

1 D4 - TEKKOM B

MEMORY DINAMIS



Nama	:	Septian Bagus Jumentoro
Kelas	:	1 – D4 Teknik Komputer B
NRP	:	3221600039
Dosen	:	Dr Bima Sena Bayu Dewantara S.ST, MT.
Mata Kuliah	:	Praktikum Pemrograman Dasar 2
Hari/Tgl. Praktikum	:	Rabu, 09 Maret 2022

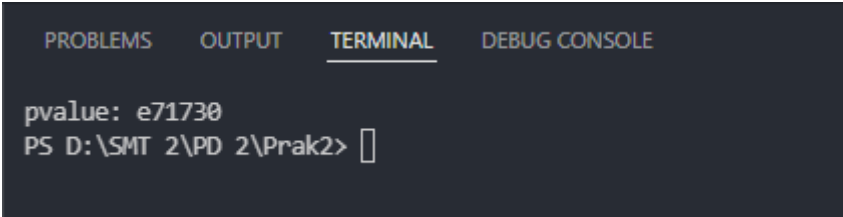


Percobaan 1

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    system("cls");
    double* pvalue=NULL;
    if(!(pvalue = new double)){
        printf("Error: out of memory");
        exit(1);
    }
    else{
        printf("pvalue: %x",pvalue);
        exit(1);
    }
}
```

Output



The screenshot shows a terminal window with tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. The TERMINAL tab is active, displaying the output of the program: "pvalue: e71730" followed by a prompt "PS D:\SMT 2\PD 2\Prak2>".

Analisa

Berdasarkan program tersebut pvalue diberikan alokasi memori. Namun ketika gagal maka yang tampil akan error, tetapi ketika berhasil alamat tersebut akan ditampilkan pada layar dengan memaminggil pvalue.

Percobaan 2

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>

using namespace std;
int main ()
{
    system("cls");
    double* pvalue = NULL; // Pointer initialized with null
```

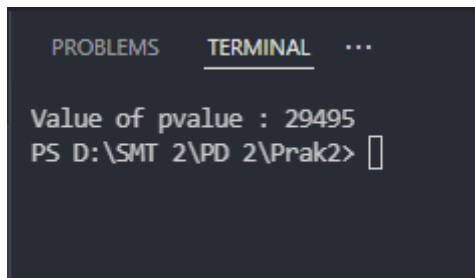
```

    pvalue = new double; // Request memory for the variable
    *pvalue = 29494.99; // Store value at allocated address
    cout << "Value of pvalue : " << *pvalue << endl;
    delete pvalue; // free up the memory.

    _sleep(500);
}

```

Output



```

PROBLEMS  TERMINAL  ...
Value of pvalue : 29495
PS D:\SMT 2\PD 2\Prak2>

```

Analisa

Berdasarkan program tersebut value of pvalue telah bernilai 29494.99, setelah itu ketika di cetak maka nilai tersebut dibulatkan naik menjadi 29495. Untuk pvalue yang telah dihapus akan bernilai 0.

Percobaan 3a

```

#include <stdio.h>
#include <stdlib.h>
#include <iostream>
using namespace std;

int main()
{
    system("cls");
    int i,j;
    int ROW = 5;
    int COL = 8;
    double **pvalue = new double* [ROW];
    // Allocate memory for rows
    // Now allocate memory for columns
    for(i=0;i<ROW;i++)
    {
        pvalue[i] = new double[COL];
    }
}

```

```

    for(i=0;i<ROW;i++)
    {
        for(j=0;j<COL;j++)
        {
            pvalue[i][j]=0.1; printf("%.2f\t",pvalue[i][j]);
        }
        printf("\n");
    }

    for(i=0;i<ROW;i++)
    {
        delete[] pvalue[i];
    }
    delete [] pvalue;

    _sleep(500);
}

```

Output

PROBLEMS	OUTPUT	TERMINAL	DEBUG CONSOLE
0.10	0.10	0.10	0.10
0.10	0.10	0.10	0.10
0.10	0.10	0.10	0.10
0.10	0.10	0.10	0.10
0.10	0.10	0.10	0.10

PS D:\SMT 2\PD 2\Prak2> █

Percobaan 3b

```

#include <stdio.h>
#include <stdlib.h>

int main(){
    system("cls");
    int i,j;
    int ROW=5,COL=8;
    double** pvalue = new double* [ROW];
    double* pvalue2 = new double [COL];

    for(i=0;i<ROW;i++){
        pvalue[i] = new double [COL];
    }
    for(i=0;i<ROW;i++){

```

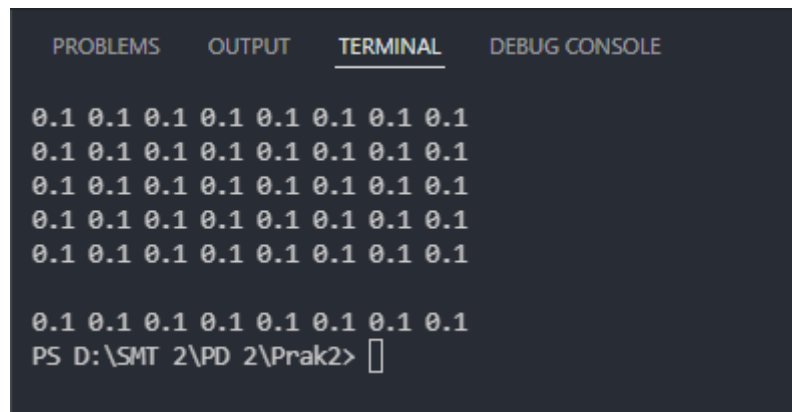
```

        for(j=0;j<COL;j++){
            pvalue[i][j]=0.1;
            printf("%g ",pvalue[i][j]);
        }
        puts("");
    }
    puts("");
    for(i=0;i<COL;i++){
        pvalue2[i]=0.1;
        printf("%g ",pvalue2[i]);
    }

    for(i=0;i<ROW;i++){
        delete[] pvalue[i];
    }
    delete[] pvalue;
    delete[] pvalue2;
}

```

Output



```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1

0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
PS D:\SMT 2\PD 2\Prak2> 

```

Analisa

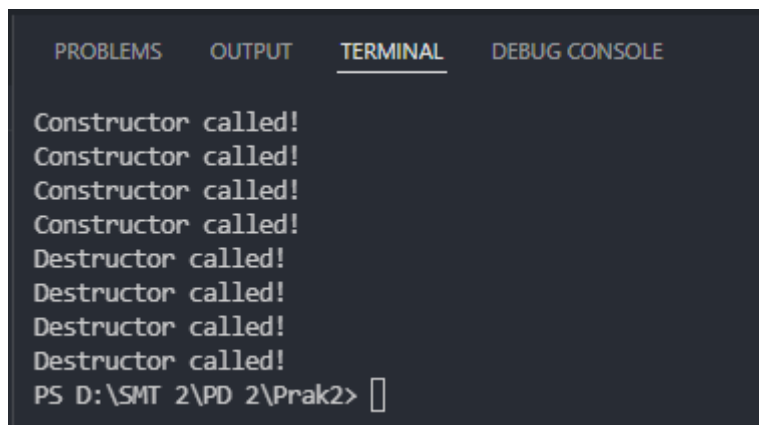
Berdasarkan program tersebut terdapat 2 bentuk array. Array 2 dimensi dan array 1 dimensi. Untuk array 2 dimensi nilai yang tampil pada layar yaitu 0.10 dan untuk array 1 dimensi nilai yang tampil di tiap barisnya 0.1. Kedua array tersebut memiliki source code yang hampir sama hanya memiliki sedikit perbedaan

Percobaan 4

```
#include <iostream>
#include <stdlib.h>
#include <stdio.h>
using namespace std;
class Box
{
public:
    Box()
    {
        cout << "Constructor called!" <<endl;
    }
    ~Box()
    {
        cout << "Destructor called!" <<endl;
    }
};

int main( )
{
    system("cls");
    Box* myBoxArray = new Box[4];
    delete [] myBoxArray;
    // Delete array
    _sleep(500);
}
```

Output



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Constructor called!
Constructor called!
Constructor called!
Constructor called!
Destructor called!
Destructor called!
Destructor called!
Destructor called!
PS D:\SMT 2\PD 2\Prak2> □
```

Analisa

Berdasarkan program tersebut class baru berfungsi untuk mencetak constructor ketika memori yang baru telah dibuat, setelah itu program mencetak destructor ketika memori dibersihkan