



Membuat Dokumen Kode Program

Visual Studio

Setiap program yang
dibuat, seharusnya
memiliki dokumentasi

Kenapa?

Alasan dibuatnya dokumentasi kode program

- ✓ Kode program yang dibuat belum tentu langsung sempurna, sesuai yang dibutuhkan pengguna. Kedepannya dapat diperlukan :
 - Pengembangan untuk memenuhi keperluan pengguna
 - Perbaiki program jika ternyata ditemukan kesalahan saat digunakan

Alasan dibuatnya dokumentasi kode program

- ✓ Untuk menginformasikan proses yang ada di dalam program
- ✓ Jika orang lain ingin mengembangkan aplikasi, maka dapat mempelajari dari dokumentasi yang ada



Apa itu dokumentasi kode program?

Dokumentasi yang harus ada dalam program, untuk memberikan penjelasan kepada setiap baris ataupun blok perintah dalam program.

Dokumentasi ini diperlukan bagi programmer untuk dapat menelusuri logika program.

Isi dari dokumentasi kode program :

- ❖ Langkah – langkah dalam program untuk menyelesaikan masalah
- ❖ Komentar dituliskan pada setiap awal modul atau fungsi
- ❖ Komentar ditulis pada setiap blok program
- ❖ Informasi tentang siapa yang membuat, kapan pertama kali dibuat, kapan terakhir diperbaiki.

Jenis Dokumentasi Kode Program

- Inline

Komentar terhadap perintah yang ditulis menggunakan syntax comment

- File Doc

File hasil export komentar yang ditulis pada documentation comment

Komentar



Untuk memberi penjelasan mengenai operasi apa saja yang dilakukan didalam sebuah potongan kode program.

Tipe Komentar

1. Komentar baris tunggal (//)
2. Komentar multi baris (/**/)
3. Komentar XML/Documentation Comment (///)

Komentar baris tunggal

Cara penulisan menggunakan dua garis miring (double slash) “//”.

```
int x = 5 + 2; //Menambahkan angka 5 dan 2  
  
//Menambahkan angka 9 dan 2  
int y = 9 + 2;
```

Komentar multi baris

Penulisan dimulai dengan tanda “/*” dan diakhiri dengan tanda “*/”.

```
/*  
    Ini adalah program Hello World di C#  
    Program ini akan menampilkan tulisan Hello World di jendela Console  
*/  
namespace HelloWorld  
{  
    0 references  
    class Program  
    {  
        0 references  
        static void Main(string[] args)  
        {  
            /* Menampilkan Tulisan Hello World */  
            Console.WriteLine("Hello World");  
        }  
    }  
}
```

Komentar XML / Documentation Comment

- Penulisan dimulai dengan 3 buah garis miring (triple slash) “///”.
- Digunakan untuk mendeskripsikan kode secara kategoris.
- Menggunakan tag XML dalam komentar.
- File dokumentasi XML akan terpisah (Di export dalam bentuk file .xml)

Melakukan identifikasi kode program

- Modul program diidentifikasi
 - Mengetahui bahasa pemrograman serta database yang digunakan
 - Mengetahui cara instalasi serta menjalankan komponen pendukung

Melakukan identifikasi kode program

- Parameter yang digunakan diidentifikasi

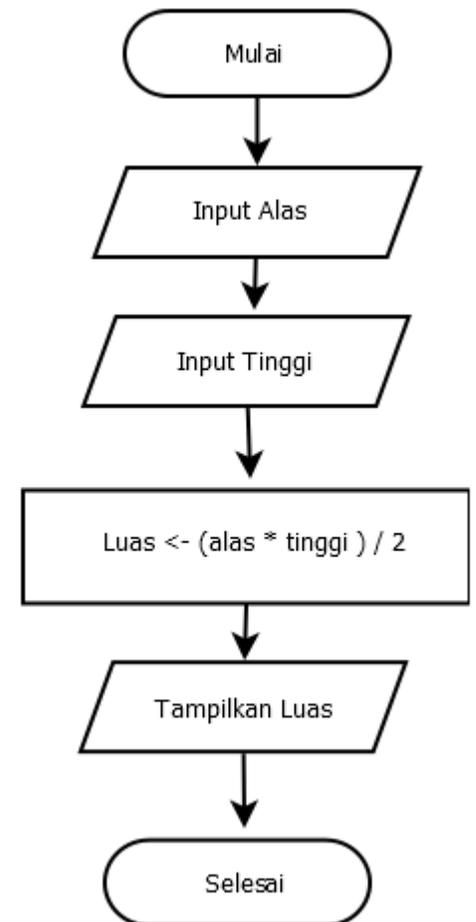
```
class Program
{
    0 references
    static int penjumlahan(int x, int y)
    {
        int hasil = x + y;
        return hasil;
    }

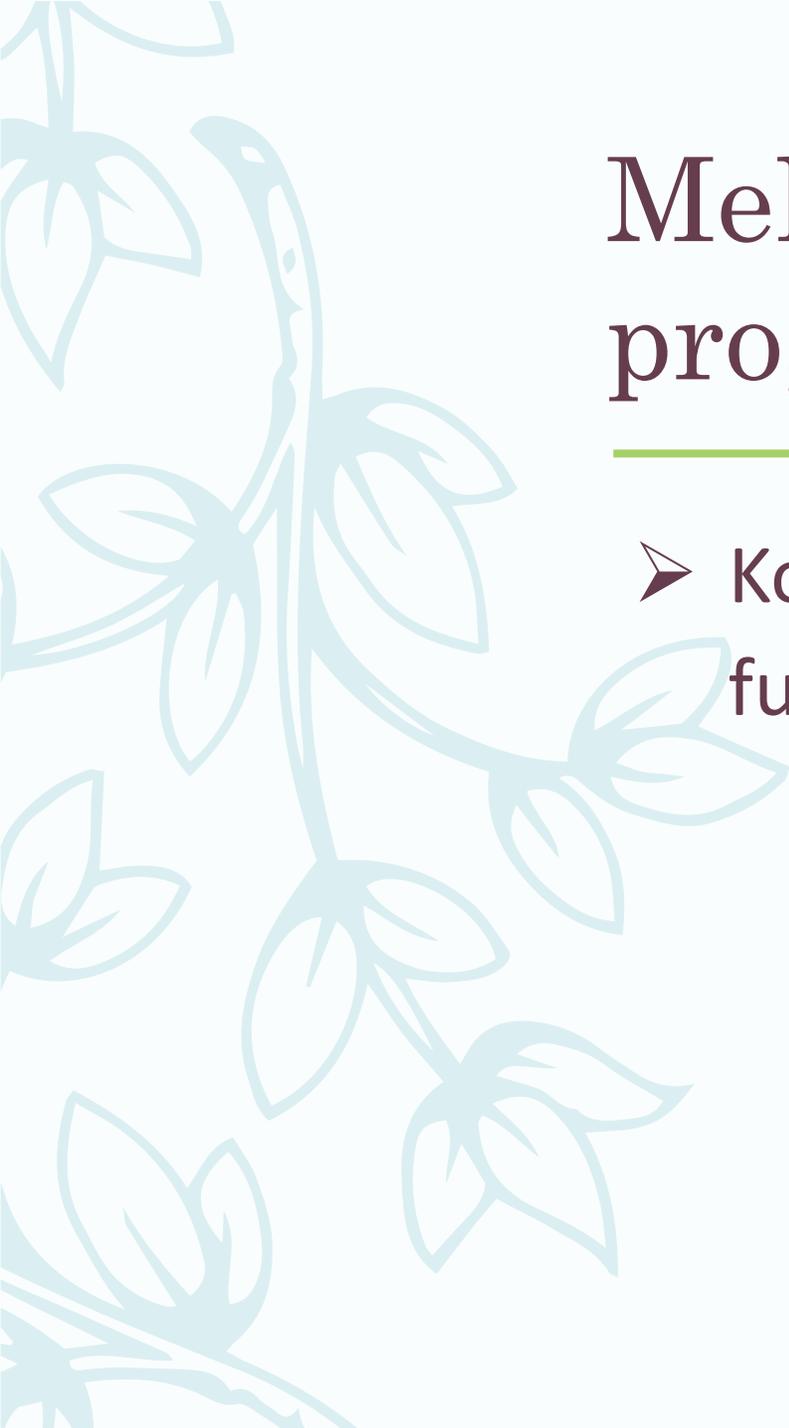
    0 references
    static void Main(string[] args)
    {
        Console.WriteLine(4 + 5);
        Console.ReadLine();
    }
}
```

➔ Parameter

Melakukan identifikasi kode program

- Algoritma dijelaskan cara kerjanya
 - a. Masukkan nilai alas (a) dan nilai tinggi segitiga (t)
 - b. Maka untuk menghitung luas digunakan rumus yang sudah ditentukan
 - c. Rumus untuk menghitung Luas Segitiga
$$L = \frac{1}{2} * a * t$$
 - d. Nilai Luas dicetak sebagai output ke Perangkat output





Melakukan identifikasi kode program

- Komentar setiap baris kode termasuk data, fungsi, prosedur, dan class diberikan

Melakukan identifikasi kode program

```
//Nama project/solution adalah ConsoleApplication1
namespace ConsoleApplication1
{
    //class diberi nama Program
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            //Perintah Console.Write untuk menampilkan pertanyaan di dalam satu baris yang sama
            Console.Write("Masukkan Nama : ");

            //Perintah dibawah adalah perintah untuk meminta input dimana tipe data variabel nama adalah string.
            //ReadLine() akan membaca teks yang nanti akan diketik
            string nama = Console.ReadLine();

            /*Perintah dibawah akan menampilkan teks di dalam tanda kurung dan tanda "+" untuk menambahkan isi variabel
            yang sebelumnya sudah dimasukkan*/
            Console.WriteLine("Nama Saya Adalah " + nama);

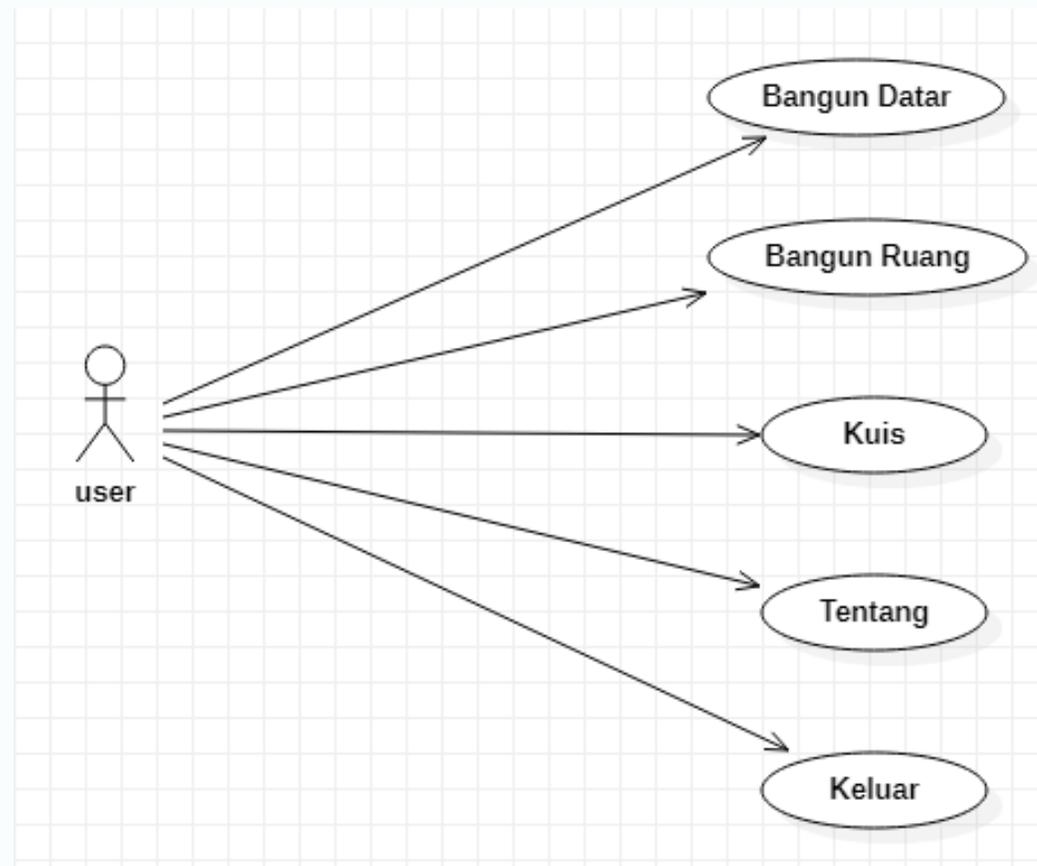
            //Dibawah ini merupakan perintah agar program tidak langsung menutup
            Console.ReadLine();
        }
    }
}
```

Membuat dokumentasi modul program

- Kegunaan modul dijelaskan

Salah satu diagram UML yang dapat digunakan untuk memberikan penjelasan mengenai modul dari sistem adalah Use Case Diagram

Membuat dokumentasi modul program



Membuat dokumentasi modul program

- Dokumen direvisi sesuai perubahan kode program
 - ✓ Versi
 - Bentuk perubahan baru dari bentuk sebelumnya
 - a. Pre-Alpha
 - Produk masih tahap pengujian.
 - b. Alpha
 - Aplikasi sudah jadi dengan fitur lengkap. Tetapi masih dilakukan pengujian atau pengurangan dan penambahan fitur

Membuat dokumentasi modul program

c. Beta

Masih dalam tahap pengujian, tetapi sudah dapat didistribusikan

d. Release Candidate

Aplikasi sudah mengalami pengujian pada versi beta, aplikasi sudah siap dipasarkan namun belum tergolong stabil



Membuat dokumentasi modul program

e. Stable

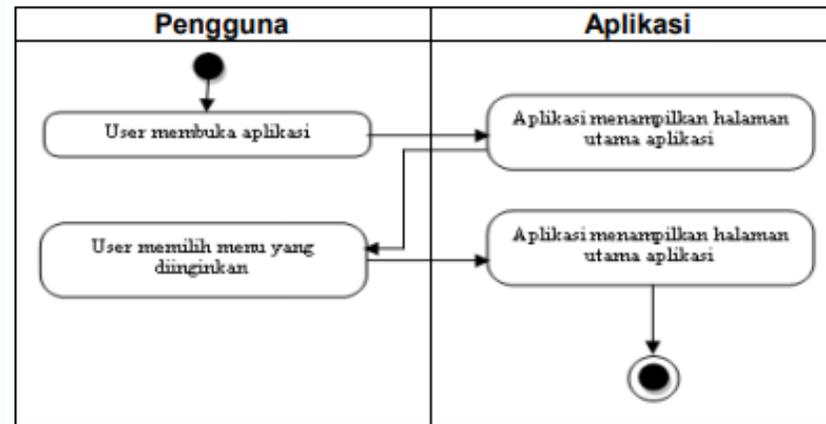
Aplikasi sudah dapat digunakan dan dipasarkan secara masal.

Membuat dokumentasi modul program

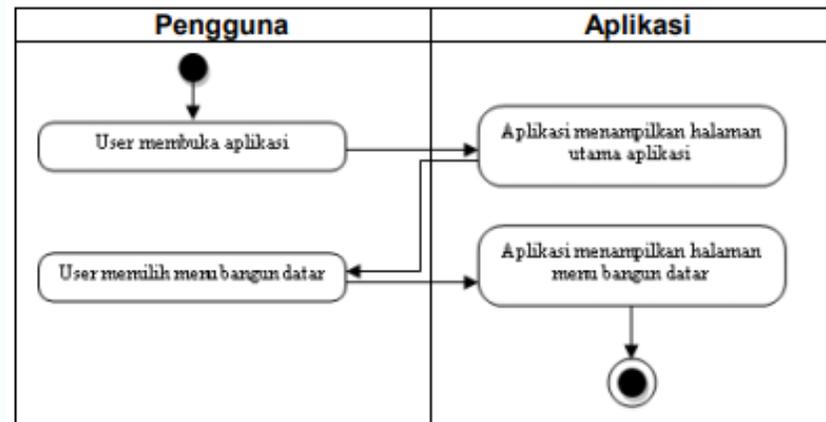
- Dokumentasi fungsi, prosedur, atau method dibuat.

Penggambaran dokumentasi ini dapat menggunakan Activity Diagram dan ERD

Membuat dokumentasi modul program



Gambar 3.6 Activity Diagram Fungsionalitas



Gambar 3.7 Activity Diagram Bangun Datar

Men-generate Dokumentasi

Tools untuk generate dokumentasi

❖ Microsoft Word



❖ HelpNDoc



❖ Document 360

DOCUMENT360

❖ Atlassian Confluence

 Confluence