nidGe6OM:&docid=FDkoJS5Yvp6t2M&ei=3IZmVtfXAtHmuQTX146QBw&tbm=isch&ved=0ahUKEwjXtaz63svJAhVRc44KHderA3IQMwg1KAYwBg

atau melukis tangan orang yang sedang melukis tangan orang yang sedang melukis

Sumber :

https://www.google.co.id/search?q=recursive+painting&biw=1517&bih=665&tbm=isch&imgil=elnpVMBrIGJyBM%253A%253BXnYN2Ep1obS5bM%253Bhttp%25253A%25252F%25252Fwww.superbwallpapers.com%25252Fdigital-art%25252Frecursive-painting-9137%25252F&source=iu&pf=m&fir=elnpVMBrIGJyBM%253A%252CXnYN2Ep1obS5bM%252C_&usg=__1_4qPEDjCRPsA6_JXj1S9tr80SU%3D

Kondisi dimana rekursif berlangsung disebut rekurens. Dalam suatu rekurens akan selalu ada

1. Basis : sebagai titik di mana rekursif itu berakhir(Jika tidak ada maka tidak akan berhenti).
2. Rekurens : sebagai bagian yang berulang terus – menerus dengan memanggil dirinya sendiri disertai inkremen atau dekremen untuk mencapai basisnya.

Contoh rekurens dalam matematika :

$$f(x) = f(x-1) + f(x-2), x \geq 2$$
$$f(0) = 1$$
$$f(1) = 2$$

Misalnya ingin dicari nilai dari $f(5)$ maka fungsi akan berekursif sampai mencapai $f(0)$ dan $f(1)$ seperti :

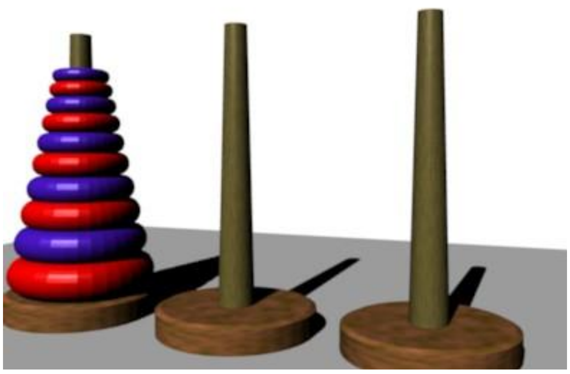
$$f(5) = f(4) + f(3)$$
$$f(5) = f(3) + f(2) + f(2) + f(1)$$
$$f(5) = f(2) + f(1) + f(1) + f(0) + f(1) + f(0) + f(1)$$
$$f(5) = f(1) + f(0) + f(1) + f(1) + f(0) + f(1) + f(0) + f(1)$$
$$f(5) = 5f(1) + 3f(0)$$

Sehingga

$$f(5) = 2 + 1 + 2 + 2 + 1 + 2 + 1 + 2 = 13$$

Contoh lain rekursif dalam matematika yaitu :

1. Menara Hanoi

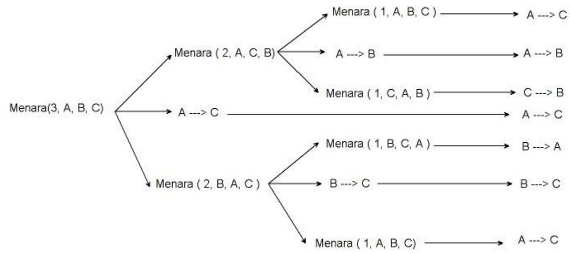


Sumber : https://www.google.co.id/imgres?imgurl=https://risdatinambunan.files.wordpress.com/2012/06/10_ring_hanoi.jpg&imgrefurl=https://risdatinambunan.wordpress.com/2012/06/22/menara-hanoi/&h=300&w=400&tbnid=FVeJjCs_TAP65M:&docid=rHFlOtci1uLJiM&ei=FwtnVvHAHoOduQSqr5WYAw&tbm=isch&ved=0ahUKEwixqP2H3czJAhWDTo4KHapXBTMQMwgmKAswCw

Menara Hanoi merupakan teka – teki matematika dengan peraturan

1. Hanya satu cakram yang boleh dipindahkan dalam sekali perpindahan
2. Pengambilan cakram harus dari paling atas
3. Cakram yang lebih kecil harus diatas cakram yang lebih besar

Berikut algoritma rekursif untuk menara hanoi



Sumber : https://rusdyana.files.wordpress.com/2009/11/asd1.jpg

Jika dituliskan maka algoritmanya adalah

```
If ( N == 1) then
  Pindahkan cakram dari tiang asal ke tujuan
Else
  Menara(N-1,asal,tujuan,bantu)
  Pindahkan cakram ke n dari asal ke tujuan
  Menara(N-1,bantu,asal,tujuan)
```

2. Faktorial

   Rekursif dalam factorial yaitu

$$n! = n(n-1)(n-2)..1$$

3. Rasio emas

$$\phi = 1 + \frac{1}{\phi}$$

B. *Matriks*

Matriks merupakan kumpulan bilangan, simbol atau ekspresi yang disusun berdasarkan baris dan kolom.

Isi dari suatu matriks disebut elemen matriks.

Contoh matriks

$$\begin{bmatrix} 1 & 3 & 0 & 2 \\ 2 & 4 & 6 & 7 \\ 5 & 9 & 1 & 8 \end{bmatrix}$$

Jenis – jenis matriks :

1. Matriks Persegi

   Matriks yang jumlah baris dan kolomnya sama.

2. Matriks Diagonal

   Matriks yang semua elemennya nol kecuali elemen diagonalnya

   Matriks diagonal :

$$\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}$$

Dimana a, b, c dapat diisi apa saja kecuali nol.

3. Matriks Segitiga
   Matriks segitiga terdiri dari 2 yaitu :
   1. Matriks segitiga atas

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

   2. Matriks segitiga bawah

$$\begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

4. Matriks Simetris
   Misalkan ada matriks A, jika A = A^T maka A adalah matriks simetris.
   Contohnya :

$$\begin{bmatrix} a & b & e \\ b & d & h \\ e & h & u \end{bmatrix}$$

Operasi dasar pada matriks mencakup :

1. Penjumlahan matriks (baris dan kolom kedua matriks harus sama)

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & h \\ f & g \end{bmatrix} = \begin{bmatrix} a+e & b+h \\ c+f & d+g \end{bmatrix}$$

2. Pengurangan matriks (baris dan kolom kedua matriks harus sama)

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} e & h \\ f & g \end{bmatrix} = \begin{bmatrix} a-e & b-h \\ c-f & d-g \end{bmatrix}$$

3. Perkalian dengan skalar

$$k\,.\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ka & kb \\ kc & kd \end{bmatrix}$$

4. Transpose

$$\begin{bmatrix} a & f & g \\ b & e & h \\ c & d & i \end{bmatrix}^T = \begin{bmatrix} a & b & c \\ f & e & d \\ g & h & i \end{bmatrix}$$

C. Perkalian Matriks dengan Matriks

Matriks bisa dikalikan dengan matriks dengan syarat, jumlah kolom matriks I harus sama dengan jumlah baris kolom kedua.

Perkalian antara dua matriks dilakukan dengan peraturan sebagai berikut :

Misalkan ada

$$M_{ixm}\,.\,M_{mxj} = M_{ixj}$$

Maka elemen matriks hasil perkalian didapatkan dengan :

$$H_{ij} = \sum_{k=1}^{m} a_{ik}\,.\,b_{kj}$$

Dengan m adalah jumlah kolom matriks I atau jumlah baris matriks II, i adalah jumlah baris matriks I dan j adalah jumlah kolom matriks II.

Jadi ukuran matriks hasil perkalian yaitu baris dari matriks I dan kolom dari matriks II.

Contohnya :

$$\begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} . \begin{bmatrix} 5 & 2 & 9 \\ 6 & 7 & 8 \end{bmatrix}$$

$$= \begin{bmatrix} (1*5+2*6) & (1*2+2*7) & (1*9+2*8) \\ (4*5+3*6) & (4*2+3*7) & (4*9+3*8) \end{bmatrix}$$

D. Minor dan Kofaktor

Minor adalah matriks yang elemennya merupakan determinan dari matriks selain baris dan kolom minor itu sendiri.

Misalkan ada matriks A

$$A = \begin{bmatrix} a & d & i \\ b & e & h \\ c & f & g \end{bmatrix}$$

Matriks minornya adalah

$$M_A = \begin{bmatrix} \det\begin{pmatrix} e & h \\ f & g \end{pmatrix} & \det\begin{pmatrix} b & h \\ c & g \end{pmatrix} & \det\begin{pmatrix} b & e \\ c & f \end{pmatrix} \\ \det\begin{pmatrix} d & i \\ f & g \end{pmatrix} & \det\begin{pmatrix} a & i \\ c & g \end{pmatrix} & \det\begin{pmatrix} a & d \\ c & f \end{pmatrix} \\ \det\begin{pmatrix} d & i \\ e & h \end{pmatrix} & \det\begin{pmatrix} a & i \\ b & h \end{pmatrix} & \det\begin{pmatrix} a & d \\ b & e \end{pmatrix} \end{bmatrix}$$

Sedangkan kofaktor adalah matriks yang elemennya sama dengan minor tetapi ada perbedaan tanda pada beberapa elemennya. Elemen dari kofaktor dapat dihitung dengan menggunakan rumus

$$C_{ij} = (-1)^{(i+j)} M_{ij}$$

E. Determinan Matriks

Determinan adalah sesuatu yang berfungsi untuk menghubungkan bilangan riil dengan matriks persegi.

Misalkan matriks $A_{2x2} = \begin{bmatrix} a & d \\ b & c \end{bmatrix}$ maka

$$detA = ac - bd$$

Untuk matriks dengan jumlah baris dan kolom diatas 2, determinannya dapat dicari dengan cara :

Misal ada matriks A

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1j} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2j} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3j} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ a_{i1} & a_{i2} & a_{i3} & \dots & a_{ij} \end{bmatrix}$$

Dengan minor dan kofaktor

$$M_A = \begin{bmatrix} M_{11} & M_{12} & M_{13} & \dots & M_{1j} \\ M_{21} & M_{22} & M_{23} & \dots & M_{2j} \\ M_{31} & M_{32} & M_{33} & \dots & M_{3j} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ M_{i1} & M_{i2} & M_{i3} & \dots & M_{ij} \end{bmatrix}$$

$$C_A = \begin{bmatrix} M_{11} & -M_{12} & \dots & (-1)^{1+j} M_{1j} \\ -M_{21} & M_{22} & \dots & (-1)^{2+j} M_{2j} \\ M_{31} & -M_{32} & \dots & (-1)^{3+j} M_{3j} \\ \vdots & \vdots & \dots & \vdots \\ (-1)^{i+1} M_{i1} & (-1)^{i+2} M_{i2} & \dots & (-1)^{i+j} M_{ij} \end{bmatrix}$$

Maka

$$\det(A) = \sum_{k=1}^{j} a_{1k} C_{1k}$$

Determinan juga dapat dicari dengan cara lain yaitu melakukan operasi barisan elementer sampai didapatkan suatu matriks segitiga atas kemudian determinannya di hitung dengan mengalikan seluruh elemen di diagonalnya kemudian dikali dengan $(-1)^i$ dimana i adalah jumlah baris yang ditukar pada saat operasi barisan elementer.

Misalkan ada matriks A

$$A = \begin{bmatrix} a & h & i & p \\ b & g & j & o \\ c & f & k & n \\ d & e & l & m \end{bmatrix}$$

Setelah di OBE

$$A' = \begin{bmatrix} a' & h' & i' & p' \\ 0 & g' & j' & o' \\ 0 & 0 & k' & n' \\ 0 & 0 & 0 & m' \end{bmatrix}$$

Maka

$$\det(A') = a' * g' * k' * m'$$

dan

$$\det(A) = (-1)^i \det(A')$$

A. III. IMPLEMENTASI

A. Perkalian Matriks

Seperti yang kita lihat di Bab sebelumnya yaitu bahwa perkalian adalah suatu proses penambahan yang berulang – ulang dan berhenti ketika semua elemen matriks hasil perkalian berisi.

Perkalian matriks memanfaatkan 3 penunjuk yang memiliki fungsi masing masing, pada program yang diimplementasikan, digunakan 3 penunjuk (kita misalkan i, j, k). Penunjuk i berfungsi untuk pergantian baris agar baris dari matriks hasil perkalian dapat diproses semuanya, penunjuk j fungsinya hampir sama dengan i tetapi yang ditunjuk oleh j adalah kolomnya, dan yang terakhir yaitu penunjuk k yang berfungsi dalam proses perkalian matriksnya, penunjuk k berinkremen secara kolom pada matriks 1 dan berinkremen secara baris pada matriks 2, dan akan selalu di set mengikuti baris jika terjadi pergantian baris oleh penunjuk i atau di set mengikuti kolom jika terjadi pergantian kolom oleh penunjuk j.

Berikut implementasinya dalam bahasa C :

```
void Perkalian (int B1, int K1, int M1[][],
    int B2, int K2, int M1[][], int *c[][])
{
    static int i = 0, j = 0, k = 0;

    if (i >= B1)
    {
        return;
    }
    else if (i < B1)
    {
        if (j < K2)
        {
            if (k < K1)
            {
                c[i][j] += M1[i][k] * M2[k][j];
                k++;
                Perkalian(B1, K1, M1, B2, K2, M2, c);
            }
            k = 0;
            j++;
            Perkalian(B1, K1, M1, B2, K2, M2, c);
        }
        j = 0;
        i++;
        Perkalian(B1, K1, M1, B2, K2, M2, c);
    }
}
```

Sumber :

https://www.google.co.id/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&cad=rja&uact=8&ved=0ahUKEwjVps2i2MvJAhVOVY4KHVO_BZYQFgg9MAU&url=http%3A%2F%2Fwww.sanfoundry.com%2Fc-program-matrix-multiplication-using-recursion%2F&usg=AFQjCNHY6y3-ptsT2I6E4gsy4hK29OOILw&sig2=5MikxWGV2DgVI2i7mWPfbg

Prosedur rekursif dari implementasi perkalian matriks di atas menggunakan basis yaitu saat penunjuk i nilainya lebih besar sama dengan baris matriks 1 yang artinya adalah semua elemen telah dihitung hasil perkaliannya (indeks baris terakhir adalah (B1-1) karena array pada bahasa pemograman C dimulai dari indeks 0). Sedangkan bagian rekurensnya secara berulang memproses matriks c (matriks hasil perkalian).

B. Determinan Matriks

Di bab sebelumnya di bagian determinan telah dijelaskan bahwa untuk mencari determinan dari suatu matriks yang ordonya lebih besar dari 2x2 diperlukan determinan dari sub – matriksnya (minor). Untuk matriks yang lebih besar ordonya maka akan sangat merepotkan untuk melakukan perhitungan manualnya, program.

Implementasi rekursif dalam menghitung determinan suatu matriks persegi yaitu dengan membuat suatu fungsi rekursif yang basisnya adalah determinan matriks berordo 2x2 dan bagian rekursennya yaitu menentukan nilai minornya yang akan direkursif sampai sub – matriks 2x2).

Kemudian dilanjutkan dikali dengan $(-1)^{(i+j)}$ untuk mendapat kofaktornya dan diakumulasi dengan jumlah sebelumnya.

Berikut implementasinya dalam bahasa C :

```
double Determinant(double **a,int n)
{
   int i,j,j1,j2;
   double det = 0;
   double **m = NULL;

   if (n < 1) { /* Error */

   } else if (n == 1) { /* Shouldn't get used */
      det = a[0][0];
   } else if (n == 2) {
      det = a[0][0] * a[1][1] - a[1][0] * a[0][1];
   } else {
      det = 0;
      for (j1=0;j1<n;j1++) {
         m = malloc((n-1)*sizeof(double *));
         for (i=0;i<n-1;i++)
            m[i] = malloc((n-1)*sizeof(double));
         for (i=1;i<n;i++) {
            j2 = 0;
            for (j=0;j<n;j++) {
               if (j == j1)
                  continue;
               m[i-1][j2] = a[i][j];
               j2++;
            }
         }
         det += pow(-1.0,1.0+j1+1.0) * a[0][j1] * Determinant(m,n-1);
         for (i=0;i<n-1;i++)
            free(m[i]);
         free(m);
      }
   }
   return(det);
}
```

Sumber :

https://www.google.co.id/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwiI29mBiszJAhVBT44KHRSqA7sQFggmMAI&url=https%3A%2F%2Fgist.github.com%2F4037974&usg=AFQjCNEnrL0NM0pUVojAk21dMwTYyFU8HA&sig2=PZ6nxeLzk4uj0obCP9GnFQ&bvm=bv.109332125,d.c2E

V. KESIMPULAN

Rekurens sangat banyak aplikasinya dalam kehidupan untuk menyelesaikan permasalahan salah satunya permasalahan matematika, dan akan lebih mudah dan efisien lagi jika dapat diimplementasikan ke dalam program.

DAFTAR PUSTAKA

Munir,rinaldi."Diktat Kuliah IF2120 Matematika Diskrit", edisi keempat,Program Studi Teknik Informatika STEI ITB,2006.

http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2013-2014/Makalah2013/MakalahIF2120-2013-051.pdf
Waktu akses : 08 Desember 2015, 00.00 WIB

http://rumus-matematika.com/materi-matriks-lengkap-dan-contohnya/
Waktu akses : 08 Desember 2015, 02.30 WIB

http://paulbourke.net/miscellaneous/determinant/
Waktu akses : 08 Desember 2015, 03.00 WIB

Abstract Data Type Matriks Algoritma Struktur Data

http://www.sanfoundry.com/c-program-matrix-multiplication-using-recursion/
Waktu akses : 08 Desember 2015, 23.00 WIB

http://www.sourcecodesworld.com/source/show.asp?ScriptID=1054
Waktu akses : 09 Desember 2015, 00.30 WIB

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 8 Desember 2015

Kristianto Karim – 13514075