



**STMIK AMIK Bandung
Preprocessor libs and Functions**

Assignment

**Pramudya Arya Wicaksana
2242805**

**FAKULTAS INFORMATIKA
PROGRAM STUDI TEKNIK INFORMATIKA
BANDUNG**

CONTENTS

1	Pendahuluan	1
1.1	1. C application with math lib on it	1
1.2	2. Function prototype and definition	3
1.2.1	Function prototype	3
1.2.2	Function definition	3
1.3	3. Function under main()	4
1.4	4. Function call	4
1.5	5. Function without prototype	4
2	Chapter 2	5
2.1	Data	5
3	Chapter 3	6
3.1	Data	6

LIST OF FIGURES

CHAPTER 1

PENDAHULUAN

1.1 1. C application with math lib on it

Use cases : Membuat kotak berputar dengan bahasa C

```
#include <math.h>
#include <stdio.h>
#include <string.h>

double A, B, C;
float cubeWidth = 10;
int width = 160, height = 44;
float zBuffer[160 * 44];
char buffer[160 * 44];
int backgroundASCII = ' ';
int distanceFromCam = 60;
float K1 = 40;

float incrementSpeed = 0.6;

float x, y, z;
float ooz;
int xp, yp;
int idx;

float calculateX(int i, int j, int k) {
    return j * sin(A) * sin(B) * cos(C) - k * cos(A) * sin(B) * sin(C) +
           j * cos(A) * sin(C) + k * sin(A) * sin(C) + i * cos(B) * cos(C);
}

float calculateY(int i, int j, int k) {
    return j * cos(A) * cos(C) + k * sin(A) * cos(C) -
           j * sin(A) * sin(B) * sin(C) + k * cos(A) * sin(B) * sin(C) -
           i * cos(B) * sin(C);
}

float calculateZ(int i, int j, int k) {
```

```

    return k * cos(A) * cos(B) - j * sin(A) * cos(B) + i * sin(B);
}

void calculateSurface(float cubeX, float cubeY, float cubeZ, int ch) {
    x = calculateX(cubeX, cubeY, cubeZ);
    y = calculateY(cubeX, cubeY, cubeZ);
    z = calculateZ(cubeX, cubeY, cubeZ) + distanceFromCam;

    ooz = 1 / z;

    xp = (int)(width / 2 + K1 * ooz * x * 2);
    yp = (int)(height / 2 + K1 * ooz * y);

    idx = xp + yp * width;

    if (idx >= 0 && idx < width * height) {
        if (ooz > zBuffer[idx]) {
            zBuffer[idx] = ooz;
            buffer[idx] = ch;
        }
    }
}

int main() {
    printf("\x1b[2J");
    while (1) {
        memset(buffer, backgroundASCII, width * height);
        memset(zBuffer, 0, width * height * 4);
        for (float cubeX = -cubeWidth; cubeX < cubeWidth; cubeX += incrementSpeed) {
            for (float cubeY = -cubeWidth; cubeY < cubeWidth;
                 cubeY += incrementSpeed) {
                calculateSurface(cubeX, cubeY, -cubeWidth, '#');
                calculateSurface(cubeWidth, cubeY, cubeX, '$');
                calculateSurface(-cubeWidth, cubeY, -cubeX, '$');
            }
        }
        printf("\x1b[H");
        for (int k = 0; k < width * height; k++) {
            putchar(k % width ? buffer[k] : 10);
        }
    }
}

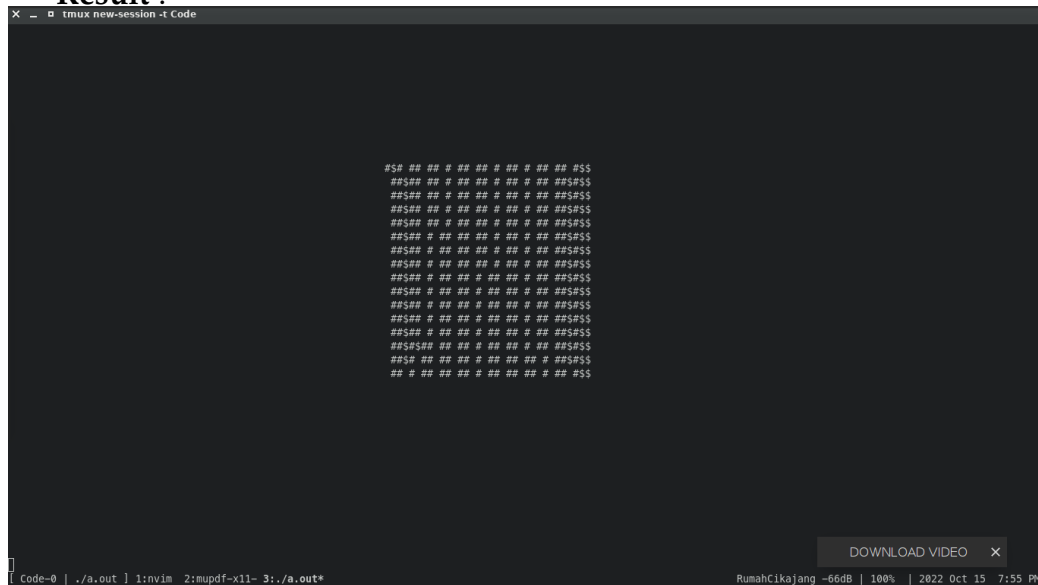
```

```

    A += 0.005;
    B += 0.005;
}
return 0;
}

```

Result :



1.2 2. Function prototype and definition

1.2.1 Function prototype

Function prototype adalah sebuah fungsi prototipe yang berguna untuk memberi tahu informasi kepada compiler (dalam kasus ini GCC Gnu Compiler Collection) tentang return type, parameter yang diterima Sebagai contoh berikut

```
float calculateX(int i, int j, int k);
```

Dalam codeblock diatas terdapat sebuah Function prototype berupa fungsi untuk menghitung X, dengan return type float dan menerima 3 jenit parameter int yaitu i, j dan k

1.2.2 Function definition

Function definition adalah fungsi yang mempunyai body atau statement yang harus dikerjakan oleh fungsi tersebut, biasanya diinisialisasi oleh Func-

tion prototype Sebagai contoh berikut

```
float calculateX(int i, int j, int k); // Function prototype

float calculateX(int i, int j, int k) { // Function definition
    return j * sin(A) * sin(B) * cos(C) - k * cos(A) * sin(B) * sin(C) +
           j * cos(A) * sin(C) + k * sin(A) * sin(C) + i * cos(B) * cos(C);
}
```

Dalam codeblock diatas adalah fungsi prototipe dan definisi, yang mana definisi mempunyai statement kalkulasi yang diperlukan oleh fungsi prototipe

1.3 3. Function under main()

Jawabannya, tergantung bahasa pemrograman, jika dalam bahasa GO kita bisa deklarasi fungsi dibawah main, dan jika untuk bahasa C kita tidak bisa mendeklarasikan fungsi setelah main, karena C menggunakan sistem **Syncronous** atau pembacaan dari atas kebawah

1.4 4. Function call

Jawabannya, bisa, dan cara pengaturan letaknya kembali lagi tergantung di bahasa masing masing, jika di bahasa GO bisa menyimpannya dimanapun kita mau, file lain, tempat lain, diatas atau dibawah main dan tanpa interface juga bisa, kecuali jika bahasa C, Bahasa C jika ingin memanggil sebuah function dari function, si function yang dipanggil harus berada di atas function yang memanggil, atau bisa dibawah function yang memanggil asal sudah diinialisasi prototypenya dibagian atas

1.5 5. Function without prototype

Jawabannya, bisa, dengan syarat fungsi yang dibuat harus disimpan dibagian atas agar terbaca oleh compiler

CHAPTER 2

CHAPTER 2

2.1 Data

CHAPTER 3

CHAPTER 3

3.1 Data