

# Buku Catatan Bootcamp

Rizky Denianto<sup>1</sup>

Desember 2023

<sup>1</sup>Dibuat dengan cinta dan kasih sayang.

Saya ingin mengucapkan terima kasih kepada diri sendiri yang telah memiliki niat dan usaha dalam penulisan buku ini menggunakan L<sup>A</sup>T<sub>E</sub>X.

# Daftar Isi

<b>1</b>	<b>Pelatihan <i>SQL Server</i></b>	<b>5</b>
<b>2</b>	<b>Pelatihan .NET</b>	<b>7</b>
2.1	Pembuatan Proyek . . . . .	7
2.2	Penulisan Kode C# . . . . .	7
2.2.1	Mencetak string di <i>Command Line Interface (CLI)</i> . . . .	7
2.2.2	Tipe data . . . . .	7
2.2.3	Manipulasi string . . . . .	8
2.2.4	Manipulasi tanggal . . . . .	8
2.2.5	Manipulasi angka . . . . .	8
2.2.6	Tipe data yang dapat <i>null</i> . . . . .	9
2.2.7	Kondisi IF . . . . .	9
2.2.8	Switch Case . . . . .	10
2.2.9	Seleksi . . . . .	10
2.2.10	Perulangan FOR . . . . .	10
2.2.11	Perulangan WHILE . . . . .	10
2.2.12	Susunan/Array Satu Dimensi) . . . . .	14
2.2.13	Susunan/Array Banyak Dimensi . . . . .	14
2.2.14	Susunan/Array Bergerigi ( <i>Jagged Array</i> ) . . . . .	14
2.2.15	List . . . . .	15
2.2.16	ArrayList . . . . .	16
2.2.17	Dictionary . . . . .	16
2.2.18	Fungsi/Metode ( <i>Method</i> ) . . . . .	16
2.2.19	Fungsi/Metode Overloading . . . . .	20
2.2.20	Dasar OOP . . . . .	20



Bagian 1

Pelatihan *SQL Server*



## Bagian 2

# Pelatihan .NET

### 2.1 Pembuatan Proyek

---

```
dotnet new sln --name <jalur-berkas>

dotnet new console --language "C#" --framework net6.0
    --use-program-main --output <jalur-berkas>

dotnet sln <jalur-berkas-.sln> add <jalur-berkas-.csproj>
```

---

### 2.2 Penulisan Kode C#

#### 2.2.1 Mencetak string di *Command Line Interface (CLI)*

---

```
Console.WriteLine("Hello, World!");
```

---

#### 2.2.2 Tipe data

---

```
int var = 23;
decimal var = 232000;
double var = 0.2;
float var = 0.2;
string var = "Ini string";
char var = 'o';
bool var = false;
DateTime var = DateTime.Parse("2023-12-27 20:30:11");
```

---

### 2.2.3 Manipulasi string

---

```
string namaDepan = "Rizky";
string namaBelakang = "Denianto";
string namaPanjang = namaDepan + ' ' + namaBelakang;

StringBuilder stringBuilder = new StringBuilder();
stringBuilder.Append(namaDepan);
stringBuilder.Append(' ');
stringBuilder.Append(namaBelakang);
namaPanjang = stringBuilder.ToString();

namaPanjang = string.Format("{0} {1}", namaDepan, namaBelakang);

namaPanjang = $"{namaDepan} {namaBelakang}";

int panjangString = namaPanjang.Length;
string trimString = namaPanjang.Trim();
string potongString = namaPanjang.Substring(0, 4);
string kapitalString = namaPanjang.ToUpper();
string kecilString = namaPanjang.ToLower();
int indeksKarakter = namaPanjang.IndexOf('r');
```

---

### 2.2.4 Manipulasi tanggal

---

```
DateTime waktudanTanggal = DateTime.Now;
DateTime minimal = DateTime.MinValue;
DateTime maksimal = DateTime.MaxValue;

int tahun = waktudanTanggal.Year;
int bulan = waktudanTanggal.Month;
int hari = waktudanTanggal.Day;
int jam = waktudanTanggal.Hour;

waktudanTanggal = DateTime.Parse("2023-12-05 13:30:00");
string stringWaktudanTanggal = waktudanTanggal.ToString(
    "dd MM yyyy HH:mm:ss",
    CultureInfo.CreateSpecificCulture("id-ID")
);

DateTime semingguLagi = waktudanTanggal.AddDays(7);

TimeSpan rentangWaktu = waktudanTanggal - semingguLagi;
```

---

### 2.2.5 Manipulasi angka



---

```
decimal harga = 20_000;
string stringHarga = harga.ToString(
    "C3",
    CultureInfo.CreateSpecificCulture("id-ID")
);

double diskon = 0.25;
string stringDiskon = diskon.ToString("P0");
```

---

### 2.2.6 Tipe data yang dapat *null*

---

```
int? angka = 0;

bool adaIsiAngka = angka.HasValue;
int isiAngka = angka.Value; // Tidak bisa dicetak jika isinya null
int isiAwalAngka = angka.GetValueOrDefault(0);
```

---

### Latihan

---

```
Console.WriteLine("Masukkan operator:");
string op = Console.ReadLine();
Console.WriteLine("Masukkan angka pertama:");
string angka = Console.ReadLine();
Console.WriteLine("Masukkan angka kedua:");
string angka2 = Console.ReadLine();

double hasil = double.Parse(angka + op + angka2);
Console.WriteLine(hasil);
```

---

### 2.2.7 Kondisi IF

---

```
int kondisi = 0;

if (kondisi == 0){
    Console.WriteLine("Nol");
} else if (kondisi == 1) {
    Console.WriteLine("Satu");
} else {
    Console.WriteLine("Dua");
}
```

---

### 2.2.8 Switch Case

---

```
int kondisi = 0;

switch (kondisi)
{
    case 0:
        Console.WriteLine("Nol");
        break;
    case 1:
        Console.WriteLine("Satu");
        break;
    case 2:
        Console.WriteLine("Dua");
        break;
    default:
        Console.WriteLine("Lainnya");
        break;
}
```

---

### 2.2.9 Seleksi

---

```
int kondisi = 0;
string kondisi2 = kondisi == 0 ? "Nol" : "Lainnya";
```

---

### 2.2.10 Perulangan FOR

---

```
int kondisi = 0;
for (int i = 0; i < 8; i++)
{
    kondisi++;

    for (int j = 0; j < 8; j++){
        if (j % 2 == 0) continue;
        else if (j == 4) break;

        kondisi++;
    }
}
```

---

### 2.2.11 Perulangan WHILE

---

```
int kondisi = 0;
```

```
while (kondisi < 8)
{
    kondisi++;

    Console.WriteLine("While " + kondisi);
}

do
{
    kondisi++;

    Console.WriteLine("Do While " + kondisi);
}
while (kondisi < 16);
```

---

## Latihan

---

```
while (true)
{
    Console.WriteLine("Masukkan kalimat:");
    string kalimat = Console.ReadLine();
    bool valid = true;

    int pdKalimat = kalimat.Length;
    for (int i = 0; i < pdKalimat; i++)
    {
        if (Int32.TryParse(kalimat[i].ToString(), out int _))
        {
            valid = false;
            break;
        }
    }

    if (valid)
    {
        Console.WriteLine("=====");
        Console.WriteLine("Camel Case");
        string camelCase = "";
        bool kapital = false;
        for (int i = 0; i < pdKalimat; i++)
        {
            if (kalimat[i] != ' ') kapital = false;
            else
            {
                kapital = true;
                i++;
            }
        }
    }
}
```

```
        if (!kapital) camelCase += kalimat[i];
        else camelCase += kalimat[i].ToString().ToUpper();
    }
    Console.WriteLine(camelCase);

    Console.WriteLine("Pascal Case");
    string pascalCase = "";
    bool awal = true;
    kapital = false;
    for (int i = 0; i < pdKalimat; i++)
    {
        if (kalimat[i] != ' ') kapital = false;
        else
        {
            kapital = true;
            i++;
        }

        if (!kapital && !awal) pascalCase += kalimat[i];
        else
        {
            pascalCase += kalimat[i].ToString().ToUpper();
            awal = false;
        }
    }

    Console.WriteLine(pascalCase);

    Console.WriteLine("Snake Case");
    string snakeCase = "";
    for (int i = 0; i < pdKalimat; i++)
    {
        if (kalimat[i] == ' ')
        {
            snakeCase += '_';
            continue;
        }

        snakeCase += kalimat[i];
    }
    Console.WriteLine(snakeCase + '\n');
}
```

---

## Latihan

---

```
while (true)
{
    Console.WriteLine("Masukkan kalimat:");
    string kalimat = Console.ReadLine();
    bool valid = true;

    int pdKalimat = kalimat.Length;
    for (int i = 0; i < pdKalimat; i++)
    {
        if (Int32.TryParse(kalimat[i].ToString(), out int _))
        {
            valid = false;
            break;
        }
    }

    if (valid)
    {
        string hurufKonsonan = "";
        string hurufVokal = "";

        for (int i = 0; i < pdKalimat; i++)
        {
            if (kalimat[i] == ' ')
            {
                hurufKonsonan += ' ';
                hurufVokal += ' ';
            }
            else
            {
                if (
                    kalimat[i].ToString().Contains('a')
                    || kalimat[i].ToString().Contains('i')
                    || kalimat[i].ToString().Contains('u')
                    || kalimat[i].ToString().Contains('e')
                    || kalimat[i].ToString().Contains('o')
                )
                {
                    hurufVokal += kalimat[i];
                }
                else
                {
                    hurufKonsonan += kalimat[i];
                }
            }
        }

        Console.WriteLine("=====");
        Console.WriteLine("Huruf konsonan:");
        Console.WriteLine(hurufKonsonan);
    }
}
```

```
        Console.WriteLine("Huruf vokal:");
        Console.WriteLine(hurufVokal + '\n');
    }
}
```

---

### 2.2.12 Susunan/Array Satu Dimensi)

---

```
int[] susunan = new int[8];
int[] susunan = new int[]{
    0,
    1,
    2,
    3,
    4
};
int[] susunan = {
    0,
    1,
    2,
    3,
    4
};

susunan[5] = 0;

Console.WriteLine(susunan.Length);
```

---

### 2.2.13 Susunan/Array Banyak Dimensi

---

```
int[,] susunan = new int[2, 3];
int[,] susunan = {{0, 1, 2}, {2, 3, 4}, {7, 8, 9}};
object[,] susunan = { { "1", 2, 3, "C" }, { "Aku", 5, 6, "B" }, {
    "Kamu", 8, 9, "A" } };

int dimensiSusunan = susunan.Rank;
Console.WriteLine(dimensiSusunan);

foreach (object o in susunan) Console.WriteLine(o);
```

---

### 2.2.14 Susunan/Array Bergerigi (*Jagged Array*)

---

```
int[][] susunan = new int[2][];
susunan[0] = new int[2];
```

```

susunan[0][0] = 0;
susunan[0][1] = 0;
susunan[1] = new int[2];
susunan[1][0] = 0;
susunan[1][1] = 0;

int[] [] susunan = new int[3] [] {
    new int[2]{0, 2},
    new int[1]{2},
    new int[3]{0, 8, 9}
};

foreach (int i in susunan[0]) Console.WriteLine(i);
foreach (int i in susunan[1]) Console.WriteLine(i);

```

---

### 2.2.15 List

---

```

List<string> daftar = new List<string>();
daftar.Add("Harimau");
daftar.Add("Singa");
daftar.Add("Kucing");
daftar.Insert(0, "Kucing");
daftar.Insert(4, "Kucing");
daftar.Remove("Harimau");
daftar.RemoveAt(0);
daftar.Reverse();
daftar.Clear();
// daftar.RemoveAll();
daftar.AddRange(daftar); // Gabung List<>

List<string> daftar = new List<string>(){
    "Harimau",
    "Singa",
    "Kucing"
};

List<string[]> daftar = new List<string[]>(){
    new string[]{
        "Harimau",
        "Ikan",
        "Kura-Kura",
        "Kupu-Kupu",
        "Undur-Undur"
    },
    new string[]{
        "Mawar",
        "Melati",
        "Poon",
    }
};

```

```
        "Terompet",  
        "Matahari"  
    }  
};  
  
Console.WriteLine(daftar.Count);  
  
foreach (string d in daftar) Console.WriteLine(d);
```

---

### 2.2.16 ArrayList

```
ArrayList daftar = new ArrayList();  
daftar.Add("Banyak");  
daftar.Add(324);  
daftar.Add(false);  
daftar.Insert(0, 20_000);  
  
Console.WriteLine(daftar.IsFixedSize);  
foreach(object d in daftar) Console.WriteLine(d);
```

---

### 2.2.17 Dictionary

```
Dictionary<int, string> daftar = new Dictionary<int, string>();  
daftar.Add(1, "Harimau");  
daftar.Add(2, "Singa");  
daftar.Add(3, "Macan");  
  
foreach (KeyValuePair<int, string> d in daftar)  
{  
    Console.WriteLine(d.Key);  
    Console.WriteLine(d.Value);  
}  
foreach (var d in daftar)  
{  
    Console.WriteLine(d.Key);  
    Console.WriteLine(d.Value);  
}
```

---

### 2.2.18 Fungsi/Metode (*Method*)

```
public static void FungsiVoid(string teks, string[] susunanTeks,  
    params string[] susunanTeks2)  
{
```



```

        Console.WriteLine("Ini fungsi " + teks);
        foreach (string t in susunanTeks) Console.WriteLine("Ini fungsi "
            + t);
        foreach (string t in susunanTeks2) Console.WriteLine("Ini fungsi "
            + t);
    }

    public static void FungsiVoid2(string varTidakWajib = "tidak wajib
        diisi")
    {
        Console.WriteLine(varTidakWajib);
    }

    public static string FungsiString(string teks)
    {
        return teks + teks;
    }

    public static string FungsiEstafet()
    {
        return FungsiString("Fungsi estafet dijalankan ");
    }

    public static void FungsiRekursif(int i = 0)
    {
        i++;
        Console.WriteLine(i);

        if (i >= 100) return;
        else FungsiRekursif(i);
    }

```

---

## Latihan

---

```

    public static void Menu(string[] daftarSnack, string[] daftarMinuman,
        int[] hasilPilihan)
    {
        if (hasilPilihan[0] != 0 && hasilPilihan[1] != 0)
        {
            Console.WriteLine($"Snack dan minuman yang Anda pilih adalah
                {daftarSnack[hasilPilihan[0] - 1]} dan
                {daftarMinuman[hasilPilihan[1] - 1]}");

            Console.WriteLine("Pilih ulang? (y/t)");
            string pilihUlang = Console.ReadLine();
            if (pilihUlang.ToLower() == "t") Console.WriteLine("Kelar");
            else if (pilihUlang.ToLower() == "y") Menu(daftarSnack,
                daftarMinuman, new int[] { 0, 0 });
        }
    }

```

```

else
{
    Console.WriteLine("Pilihan tidak valid\n");
    Menu(daftarSnack, daftarMinuman, hasilPilihan);
}
}
else
{
    try
    {
        Console.WriteLine("Pilih Menu:");
        Console.WriteLine("1. Snack");
        Console.WriteLine("2. Minuman");
        int pilihan = Int32.Parse(Console.ReadLine());

        if (pilihan <= 0 || pilihan > 2)
        {
            Console.WriteLine("Pilihan tidak valid\n");
            Menu(daftarSnack, daftarMinuman, hasilPilihan);
        }
        else
        {
            if (pilihan == 1)
            {
                int n = Snack(daftarSnack);
                hasilPilihan[0] = n != 0 ? n : hasilPilihan[0];
            }
            else if (pilihan == 2)
            {
                int n = Minuman(daftarMinuman);
                hasilPilihan[1] = n != 0 ? n : hasilPilihan[1];
            }
            Menu(daftarSnack, daftarMinuman, hasilPilihan);
        }
    }
    catch
    {
        Console.WriteLine("Pilihan tidak valid\n");
        Menu(daftarSnack, daftarMinuman, hasilPilihan);
    }
}
}

public static int Snack(string[] daftarSnack)
{
    try
    {
        Console.WriteLine("Pilih Snack:");
        for (int i = 0; i < daftarSnack.Length; i++)
            Console.WriteLine($"{i + 1}. {daftarSnack[i]}");
    }
    catch
    {
        Console.WriteLine("Pilihan tidak valid\n");
        Menu(daftarSnack, daftarMinuman, hasilPilihan);
    }
}

```

```
int pilihan = Int32.Parse(Console.ReadLine());

if (pilihan > 0 && pilihan < daftarSnack.Length)
{
    Console.WriteLine($"Snack yang anda pilih adalah
        {daftarSnack[pilihan - 1]}\n");
    return pilihan;
}
else
{
    if (pilihan != daftarSnack.Length)
    {
        Console.WriteLine("Pilihan tidak valid\n");
        Snack(daftarSnack);
    }
    return 0;
}
}
catch
{
    Console.WriteLine("Pilihan tidak valid\n");
    Snack(daftarSnack);
    return 0;
}
}

public static int Minuman(string[] daftarMinuman)
{
    try
    {
        Console.WriteLine("Pilih minuman:");
        for (int i = 0; i < daftarMinuman.Length; i++)
            Console.WriteLine($"{i + 1}. {daftarMinuman[i]}");
        int pilihan = Int32.Parse(Console.ReadLine());

        if (pilihan > 0 && pilihan < daftarMinuman.Length)
        {
            Console.WriteLine($"Minuman yang anda pilih adalah
                {daftarMinuman[pilihan - 1]}\n");
            return pilihan;
        }
        else
        {
            if (pilihan != daftarMinuman.Length)
            {
                Console.WriteLine("Pilihan tidak valid\n");
                Minuman(daftarMinuman);
            }
            return 0;
        }
    }
}
```

```
    }  
    catch  
    {  
        Console.WriteLine("Pilihan tidak valid\n");  
        Minuman(daftarMinuman);  
        return 0;  
    }  
}
```

---

### 2.2.19 Fungsi/Metode Overloading

---

```
public static string Fungsi()  
{  
    return "";  
}  
public static string Fungsi(string a)  
{  
    return "";  
}  
public static int Fungsi(int a)  
{  
    return a;  
}  
public static decimal Fungsi(decimal a)  
{  
    return a;  
}  
public static int Fungsi(int a, int b)  
{  
    return a + b / a - b;  
}
```

---

### 2.2.20 Dasar OOP

---

```
namespace ChapterFive;  
  
public class Karyawan  
{  
    public string Nama = "";  
  
    private string _kelamin = "";  
    public string Kelamin  
    {  
        get => _kelamin;  
        set => _kelamin = value;  
    }  
}
```

```
public string Alamat { get; set; } = "";

public string Alamat2 { get; } = "";
public Karyawan(string alamat, string alamat2)
{
    this.Alatam = alamat;
    this.Alatam2 = alamat2;
}
public Karyawan(string alamat2)
{
    this.Alatam2 = alamat2;
}

private DateTime _lahir;

public DateTime Lahir
{
    get
    {
        return this._lahir;
    }
    set
    {
        this._lahir = value;
    }
}

public string AmbilNama()
{
    return this>Nama;
}

public void AturNama(string nama)
{
    this>Nama = nama;
}
}
```

---