



BAB II

LANDASAN TEORI



A Bahasa Program C#

Menurut Budi Raharjo (2022:1) “Pengertian C#, dikatakan sebagai C Sharp, merupakan bahasa pemrograman berorientasi objek yang dikembangkan oleh Microsoft pada awal 2000-an, dipimpin oleh Anders Hejlsberg”. Sebuah bagian dari kerangka .Net dan dimaksudkan untuk menjadi bahasa pemrograman tujuan umum sederhana yang dapat digunakan untuk mengembangkan berbagai jenis aplikasi, termasuk aplikasi konsol, windows, web, dan seluler. Seperti semua bahasa pemrograman modern, kode C# menyerupai bahasa Inggris yang tidak dapat dipahami oleh komputer. Oleh karena itu, kode C# harus diubah menjadi bahasa mesin menggunakan apa yang dikenal sebagai compiler (lihat catatan kaki). Kompiler yang akan kita gunakan dalam buku ini adalah Visual Studio Community 2015 gratis yang disediakan oleh Microsoft.

Berikut ini adalah struktur dasar program C# menurut Budi Raharjo (2021:6):

1. Pengarahan

Dari baris 1 sampai 5, kita memiliki beberapa pernyataan yang dimulai dengan kata using. Pernyataan-pernyataan ini dikenal sebagai direktif. Mereka memberi tahu kompiler bahwa program kita menggunakan namespace tertentu.

2. Namespace

Namespace hanyalah pengelompokan elemen kode terkait. Elemen-elemen ini termasuk kelas, antarmuka, enum dan struct (kita akan membahas masing-masing elemen ini di bab berikutnya).

Hak Cipta Dilindungi Undang-Undang

© Hak cipta milik IBI Kiat (Institut Bisnis dan Informatika Kwik Kian Gie)

Institut Bisnis dan Informatika Kwik Kian Gie

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



C# hadir dengan sejumlah besar kode pra-tertulis yang diatur ke dalam ruang nama yang berbeda. Namespace System berisi kode untuk metode yang memungkinkan kita berinteraksi dengan pengguna kita. Kita menggunakan dua metode ini dalam program kita metode `WriteLine()` dan `Read()`. Ruang nama lain tidak diperlukan dalam program kita. Namun, karena ruang nama ini disertakan dalam template default, kita akan membiarkannya dalam kode kita. Selain ruang nama pra-tertulis yang disediakan oleh Microsoft, kita juga dapat mendeklarasikan ruang nama kita sendiri. Satu keuntungan mendeklarasikan namespace adalah mencegah konflik penamaan. Dua atau lebih elemen kode dapat memiliki nama yang sama selama mereka termasuk dalam ruang nama yang berbeda.

© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Institut Bisnis dan Informatika Kwik Kian Gie

3. Metode Main()

Metode `Main()` adalah titik dari semua aplikasi konsol C#. Setiap kali aplikasi konsol dijalankan, metode `Main()` adalah metode pertama yang dipanggil.

4. Komentar

Kita telah membahas cukup banyak dalam bab ini. Sekarang harus memiliki pemahaman dasar tentang pemrograman C# dan cukup nyaman dengan VCS (Version Control System). Sebelum kita mengakhiri bab ini, ada satu hal lagi yang perlu dipelajari – komentar. Jika merujuk kembali ke contoh “HelloWorld” kita dan melihat baris 10, akan melihat bahwa baris ini dimulai dengan dua garis miring (`//`).

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

Berikut ini adalah tipe data dalam C# menurut Budi Raharjo (2021:9):

- © Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)**
- Hak Cipta Dilindungi Undang-Undang
1. Int (Integer)
Int singkatan dari integer (yaitu angka tanpa bagian desimal atau pecahan) dan memegang angka.
 2. Byte
Byte juga mengacu pada bilangan integral, tetapi memiliki rentang yang lebih sempit dari 0 hingga 255. Sebagian besar waktu, kita menggunakan int (Integer) daripada byte untuk bilangan integral. Namun, memprogram untuk mesin yang memiliki ruang memori terbatas, harus menggunakan byte jika yakin nilai variabel tidak akan melebihi rentang 0 hingga 255. Misalnya, jika perlu menyimpan usia pengguna, dapat menggunakan tipe data byte karena kecil kemungkinan usia pengguna akan melebihi 255 tahun.
 3. Float
Float mengacu pada angka floating point, yang merupakan angka dengan tempat desimal seperti 12,43, 5,2 dan -9,12. Float dapat menyimpan angka dari $-3,4 \times 10^{38}$ hingga $+ 3,4 \times 10^{38}$. Menggunakan penyimpanan 8 byte dan memiliki presisi sekitar 7 digit. Artinya, jika menggunakan float untuk menyimpan angka seperti 1.23456789 (10 digit), angka tersebut akan dibulatkan menjadi 1.234568 (7 digit).
 4. Double
Double juga merupakan angka floating point, tetapi dapat menyimpan rentang angka yang jauh lebih luas. Ini dapat menyimpan angka dari $(+/-)5,0 \times 10^{-324}$ hingga $(+/-)1,7 \times 10^{308}$ dan memiliki presisi sekitar 15 hingga 16 digit. Double



adalah tipe data floating point default di C#. Dengan kata lain, jika menulis angka seperti 2.34, C# memperlakukannya sebagai double secara default.

© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

5. Decimal

Decimal menyimpan angka desimal tetapi memiliki rentang yang lebih kecil daripada float dan double. Namun, ia memiliki presisi yang jauh lebih besar sekitar 28-29 digit. Jika program memerlukan tingkat presisi yang tinggi saat menyimpan bilangan non integral, harus menggunakan tipe data decimal. Contohnya adalah ketika menulis aplikasi keuangan di mana presisi sangat penting.

6. Char (Character)

Char adalah singkatan dari character dan digunakan untuk menyimpan satu karakter Unicode seperti 'A', '%', '@' dan 'p'.

7. Bool (Boolean)

Bool adalah singkatan dari boolean dan hanya menampung dua nilai: true dan false. Ini biasanya digunakan dalam pernyataan aliran kontrol.'

Menurut Dian Nandiwardhana (2019:2) C# merupakan bahasa pemrograman berbasis obyek modern yang dikembangkan pada tahun 2000 oleh Anders Hejlsber di Microsoft sebagai rival pemrograman Java. Pada saat itu, Sun (sekarang Oracle) tidak ingin Microsoft melakukan perubahan pada Java, sehingga Microsoft memutuskan untuk mengembangkan bahasa pemrogramannya sendiri.

Menurut Vivian Siahaan dan Rismon Hasiholan Sianipar (2020:1) C# merupakan pemrograman berorientasi objek. Dasar-dasar teknologi objek dan banyak teknik lain

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



pada buku ini. C# merupakan bahasa *event-driven*, yang dapat merespon event yang dipicu oleh pengguna seperti klik mouse, ketikan pada *keyboard*, pewaktu, dan lainnya.

Visual C# merupakan bahasa pemrograman visual. Bahasa visual ini memberikan beberapa statemen program untuk membangun beberapa bagian dari aplikasi.

Menurut Mochammad Rizki Romdoni (2024:1) C# merupakan bahasa pemrograman berorientasi objek yang dibuat oleh Microsoft, berjalan pada .NET Framework. C# berakar dari keluarga C, dan bahasanya mirip dengan bahasa populer lainnya seperti C++ dan Java.

Berikut ini adalah dasar-dasar C# menurut Mochammad Rizki Romdoni (2024:6):

1. Sintaks

Berikut ini kode untuk mencetak “Hello World!” ke layar:

```
using System;
namespace dasar1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Gambar 2. 1 Tampilan Code C# dalam Penulisan “Hello World!”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:6)

Penjelasan dari source code sebelumnya pada gambar 2.1 tersebut adalah sebagai berikut:

Cuplikan Kode	Penjelasan
<code>using System;</code>	Sertakan <i>namespace System</i> . Artinya kelas-kelas atau fungsi-fungsi (<i>method</i>) yang ada di dalam <i>namespace System</i> bisa dipanggil dari program.
<i>Namespace</i>	digunakan untuk mengatur, mengelompokkan kode, atau wadah untuk kelas dan <i>namespace</i> lainnya.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKGG.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKGG.



© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

{ }	Kurung kurawal { } menandai awal dan akhir suatu blok kode.
class	adalah wadah untuk data dan <i>method</i> , yang menghadirkan fungsionalitas pada program. Setiap baris kode yang dijalankan di C# harus berada di dalam kelas. <i>Jangan khawatir jika tidak mengerti cara kerjanya using System, namespace atau class; Anggap saja sebagai sesuatu yang (hampir) selalu muncul di program, dan nanti akan dipelajari lebih lanjut di bagian VII.</i>
Main	Hal lain yang selalu muncul dalam program C# adalah <i>method</i> Main yang berfungsi sebagai titik awal sebuah aplikasi mulai berjalan. Kode apa pun di dalam tanda kurung kurawal { } akan dieksekusi. Saat ini tidak perlu memahami kata kunci sebelum dan sesudah Main karena akan mengenal mereka sedikit demi sedikit saat membaca buku ini.
Console	Console adalah salah satu kelas dari <i>namespace System</i> yang memiliki <i>method</i> WriteLine() digunakan untuk mengeluarkan/mencetak teks. Dalam
	contoh ini akan menampilkan "Hello World!" Jika baris <i>using System</i> dihilangkan, maka harus ditulis secara lengkap System.Console.WriteLine() untuk mencetak / mengeluarkan teks.

Gambar 2. 2 Penjelasan Source Code dari gambar 2.1

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:6)

2. Komentar 1 Baris

Komentar satu baris dimulai dengan dua garis miring (//). Teks apa pun di antara // dan akhir baris diabaikan oleh C# (tidak akan dieksekusi).



```
Console.Write ("Hello World!"); // ini komentar satu baris dengan kode
```

Gambar 2. 3 Tampilan Code C# dalam Penulisan “komentar satu baris sebelum baris kode”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:9)

3. Komentar Multi Baris

Komentar multi-baris dimulai dengan /* dan diakhiri dengan */. Teks apa pun di antara /* dan */ akan diabaikan oleh kompilator C#.

```
/**
 * kode dibawah ini akan menampilkan kata Hello World
 * ke layar, dan ini keren
 */
Console.WriteLine("Hello World!");
```

Gambar 2. 4 Tampilan Code C# dalam Penulisan “Komentar Multi Baris”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:10)

4. Variable

Variabel adalah wadah untuk menyimpan nilai data. Di C#, ada berbagai jenis variabel (didefinisikan dengan kata kunci berbeda), misalnya:

1. Int (Integer) – Menyimpan bilangan bulat (bilangan bulat), tanpa desimal, seperti 123 atau -123.
2. Double – Menyimpan angka *floating point*, dengan desimal, seperti 19,99 atau -19,99.
3. Char (Character) – Menyimpan karakter tunggal, seperti ‘a’ atau ‘B’. Nilai char diapit oleh tanda kutip tunggal.
4. String – Menyimpan teks, seperti “Hello World”. Nilai string diapit oleh tanda kutip ganda.



5. Bool (Boolean) – Menyimpan nilai dengan dua status: benar atau salah.

C Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Institut Bisnis dan Informatika Kwik Kian Gie

5. Tipe Data

Seperti yang dijelaskan sebelumnya dalam sub bab variable; variabel dalam C# harus berupa tipe data tertentu. Tipe data menentukan ukuran dan tipe nilai variabel. Penting untuk menggunakan tipe data yang benar untuk variabel terkait; untuk menghindari kesalahan, menghemat waktu dan memori, tetapi juga akan membuat kode lebih mudah dipelihara dan dibaca. Tipe data yang umum di C# sebagai berikut:

Data Type	Size	Description
int	4 Bytes	Menyimpan bilangan bulat dari -2,147,483,648 s.d 2,147,483,647
long	8 bytes	Menyimpan bilangan bulat dari -9,223,372,036,854,775,808 s.d 9,223,372,036,854,775,807
float	4 bytes	Menyimpan bilangan pecahan dari $\pm 1.5 \times 10^{-45}$ s.d $\pm 3.4 \times 10^{38}$
double	8 bytes	Menyimpan bilangan pecahan dari $\pm 5.0 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$
bool	1 bit	Menyimpan nilai true atau false
char	2 bytes	Menyimpan sebuah karakter/huruf tunggal diapit oleh tanda kutip 1
string	2 Bytes per character	Menyimpan sebuah urutan karakter/huruf diapit oleh tanda kutip 2

Gambar 2. 5 Deskripsi Tipe Data Code C# yang Umum

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:13)

6. Bilangan

Jenis bilangan dibagi menjadi dua kelompok yaitu tipe integer dan floating. Tipe integer menyimpam bilangan bulat, positif atau negatif (seperti 123 atau -

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



456), tanpa desimal. Tipe yang valid adalah int dan long. Tipe *floating* mewakili angka dengan bagian pecahan, berisi satu atau lebih desimal. Tipe yang valid adalah float dan double. Tipe mana yang harus digunakan, bergantung pada kebutuhan dan numeriknya.

Adapun jenis bilangan bulat sebagai berikut:

1. Int (Integer)

Tipe data int dapat menyimpan bilangan bulat dari -2147483648 hingga 2147483647. Secara umum int adalah tipe data yang disukai ketika membuat variabel dengan nilai numerik.

```
int myNum = 100000;  
Console.WriteLine(myNum);
```

Gambar 2. 6 Tampilan Code C# pada Tipe Data “Int”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:14)

2. Long

Tipe data long dapat menyimpan bilangan bulat dari -9223372036854775808 hingga 9223372036854775807. Ini digunakan bila int tidak cukup besar untuk menyimpan nilai.

```
long myNum = 15000000000L;  
Console.WriteLine(myNum);
```

Gambar 2. 7 Tampilan Code C# pada Tipe Data “Long”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:15)

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

Sedangkan jenis bilangan pecahan sebagai berikut:

1. Float and Double

Tipe *float* digunakan setiap kali membutuhkan angka dengan desimal, seperti 9,99 atau 3,14515. Tipe data float atau double dapat menyimpan bilangan pecahan.

```
float myNum = 5.75F;
Console.WriteLine(myNum);

double myNum2 = 19.99D;
Console.WriteLine(myNum2);
```

Gambar 2. 8 Tampilan Code C# pada Tipe Data “Float and Double”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:15)

2. Bilangan Ilmiah

Bilangan *floating* juga bisa berupa bilangan ilmiah dengan huruf “e” untuk menunjukkan pangkat 10:

```
float f1 = 35e3F;
double d1 = 12E4D;
Console.WriteLine(f1);
Console.WriteLine(d1);
```

Gambar 2. 9 Tampilan Code C# pada Tipe Data “Bilangan Ilmiah”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:15)

7. Boolean

Tipe data boolean dideklarasikan dengan kata kunci bool (Boolean) dan hanya dapat mengambil nilai true atau false.

```
bool isCSharpFun = true;
bool isFishTasty = false;
Console.WriteLine(isCSharpFun); // Output True
Console.WriteLine(isFishTasty); // Output False
```



Gambar 2. 10 Tampilan Code C# dalam Penulisan “Boolean”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:16)

8. Karakter

Tipe data char digunakan untuk menyimpan satu karakter. Karakter harus diapit oleh tanda kutip tunggal, seperti ‘A’ atau ‘c’:

```
char myGrade = 'B';  
Console.WriteLine(myGrade);
```

Gambar 2. 11 Tampilan Code C# dalam Penulisan “Karakter”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:16)

9. String

Tipe data string digunakan untuk menyimpan rangkaian karakter (teks). Nilai string harus diapit oleh tanda kutip ganda:

```
string helloWorld = "Hello World";  
Console.WriteLine(helloWorld);
```

Gambar 2. 12 Tampilan Code C# dalam Penulisan “String”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:16)

10. Tipe Casting

Tipe casting adalah saat mengubah nilai dari satu tipe data ke tipe lainnya. Di C#, ada dua jenis casting:

1. Casting Implisit (secara otomatis) – mengonversi tipe yang lebih kecil ke ukuran tipe yang lebih besar: char (Character) → int (Integer) → long → float → double.



```
int myInt = 9;
double myDouble = myInt; // Automatic casting:
int to double

Console.WriteLine(myInt); // Outputs 9
Console.WriteLine(myDouble); // Outputs 9
```

Gambar 2. 13 Tampilan Code C# dalam Penulisan Tipe “Casting Implisit”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:17)

2. Casting Eksplisit (secara manual) – mengonversi tipe yang lebih besar ke tipe ukuran yang lebih kecil: double → float → long → int (Integer) → char (Character).

```
double myDouble = 9.78;
int myInt = (int) myDouble; // Manual casting: tipe
double keint

Console.WriteLine(myDouble); // hasil menjadi 9.78
Console.WriteLine(myInt); // hasil menjadi 9
```

Gambar 2. 14 Tampilan Code C# dalam Penulisan Tipe “Casting Eksplisit”

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:17)

11. Konversi Menggunakan Fungsi

Dimungkinkan juga untuk mengonversi tipe data secara eksplisit dengan menggunakan *method* bawaan seperti berikut:

1. Convert.ToBoolean.
2. Convert.ToDouble.
3. Convert.ToString.
4. Convert.ToInt (Integer)32.
5. Convert.ToInt (Integer)64.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

```
int myInt = 10;
double myDouble = 5.25;
bool myBool = true;

// konversi int ke string
Console.WriteLine(Convert.ToString(myInt));
// konversi int ke double
Console.WriteLine(Convert.ToDouble(myInt));
// konversi double ke int
Console.WriteLine(Convert.ToInt32(myDouble));
// konversi bool ke string
Console.WriteLine(Convert.ToString(myBool));
```

Gambar 2. 15 Tampilan Code C# dalam Penulisan Konversi

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:18)

12. User Input Data

Untuk mendapatkan data dari inputan pengguna menggunakan method `Console.ReadLine()`. Pada contoh berikut, pengguna dapat memasukkan username, yang disimpan dalam variabel `userName` kemudian mencetak nilai variabel `username`:

```
// Ketikan username dan tekan enter
Console.WriteLine("Masukan username:");

// Membuat sebuah variable tipe string dan dapatkan
// data dari input user yang disimpan ke variable
// userName
string userName = Console.ReadLine();

// Cetak nilai variable username
Console.WriteLine("Username is: " + userName);
```

Gambar 2. 16 Tampilan Code C# dalam Penulisan Input

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:19)

13. Operator

Operator digunakan untuk melakukan operasi pada variabel dan nilai. Pada contoh di bawah, operator `+` digunakan untuk menjumlahkan dua buah nilai:



```
int x = 100 + 50;
Console.WriteLine("Username is: " + x);
```

Gambar 2. 17 Tampilan Code C# dalam Penulisan Operator

Sumber: Buku Ajar Pemrograman C# Dasar .NET Framework (2024:19)

Meskipun operator + sering digunakan untuk menjumlahkan dua nilai, seperti pada contoh di atas, operator tersebut juga dapat digunakan untuk menjumlahkan variabel dan nilai, atau variabel dan variabel lain.

B. Platform .NET

Menurut Dian Nandiwardhana (2019:15) .NET adalah sebuah *platform* open source untuk mengembangkan dan menjalankan berbagai jenis aplikasi lintas-platform. .NET sendiri tersedia dan dapat digunakan secara gratis.

Menurut Dian Nandiwardhana (2019:15) Dengan .NET, dapat menggunakan beberapa bahasa pemrograman, *editor*, dan *library* untuk membangun aplikasi web, mobile, desktop, game, dan bahkan Internet of Things atau LoT.

Saat ini, ada tiga implementasi .NET untuk mendukung pemrograman lintas platform yaitu:

1. .NET Framework yang mendukung website, service, aplikasi desktop, dan lainnya di platform Windows.
2. .NET Core yang merupakan implementasi .NET lintas platform untuk aplikasi web, server, dan juga aplikasi konsol di macOS, Windows, dan Linux.
3. Xamarin / Mono yang merupakan implementasi .NET untuk menjalankan aplikasi di semua sistem operasi mobile utama seperti iOS (iPhone Operating System), Android, dan Windows phone.



C. Bahasa Program Java



Menurut Dwi Hastuti dan Yusril Arief (2022:1) Pencipta bahasa pemrograman Java adalah James Gosling. Beliau menciptakan bahasa ini saat masih bergabung di Sun Microsystems yang saat ini sudah menjadi bagian dari Oracle dan dirilis tahun 1995. Bahasa Java dijalankan menggunakan JVM atau Java Virtual Machine, hal ini yang menyebabkan Java lebih fleksibel dan bisa dijalankan di System operasi Windows, Mac OS atau Linux. Sesuai dengan slogannya yaitu “*Write once, Run anywhere*”.

Berikut ini terdapat tiga jenis pemrograman Java menurut Dwi Hastuti dan Yusril Arief (2022:2), antara lain:

1. Java Standart Edition

Java Standart Edition (J2SEE) atau yang biasa disebut Java SE yang secara umum banyak digunakan pemrograman Bahasa Java. Java SE menggunakan JVM atau Java Virtual Machine untuk menjalankan program bersama librarynya. Java SE biasanya berupa aplikasi berbentuk text, aplikasi berbasis window atau Graphical User Interface (GUI) dan aplikasi-aplikasi berskala kecil.

2. Java Enterprise Edition

Java Enterprise Edition (JEE) banyak dimanfaatkan untuk membuat aplikasi berskala besar. Komponen JEE terdiri dari:

- Client dan Applet adalah komponen yang berjalan di sisi Client.
- Web berupa Servlet atau Java Server Pages yang berjalan di server.
- Enterprise Javabeans (EJB) adalah teknologi yang digunakan untuk mengembangkan komponen di sisi server.



3. Java Platform Micro Edition (J2ME)

Teknologi Java yang banyak digunakan untuk membuat aplikasi mobile atau aplikasi yang bersifat embedded, biasanya berupa aplikasi handphone atau PDA.

Menurut Budi Hartono (2022:1) Sejarah bahasa pemrograman Java dikembangkan pada tahun 1991 oleh lima ahli komputer Ed Frank, Mike Sheridan, Chris Warth, Patrick Naughton, dan James Gosling yang semuanya bekerja untuk Sun Microsystems Inc. Mereka membutuhkan waktu 18 bulan untuk mengembangkan program, yang awalnya bernama "Oak". Itu berganti nama menjadi Java pada tahun 1995 karena masalah hak cipta. Konsepnya adalah untuk membuat bahasa pemrograman yang dapat digunakan di seluruh platform dan yang dapat membangun perangkat lunak tertanam untuk elektronik konsumen. Bahasa pemrograman populer pada saat itu, C dan C++ tidak efisien untuk tujuan ini, karena mereka bergantung pada platform karena program yang ditulis di dalamnya harus dikompilasi terlebih dahulu untuk perangkat keras tertentu sebelum diluncurkan. Selain itu, kode yang dikompilasi tidak efisien untuk prosesor lain dan harus dikompilasi ulang. Oleh karena itu, tim lima, yang juga dikenal sebagai Tim Hijau, mulai bekerja membangun bahasa pemrograman yang lebih mudah. Mereka bermain-main selama satu setengah tahun dalam menciptakan bahasa pemrograman platform independen yang kompak, yang memungkinkan seorang programmer untuk membangun kode yang dapat berjalan pada prosesor yang berbeda di bawah berbagai lingkungan.

Hak Cipta Dilindungi Undang-Undang

© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

Berikut ini adalah fitur utama Java menurut Budi Hartono (2022:2), sebagai berikut:

© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Institut Bisnis dan Informatika Kwik Kian Gie

1. Sederhana

Java dianggap sebagai bahasa yang sederhana, karena tidak memiliki fitur yang rumit seperti alokasi memori eksplisit, pointer, pewarisan berganda, dan overloading Operator.

2. Aman

Java dilengkapi dengan firewall virtual antara komputer dan aplikasi. Kode java dibatasi di dalam Java Runtime Environment (JRE), yang tidak menyetujui akses tidak sah untuk sumber daya sistem.

3. Portabel

Sebuah kode yang ditulis dalam Java pada satu platform dapat berjalan pada platform lain pada mesin yang berbeda. Kode byte dapat diangkut ke platform apa pun untuk operasi, yang membuat kode java sangat portabel.

4. Platform Independen

Platform mengacu pada pengaturan yang telah ditetapkan sebelumnya untuk menjalankan program, mematuhi batasannya, dan menggunakan fitur-fiturnya. Selama fase kompilasi, program java diubah menjadi kode byte, yang dapat digunakan untuk platform apa pun seperti Mac/OS (Operating System), Linux, atau Windows dan sebaliknya. Itulah mengapa Java adalah bahasa pemrograman platform independen.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



5. Multi-utas

Java mendukung multi-threading karena memungkinkan program untuk melakukan beberapa tugas sekaligus.

6. Berorientasi pada objek

Java adalah bahasa pemrograman berorientasi objek, karena dapat mengatur program sebagai sekelompok objek yang masing-masing mewakili turunan dari kelas. Empat konsep utama OOP (Object-Oriented Programming) adalah: polimorfisme, pewarisan, enkapsulasi, dan abstraksi.

7. Didistribusikan

Melalui pemrograman Java, dapat mengembangkan aplikasi terdistribusi. Enterprise Java Beans (EJB) dan Remote Method Invocation (RMI) digunakan untuk mengembangkan aplikasi terdistribusi menggunakan Java. Sederhananya, dapat mendistribusikan program Java pada beberapa sistem yang terhubung satu sama lain melalui Internet. Objek dalam Java Virtual Machine (JVM) dapat menjalankan protokol menggunakan JVM (Java Virtual Machine) jarak jauh.

8. Kokoh

Kesalahan runtime yang salah penanganan dan kesalahan manajemen memori adalah dua masalah utama yang menyebabkan kegagalan program. Java dapat menangani masalah ini dengan efisiensi tinggi. Kesalahan runtime yang salah dapat diselesaikan melalui protokol Penanganan Pengecualian, sementara kesalahan manajemen memori dapat diselesaikan dengan pengumpulan

© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Institut Bisnis dan Informatika Kwik Kian Gie

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



sampah, yang merupakan de-alokasi otomatis objek yang sudah tidak diperlukan.



Menurut Barda Hudaya (2024:2) Java adalah salah satu bahasa pemrograman yang paling populer dan banyak digunakan. Dikenal dengan prinsip “Write Once, Run Anywhere” (WORA), Java memungkinkan pengembang menulis kode satu kali dan menjalankannya di berbagai platform tanpa perubahan besar.

Berikut ini beberapa prinsip dasar pengembangan Java menurut Barda Hudaya (2024:4):

1. Sederhana, Berorientasi Objek, dan Familiar: Java dirancang agar mudah dipelajari oleh pemrograman yang sudah familiar dengan bahasa pemrograman lain seperti C dan C++.
2. Dapat diportasikan: Salah satu kekuatan utama Java adalah kemampuannya untuk berjalan di platform tanpa perlu diubah kodenya, sesuai dengan prinsip “Write Once, Run Anywhere” (WORA).
3. Aman dan Kuat: Java mengadopsi fitur keamanan yang kuat, terutama dalam mengelola memori dan menangani kesalahan yang tidak terduga, menjadikannya bahasa yang andal untuk pengembangan perangkat lunak berskala besar.
4. Multi-threaded: Java mendukung pemrograman multithreading, yang memungkinkan beberapa bagian program berjalan secara paralel, meningkatkan efisiensi dan performa aplikasi.
5. Berkinerja Tinggi: Meski menggunakan bytecode yang diinterpretasi oleh Java Virtual Machine (JVM), Java tetap memiliki kinerja yang tinggi berkat optimisasi Just-In-Time (JIT).

Hak Cipta Dilindungi Undang-Undang

© Hak cipta dimiliki oleh Institut Bisnis dan Informatika Kwik Kian Gie

Institut Bisnis dan Informatika Kwik Kian Gie

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



Berikut adalah beberapa versi utama Java beserta fitur-fitur penting menurut Barda

Hudaya (2024:5):

1. Java 2 (J2SE 1.2, 1998): Dengan rilis ini, Java mulai dibagi menjadi tiga edisi utama: J2SE (Java Standard Edition) untuk aplikasi desktop, J2EE (Java Enterprise Edition) untuk pengembangan aplikasi enterprise, dan J2ME (Java Micro Edition) untuk perangkat mobile dan embedded. Versi ini juga memperkenalkan *Swing* untuk membuat GUI (Graphical User Interface) serta meningkatkan performa JVM (Java Virtual Machine).
2. Java 5 (2004): Salah satu pembaruan terbesar dalam sejarah Java. Rilis ini memperkenalkan fitur baru seperti *generics*, *annotations*, dan *enhanced for-loop*. Perubahan signifikan lainnya adalah penggantian penomoran dari 1.x ke 5.0.
3. Java 6 (2006): Meningkatkan performa JVM (Java Virtual Machine), memperkenalkan *Compiler API* (Application Programming Interface), dan integrasi dengan sistem desktop (misalnya, kemampuan membuat aplikasi tray icon dan splash screen).
4. Java 7 (2011): Membawa fitur-fitur baru seperti *try-with-resources*, yang memudahkan pengelolaan sumber daya, dan peningkatan sintaks, seperti *diamond operator* untuk *generics*.
5. Java 8 (2014): Java 8 adalah salah satu versi yang paling signifikan dengan penambahan *lambda expressions* dan *stream API* (Application Programming Interface), yang membuka pintu untuk gaya pemrograman fungsional di Java. Selain itu, *Date and Time API* (Application Programming Interface) baru juga diperkenalkan untuk menggantikan *API* (Application Programming Interface) tanggal yang lama dan tidak efisien.

Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Institut Bisnis dan Informatika Kwik Kian Gie

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



6. Java 9 (2017): Memperkenalkan module system (Project Jigsaw), yang memungkinkan pengembang untuk membagi program menjadi modul-modul yang dapat diatur, serta meningkatkan keamanan dan fleksibilitas pengembangan aplikasi besar.
7. Java 11 (2018): Versi ini menjadi titik di mana Oracle mulai mengadopsi siklus rilis baru setiap enam bulan. Java 11 menambahkan dukungan untuk *Local-Variable Syntac* untuk variabel lambda, dan mendukung penerapan yang lebih ringan.
8. Java 17 (2021): Ini adalah salah satu versi dukungan jangka panjang Long-Term Support (LTS) yang diperkenalkan dengan berbagai peningkatan performa dan stabilitas, serta penghapusan fitur-fitur yang tidak digunakan lagi. Pada versi ini, terdapat peningkatan signifikan pada JVM (Java Virtual Machine), garbage collector baru (ZGC), serta pengenalan *sealed classes*.

Berikut adalah pengenalan Java menurut Vivian Siahaan (2020:1) Java dikenal sebagai bahasa pemrograman untuk aplikasi-aplikasi internet. Namun, buku ini, dan banyak buku lain, memandang Java sebagai bahasa pemrograman umum yang cocok untuk hampir semua aplikasi, baik yang melibatkan internet atau tidak.

Menurut Vivian Siahaan (2020:1) Pada tahun 1991, James Gosling memimpin sebuah tim di Sun Microsystems yang mengembangkan versi pertama dari Java (saat ini belum dikenal dengan Java). Versi pertama dari bahasa ini didesain untuk memprogram peralatan-peralatan rumah tangga, seperti mesin cuci dan televisi. Meskipun bukan bidang aplikasi tingkat tinggi, bukanlah pekerjaan mudah untuk merancang bahasa semacam itu. Peralatan rumah tangga dikendalikan oleh berbagai prosesor komputer (chip). Bahasa yang dikembangkan oleh Goslin diperlukan agar dapat diterapkan pada semua prosesor yang berbeda-beda. Selain itu, peralatan rumah tangga umumnya tidak

Hak Cipta Dilindungi Undang-Undang

© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



mahal, sehingga pihak manufaktur tidak berminat untuk menginvestasikan dana yang besar untuk pengembangan kompuler yang kompleks. (Kompuler bertugas untuk menerjemahkan sebuah bahasa menjadi bahasa yang dapat dipahami oleh prosesor). Untuk menyederhanakan pekerjaan penerjemahan program untuk setiap kategori peralatan, tim menggunakan proses translasi dua langkah. Program pertama-tama diterjemahkan menjadi sebuah bahasa perantara yang sama untuk semua peralatan (atau semua komputer), dan kemudian program menerjemahkan bahasa perantara ini menjadi bahasa mesin untuk peralatan atau komputer tertentu. Bahasa perantara ini dikenal dengan Java bytecode atau bytecode. Karena hanya ada satu bahasa perantara, langkah tersulit dari translasi dua langkah dari program ke bahasa perantara ke bahasa mesin menjadi sama untuk semua peralatan (atau semua komputer). Bahasa Java, ternyata pada prosesnya, tidak hanya diaplikasikan pada peralatan-peralatan rumah tangga, tetapi telah berevolusi menjadi bahasa pemrograman yang paling luas digunakan. Sekarang, Java dimiliki oleh Oracle Corporation, yang membeli Sun Microsystems pada tahun 2010.

D. Algoritma

Menurut Dwi Hastuti dan Yusril Arief (2022:13) Algoritma adalah seluruh urutan langkah-langkah logis untuk mempermudah menyelesaikan suatu masalah. Algoritma berasal dari nama Abu Ja'far Muhammad Ibn Musa Al Khwarizmi (770 – 840). Di dalam literatur barat beliau dikenal dengan sebutan Algorizm.

Menurut Microsoft Press Computer and Internet Dictionary (1997, 1998:1) Algoritma adalah urutan langkah logis tertentu untuk memecahkan suatu masalah. Yang ditekankan adalah urutan langkah logis, yang berarti algoritma harus mengikuti suatu urutan tertentu, tidak boleh melompat-lompat.

Hak Cipta Dilindungi Undang-Undang

Hak Cipta Milik IBIKKG (Kwik Kian Gie) dan Infomatika Kwik Kian Gie

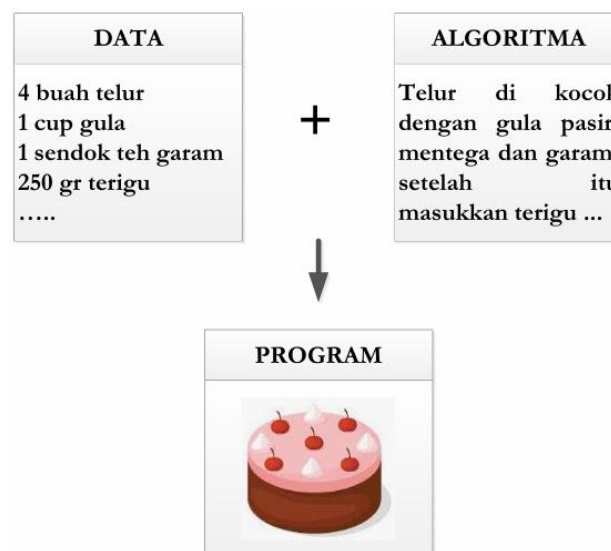
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



Menurut Kani (2020:1.19) Algoritma adalah suatu upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu output tertentu.

Berikut ini adalah komponen penting algoritma pemrograman menurut Bei Harira Irawan dan Deddy Prihadi (2022:13) sebagai berikut:

1. Input merupakan variabel atau bahan yang perlu dipersiapkan untuk dimasukkan kedalam sebuah proses agar dapat diolah.
2. Proses merupakan cara atau langkah dalam mengolah variabel atau bahan tersebut untuk menghasilkan suatu output.
3. Output adalah hasil yang didapat dari hasil proses pengolahan variabel atau bahan. Output yang dihasilkan haruslah merupakan solusi terhadap masalah pemrograman.



Gambar 2. 18 Ilustrasi Algoritma Pemrograman

Sumber: Logika Dan Algoritma Pemrograman (2022:13)

Berikut adalah lima karakteristik algoritma menurut Kusumaningtyas dan Wahyuddin (2022:2):

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



1. Input merupakan permasalahan dimana kamu berpengalaman serta mau mencari pemecahan. Algoritma mempunyai nol ataupun lebih nilai input.
2. Proses, yakni langkah-langkah yang harus dicoba buat mencapai tujuan akhir.
3. Output yakni pemecahan ataupun penanda akhir yang diperoleh dari algoritma. Algoritma mempunyai paling tidak satu output.
4. Instruksi-instruksi yang jelas dan tidak ambigu, yakni instruksi yang jelas dalam algoritma sehingga tidak terjalin kesalahan dalam menghasilkan output.
5. Tujuan akhir yang dicapai merupakan akhir program, serta program berakhir kalau tujuan akhir tercapai.

© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Berikut ini ada beberapa fungsi algoritma menurut Sulasmoro (2022:3):

1. Dapat menyederhanakan program yang lingkungan serta besar.
2. Buat memudahkan memprogram buat sesuatu permasalahan tertentu.
3. Algoritma bisa digunakan keesekian kali buat menuntaskan permasalahan.
4. Menolong mengklasifikasikan permasalahan secara logis serta sistematis.
5. Meminimalkan persiapan program.
6. Sanggup mempraktikkan pendekatan *top-down, divide and rule*.
7. Buat lebih gampang membuat program yang lebih baik serta terstruktur, lebih gampang dimengerti serta diperluas.
8. Mempermudah proses modifikasi pada program karena bisa dicoba hanya pada satu modul tanpa harus mengubah modul yang lain.
9. Kala terjalin kesalahan, algoritma dapat membantu menciptakannya karena alur kerja yang jelas.
10. Mempermudah proses dokumentasi.

Menurut Oki Arifin (2023:65) Algoritma adalah sebuah prosedur atau metode yang terdiri dari langkah-langkah terurut dan terdefinisi dengan baik untuk menyelesaikan

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



suatu masalah atau tugas. Algoritma dapat digunakan untuk berbagai macam bidang seperti matematika, ilmu komputer, dan bahkan di kehidupan sehari-hari. Dalam pengembangan perangkat lunak, algoritma sangat penting untuk memastikan bahwa program yang dibuat berjalan dengan benar dan efisien.

Berikut ini jenis-jenis algoritma menurut Putu Harry Gunawan (2022:3):

1. Structured English (SE)

Jenis algoritma Structured English (SE) ini biasanya menuliskan algoritma dalam list/urutan langkah-langkah dengan bahasa manusia. Tidak ada penulisan baku untuk menuliskan algoritma dalam bentuk ini. Sebagai contohnya dapat dilihat pada, dibawah ini:

-
1. Procedure MyProcedure
 2. Definisikan kondisi awal
 3. Hitung Δx dari persamaan (1)
 4. Hitung $\int_a^b f(x) dx$ dengan menggunakan metode X
 5. Update Δt menggunakan persamaan (13)
 6. Ulangi Langkah ke 3 jika $\Delta t < \epsilon$
-

Gambar 2. 19 Algoritma jenis Structures English (SE)

Sumber: Logika Matematika untuk Analisis Algoritma (2022:3)

2. Pseudocode

Jenis algoritma ini hampir sama dengan jenis algoritma Structured English (SE), tetapi langkah-langkah yang ditulis dalam bentuk bahasa pemrograman komputer. Contoh algoritma jenis ini dapat dilihat, dibawah ini:



1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.

```
Results: Tuliskan output
Input: Tuliskan input
Inisialisasi;
while <kondisi> do
    Instruksi;
    If <kondisi> then
        Instruksi;
    else
        Instruksi;
    end
end
```

Gambar 2. 20 Algoritma jenis pseudocode

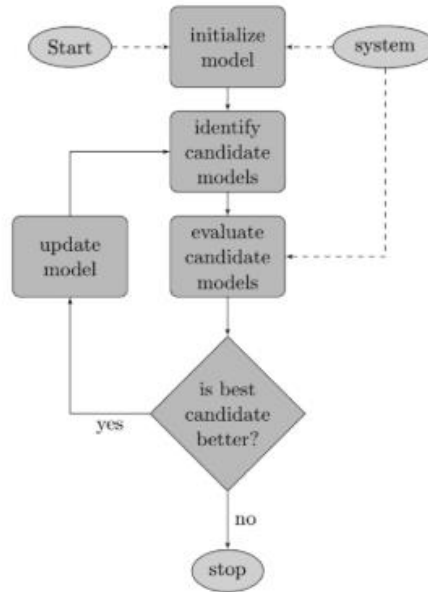
Sumber: Logika Matematika untuk Analisis Algoritma (2022:4)

3. Flowchart

Flowchart dihadirkan dalam bentuk bangun ruang dua dimensi yang dihubungkan dengan garis panah untuk mendeskripsikan arah aliran. Kita bisa menemukan bentuk flowchart untuk menggambarkan suatu algoritma atau kode pemrograman dalam buku-buku tugas akhir mahasiswa S-1. Bentuk algoritma dalam flowchart memiliki kelebihan yang mudah dibaca dan mudah diimplementasikan oleh programmer, tetapi sering memakan banyak tempat terutama algoritma yang kompleks. Contoh flowchart yang dibuat dengan menggunakan LaTeX seperti gambar dibawah:



1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



Gambar 2. 21 Algoritma dalam bentuk flowchart

Sumber: <https://texample.net/tikz/examples/simple-flow-chart/>

Berikut ini adalah langkah-langkah algoritma iteratif yang harus terpenuhi menurut Putu Harry Gunawan (2022:6):

1. Claim: pada langkah ini, kita wajib mengetahui atau mengklaim apa yang akan dikerjakan oleh algoritma yang akan dianalisis. Klaim ini penting dalam membantu pembentukan loop invariant pada langkah berikutnya.
2. Fact about algorithm: pada langkah ini, tuliskan semua informasi yang diberikan dalam algoritma. Tentu saja bagian ini sangatlah penting karena bisa diilustrasikan sebagai senjata ampuh dalam membuktikan *loop invariant* dan kebenaran dari algoritma.
3. Loop invariant: pada langkah ini, kita harus membentuk pernyataan yang akan kita lakukan dengan induksi matematika. Pernyataan yang dibuat tentunya harus memperhatikan *fact about algorithm* dan *claim* pada langkah selanjutnya.



4. Conclusion: untuk mengakhiri pembuktian maka langkah terakhir menunjukkan kebenaran berupa lagoritma selalu memberikan hasil yang benar untuk setiap input-an yang sesuai.



Hak cipta milik BI KGG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Menurut Khairunnisa (2023:58) Analisis dalam algoritma adalah proses untuk mempelajari dan mengevaluasi kinerja atau efisiensi dari sebuah algoritma. Analisis ini melibatkan pengukuran dan pemodelan matematis untuk memahami sejauh mana algoritma tersebut dapat menyelesaikan tugas dengan menggunakan sumber daya yang tersedia.

Berikut ada beberapa aspek yang sering dianalisis dalam algoritma menurut Khairunnisa (2023:58):

1. Waktu Eksekusi: Analisis ini bertujuan untuk memahami seberapa cepat algoritma dengan menyelesaikan tugasnya. Waktu eksekusi sering diukur dengan kompleksitas waktu algoritma, yang dapat dinyatakan dalam notasi Big O. Analisis ini membantu dalam membandingkan efisiensi algoritma dan memilih algoritma yang paling cepat untuk masalah yang diberikan.
2. Penggunaan memori: Analisis ini melibatkan pemodelan penggunaan memori oleh algoritma. Ini membantu dalam memahami berapa banyak memori yang akan digunakan algoritma saat dieksekusi. Penggunaan memori algoritma dapat dinyatakan dalam kompleksitas ruang, yang juga dapat dinyatakan dalam notasi Big O.
3. Kompleksitas Algoritma: Kompleksitas algoritma mengukur seberapa banyak waktu atau ruang yang dibutuhkan oleh algoritma seiring dengan ukuran masalah yang diberikan. Analisis kompleksitas membantu dalam memahami seberapa efisien algoritma dalam menangani masalah yang semakin besar. Hal

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



ini penting dalam memilih algoritma yang tepat untuk memenuhi kebutuhan dan keterbatasan sumber daya yang ada.

© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Terdapat beberapa strategi teknik pemecahan masalah dalam algoritma, untuk suatu permasalahan yang rumit menurut Sigit Susanto Putro (2019:6):

1. Divide and Conquer: Memecahkan masalah yang besar kemudian membaginya menjadi masalah yang lebih sederhana. Masalah yang lebih sederhana ini diselesaikan dan hasilnya digabungkan dengan solusi dari masalah sederhana yang lainnya sehingga didapatkan solusi permasalahan utama.
2. Decrease and Conquer: Teknik ini juga membagi masalah utama menjadi beberapa bagian, namun permasalahan ukuran permasalahan utama dikurangi terlebih dahulu sehingga menjadi lebih sederhana.
3. Transform and Conquer: Merubah penyajian masalah ke dalam bentuk lain agar lebih mudah diselesaikan, contohnya: menyelesaikan sistem persamaan linier dengan matriks.

E. Binary Search

Menurut Evi Pratiwi (2020:100) Algoritma binary search atau pencarian biner merupakan metode pencarian pada data terurut yang paling efisien. Metode pencarian biner digunakan untuk kebutuhan pencarian dengan waktu yang cepat.

Menurut Evi Pratiwi (2020:100) Pencarian Biner (*Binary Search*) dilakukan untuk:

1. Memperkecil jumlah operasi perbandingan yang harus dilakukan antara data yang dicari dengan data yang ada khususnya untuk jumlah data yang sangat besar ukurannya.



2. Prinsip dasarnya adalah melakukan proses pembagian ruang pencarian secara berulang-ulang sampai data ditemukan atau sampai ruang pencarian tidak dapat dibagi lagi (berarti ada kemungkinan data tidak ditemukan).
3. Syarat utama untuk *binary search* adalah data harus sudah terurut, misalkan terurut menaik.

Menurut Hari Soetanto (2022:132) Binary Search adalah metode terbaik dalam search. Memulai search dari lokasi tengah (MIDDLE).

Menurut Mustakim (2024:145) Pencarian biner (Binary Search) merupakan salah satu algoritma pencarian. Pada setiap langkah, algoritma ini membandingkan elemen input (x) dengan nilai elemen tengah dengan array. Jika nilainya cocok, kembalikan indeks tengah. Jika tidak, jika x kurang dari elemen tengah, maka algoritma akan kembali ke sisi kiri elemen tengah, jika tidak, akan kembali ke sisi kanan elemen tengah.

F. Logika

Menurut Bei Harira Irawan dan Deddy Prihadi (2022:9) Dalam pemrograman, logika pemrograman merupakan sebuah cara penyelesaian suatu masalah pemrograman dengan prinsip dan aturan tertentu sehingga dapat menghasilkan output yang sesuai dengan apa yang kita inginkan. Logika yang digunakan dalam pemrograman merupakan kemampuan dasar wajib yang harus dikuasai seorang programmer diawal belajar sehingga akan mudah untuk memahami cara menyelesaikan masalah pada pemrograman lanjutan lainnya yang lebih rumit.

Hak Cipta Dilindungi Undang-Undang

© Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



Beberapa logika dasar yang harus dipahami dalam pemrograman menurut Bei

Harira Irawan dan Deddy Prihadi (2022:10) antara lain:

1. Logika Operasional (Aritmatika)

Perhitungan matematika merupakan hal dasar dalam pemrograman dimana setiap pemrograman, perhitungan matematis seperti penjumlahan, pengurangan, perkalian, pembagian dan lainnya akan selalu digunakan dalam menyelesaikan masalah atau mengembangkan suatu program.

2. Logika Operasional (Perbandingan)

Operator yang digunakan untuk membandingkan nilai dengan nilai lainnya yang hasilnya adalah TRUE atau FALSE. Operator ($<$) atau kurang dari akan menghasilkan TRUE jika nilai di sebelah kiri lebih kecil dari nilai di sebelah kanan. Jika tidak demikian, maka akan menghasilkan FALSE. Contoh $7 < 10$ maka akan menghasilkan nilai TRUE.

3. Logika Fungsi

Bayangkan jika kamu diminta untuk menghitung gaji karyawan lalu ada 300 orang operator dengan gaji pokok yang sama untuk dihitung gajinya. Tentunya sistem akan menghitung satu persatu gaji karyawan sebanyak 300 orang. Fungsi merupakan sebuah deretan blok kode untuk menyelesaikan sebuah task atau fitur, sehingga apabila kita membutuhkan fitur tersebut, kita dapat dengan mudah menggunakan kode fungsi yang sudah kita buat tadi. Jadi dengan memiliki 1 fungsi rumusan perhitungan gaji, maka fungsi tersebut dapat kita gunakan berulang-ulang untuk menghitung gaji 300 karyawan. Sebuah fungsi sangat dibutuhkan agar memudahkan pengaturan kode dan pelacakan permasalahan (error).

Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)

Hak Cipta Dilindungi Undang-Undang

Institut Bisnis dan Informatika Kwik Kian Gie

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



4. Logika Looping

Logika looping atau perulangan digunakan untuk menjalankan kode atau instruksi yang sama berulang-ulang hingga kondisi terpenuhi atau jika kondisi stop/berhenti tercapai. Contoh gampangnya kita akan mencetak 100 baris nama kita masing-masing. Apabila tidak menggunakan looping, maka kita akan menuliskan 100 baris nama kita dalam program. Namun bila kita menggunakan fungsi looping, maka untuk mencetak nama 100 baris bahkan 1000 baris hanya dilakukan dalam 2 atau 3 baris coding saja.

5. Logika Kondisional

Ketika kamu mulai menulis kode, kamu pasti akan menjalankan berbagai jenis perintah dalam berbagai kondisi tertentu. Di saat seperti ini, logika kondisional adalah cara sederhana yang bisa digunakan untuk mengecek kondisi spesifik dan menjalankan perintah berdasarkan kondisi tersebut. Contohnya apabila kondisi lampu lalu lintas berwarna merah, maka kendaraanmu harus berhenti. Apabila kondisi lampu berwarna merah berarti dalam kondisi TRUE, maka kendaraan bermotor wajib berhenti, namun apabila buka merah (hijau), maka kendaraan boleh berjalan.

6. Logika Perbandingan

Logika perbandingan ini biasanya digunakan untuk membuat persyaratan atas kondisi tertentu yaitu dengan cara mengukur nilai dua atau lebih variabel apakah sama, lebih kecil, lebih besar atau tidak sama. Misalnya perintah, “jika nilai angka mahasiswa 90 maka nilai abjadnya adalah A”.



7. Logika Boolean

- © Hak cipta milik IBI KKG (Institut Bisnis dan Informatika Kwik Kian Gie)
- Logika boolean dalam pemrograman digunakan untuk membuat persyaratan tertentu dengan hasil atau kondisi akhir bernilai “BENAR” atau “SALAH”. Misalnya, penggunaan logika “dan” akan menghasilkan nilai “BENAR” jika kedua kondisi persyaratan terpenuhi.

G Struktur Data

Menurut Bei Harira Irawan dan Deddy Prihadi (2022:12) Struktur data adalah suatu cara menyimpan atau mempresentasikan data di dalam komputer agar bisa dipakai secara efisien. Pemakaian stuktur data yang tepat di dalam proses pemrograman akan menghasilkan algoritma yang lebih jelas dan tepat, sehingga menjadikan program secara keseluruhan lebih efisien dan sederhana.

H Pemrograman

Menurut Bei Harira Irawan dan Deddy Prihadi (2022:16) Pemrograman adalah sebuah proses menulis kode, melakukan pengujian, memperbaiki (*debug*) dan memelihara kode dalam membangun suatu program komputer. Kode ini tidak hanya ditulis dalam 1 bahasa pemrograman saja, namun dapat ditulis dalam ebrabgai bahasa pemrograman. Tujuan dari pemrograman ini adalah untuk memuat suatu program yang dapat dieksekusi dalam melakukan suatu perhitungan atau ‘pekerjaan’ sesuai dengan keinginan pemrogramnya (*programmer*). Untuk melakukan pemrograman, diperlukan keterampilan dalam membaca suatu algoritma, memiliki logika pemrograman yang baik, menguasai bahasa pemrograman bahkan pada banyak kasus, pengetahuan-pengetahuan lain seperti matematika dan analisa suatu proses kejadian wajib dikuasai.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.



Menurut Barda Hudaya (2024:3) Pemrograman adalah inti dari perkembangan teknologi modern.

Menurut Suzi Oktavia Kunang (2022:4) Bahasa pemrograman digunakan untuk mempermudah manusia dalam berinteraksi dengan komputer. Syarat utama untuk membuat program adalah dengan menggunakannya sesuai dengan kaidah-kaidah yang berlaku dalam bahasa pemrograman tersebut. Masing-masing bahasa pemrograman mempunyai ciri khas atau kaidah tersendiri. Oleh karena itu sebelum membuat sebuah program kita harus mengerti tentang aturan penulisan (*syntax*) dalam bahasa pemrograman tersebut.

Menurut Sigit Susanto Putro (2019:3) Pemrograman adalah proses membuat suatu program dengan algoritma dan diterapkan dalam bahasa pemrograman.

I. Benchmarking

Menurut Widhi Yahya, Adhitya Bhawiyuga dan Eko Sakti Pramukantoro (2019:212) *Benchmarking* adalah salah satu istilah dalam administrasi sistem server yang tujuannya untuk mengetahui dan mengukur kinerja dari layanan-layanan yang ada di sistem server. *Benchmarking* sangat diperlukan sebelum berbicara untuk meningkatkan kinerja, hasil dari *benchmarking* dapat digunakan untuk proses evaluasi kinerja layanan terhadap hardware yang tersedia. Pada bagian ini akan disajikan mekanisme *benchmarking* pada layanan-layanan umum yang ada pada sistem server.

J. Penelitian Terdahulu

Dalam penelitian yang dilakukan oleh Nurul Imamah dan Moch Ilham Bahari pada tahun 2021 dengan judul “*Perbandingan Algoritma Sequential Search dan Algoritma Binary Search pada Aplikasi Kamus Bahasa Indonesia Menggunakan PHP dan*



JQUEY”, penelitian tersebut membandingkan kinerja algoritma Sequential Search dan Binary Search untuk menentukan algoritma yang lebih optimal. Hasil pengujian menunjukkan bahwa algoritma Binary Search memiliki keunggulan dalam kecepatan dengan rata-rata waktu 0,0405 detik, sedangkan algoritma Sequential Search lebih efisien dengan penggunaan memori. Penelitian ini membuktikan dalam pemilihan algoritma yang tepat dapat berpengaruh terhadap performa aplikasi.

Dalam penelitian yang dilakukan oleh Moh Arif Jauhari, Dini Hamidin, dan Marwanto Rahmatuloh pada tahun 2017 yang berjudul “Komparasi Stabilitas Eksekusi Kode Bahasa Pemrograman .Net C# Versi 4.0.3019 dengan Google Golang Versi 1.4.2 Menggunakan Algoritma Bubble Sort dan Insertion Sort”, pengujian menggunakan algoritma bubble sort dan insertion sort dengan membandingkan waktu eksekusi antara bahasa pemrograman C# dan Golang. Hasil penelitian menunjukkan bahwa waktu eksekusi Golang lebih singkat pada kedua algoritma dibandingkan dengan C#, yang membutuhkan waktu lebih lama dalam proses eksekusi. Temuan ini memberikan gambaran mengenai efisiensi eksekusi kode dan dapat menjadi pertimbangan dalam pemilihan bahasa pemrograman yang optimal.

Hak Cipta Dilindungi Undang-Undang

Hak Cipta Dilindungi Undang-Undang

Institut Bisnis dan Informatika Kwik Kian Gie

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik dan tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar IBIKKG.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IBIKKG.