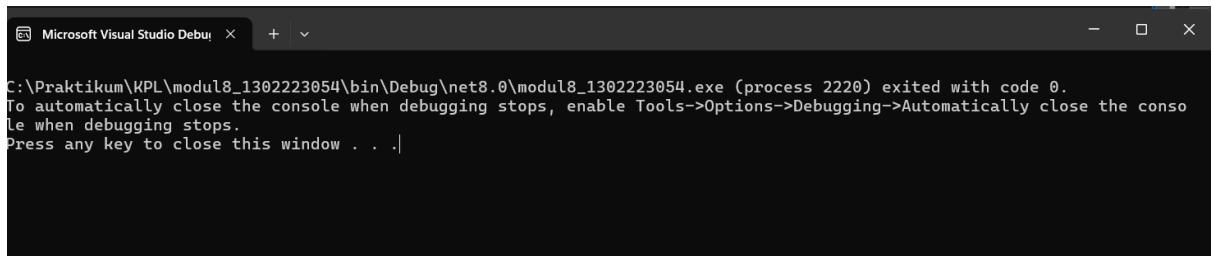


## 1. Link Github Repository

[MuhammadNurShodiq/modul8\\_1302223054 \(github.com\)](https://github.com/MuhammadNurShodiq/modul8_1302223054)

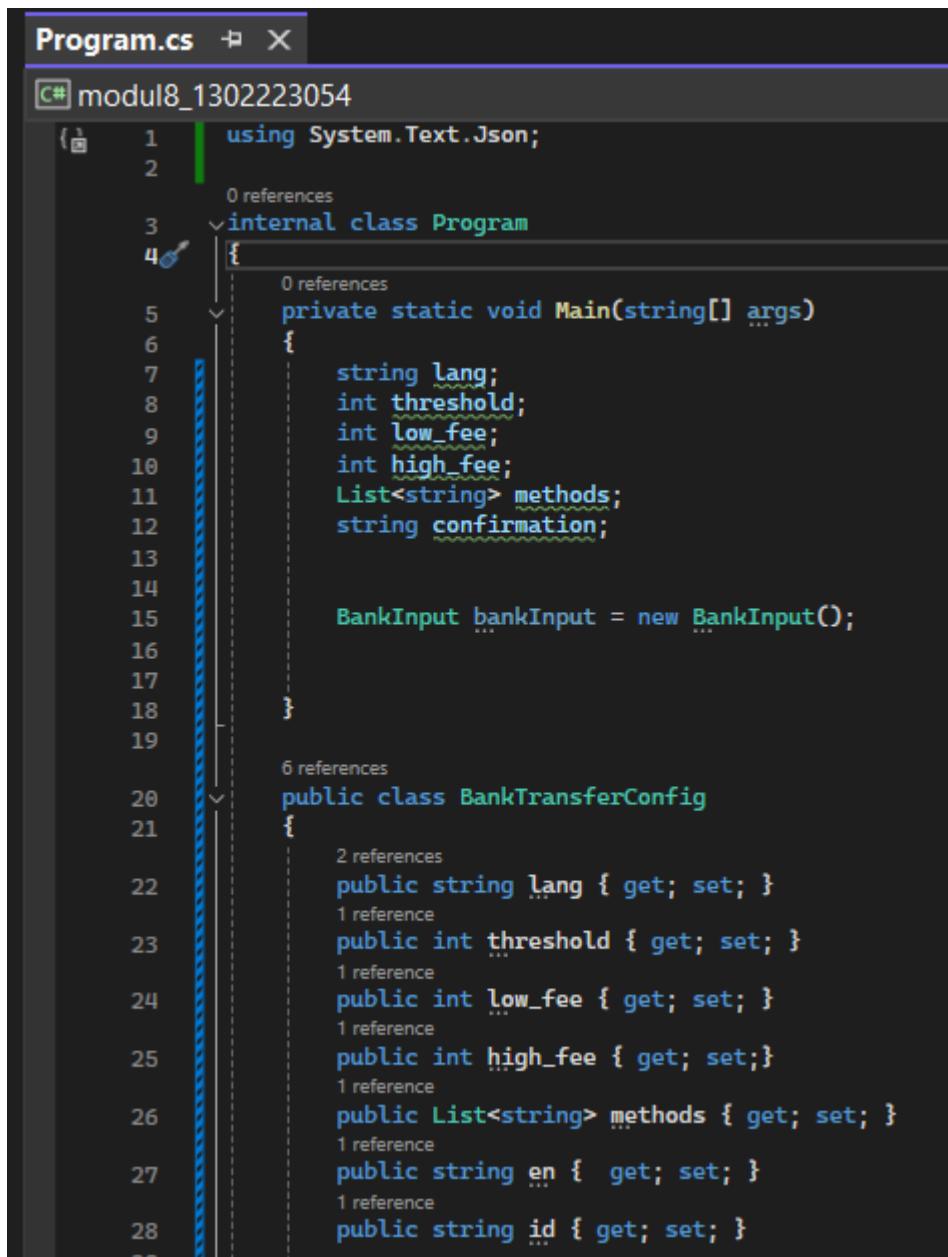
## 2. Hasil Run



The screenshot shows the Microsoft Visual Studio Debug console window. The output text is as follows:

```
C:\Praktikum\KPL\modul8_1302223054\bin\Debug\net8.0\modul8_1302223054.exe (process 2220) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .|
```

## 3. Code



The screenshot shows the Microsoft Visual Studio Code editor with the file "Program.cs" open. The code is as follows:

```
1  using System.Text.Json;
2
3  internal class Program
4  {
5      private static void Main(string[] args)
6      {
7          string lang;
8          int threshold;
9          int low_fee;
10         int high_fee;
11         List<string> methods;
12         string confirmation;
13
14
15         BankInput bankInput = new BankInput();
16
17
18     }
19
20     public class BankTransferConfig
21     {
22         public string lang { get; set; }
23         public int threshold { get; set; }
24         public int low_fee { get; set; }
25         public int high_fee { get; set; }
26         public List<string> methods { get; set; }
27         public string en { get; set; }
28         public string id { get; set; }
29     }
}
```

The image shows a code editor with two tabs open. The top tab is titled "Program.cs" and contains the following C# code:

```
1 reference
private BankTransferConfig ReadConfig()
{
    string jsonData = File.ReadAllText(filepath);
    bankTransferConfig = JsonSerializer.Deserialize<BankTransferConfig>(jsonData);
    return bankTransferConfig;
}

1 reference
private void SetDefault()
{
    bankTransferConfig = new BankTransferConfig("en", 25000000, 6500, 15000, ["RTO (real-time)", "SKN", "RTGS", "BI FAST"], "yes", "ya");
}

1 reference
private void WriteNewConfig()
{
    JsonSerializerOptions options = new JsonSerializerOptions()
    {
        WriteIndented = true,
    };

    string JsonString = JsonSerializer.Serialize(bankTransferConfig, options);
    File.WriteAllText(filepath, JsonString);
}

0 references
public void UbahData (string dataBaru)
{
    bool dataValid = (dataBaru == "en" || dataBaru == "id");

    if (dataBaru == null || !dataValid)
    {
        throw new ArgumentException();
    }

    if (dataValid)
    {
        bankTransferConfig.lang = dataBaru;
    }
}
```

The bottom tab is titled "modul8\_1302223054" and contains the following C# code:

```
28
29
30     public string id { get; set; }
31
32     0 references
33     public BankTransferConfig() { }
34
35     1 reference
36     public BankTransferConfig(string lang, int threshold, int low_fee, int high_fee, List<string> methods, string en, string id)
37     {
38         this.lang = lang;
39         this.threshold = threshold;
40         this.low_fee = low_fee;
41         this.high_fee = high_fee;
42         this.methods = methods;
43         this.en = en;
44         this.id = id;
45     }
46
47     3 references
48     public class BankInput
49     {
50         public BankTransferConfig bankTransferConfig;
51         public const String filepath = @"bank_transfer_config.json";
52
53         1 reference
54         public BankInput()
55         {
56             try
57             {
58                 ReadConfig();
59             }
60             catch (Exception)
61             {
62                 SetDefault();
63                 WriteNewConfig();
64             }
65         }
66
67         1 reference
68         if (dataValid)
69         {
70             bankTransferConfig.lang = dataBaru;
71
72             JsonSerializerOptions jsonSerializerOptions = new JsonSerializerOptions()
73             {
74                 WriteIndented = true,
75             };
76
77             string updateData = JsonSerializer.Serialize(bankTransferConfig, jsonSerializerOptions);
78             File.WriteAllText (filepath, updateData);
79         }
80     }
81 }
```

#### 4. Penjelasan Kode

Kode diatas merupakan kode program dalam C# yang bertujuan untuk mengatur konfigurasi transfer bank. Kode ini mendefinisikan class BankTransferConfig yang menyimpan informasi terkait bahasa (lang), ambang batas transfer (threshold), biaya minimum (low\_fee), biaya maksimum (high\_fee), metode transfer (methods), dan konfirmasi bahasa (en/id). Class BankInput kemudian digunakan untuk membaca konfigurasi yang tersimpan pada file bank\_transfer\_config.json. Jika file tersebut tidak ditemukan, maka konfigurasi default akan dibuat dan ditulis ke dalam file. Terakhir, fungsi UbahData pada class BankInput memungkinkan pengguna untuk mengubah bahasa (en/id) yang digunakan pada program.