

**TUGAS JURNAL  
MODUL 15**



**Disusun Oleh :**

**Pradana Argo Pangestu – 2311104079**

**Kelas : SE-07-02**

**Dosen :**

**Yudha Islami Sulistya**

**PROGRAM STUDI SOFTWARE ENGINEERING  
DIREKTORAT KAMPUS PURWOKERTO  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## I. Link Github

- [https://github.com/Pradana123/KPL\\_PradanaAP\\_2311104079/tree/main/15\\_Review\\_Tugas\\_Besar](https://github.com/Pradana123/KPL_PradanaAP_2311104079/tree/main/15_Review_Tugas_Besar)

## II. Alur Pengerjaan

### bank\_transfer\_config.json

```
1. {
2.   "lang": "id",
3.   "transfer": {
4.     "threshold": 25000000,
5.     "low_fee": 6500,
6.     "high_fee": 15000
7.   },
8.   "methods": ["RTO (real-time)", "SKN", "RTGS", "BI FAST"],
9.   "confirmation": {
10.    "en": "yes",
11.    "id": "ya"
12.  }
13. }
```

## Validasi Input

Validasi dilakukan untuk memastikan **username dan password memenuhi aturan keamanan**.

kode:

```
1. if (username.Length < 8 || username.Length > 20 || !username.All(char.IsLetter))
2.   return false;
3.
4. if (password.Length < 8 || password.Length > 20 ||
5.     !password.Any(char.IsDigit) ||
6.     !password.Any(c => "!@#$%^&*" .Contains(c)) ||
7.     password.Contains(username))
8.   return false;
```

### Penjelasan:

- Username:
  - Panjang 8–20 karakter.
  - Hanya huruf alfabet (`char.IsLetter`).
- Password:
  - Panjang 8–20 karakter.
  - Harus mengandung **minimal 1 angka** (`char.IsDigit`).

- Harus mengandung **minimal 1 karakter unik** (!@#\$\$%^&\*).
  - **Tidak boleh mengandung bagian dari username** (untuk mencegah tebakan sederhana).
- 

## 2. Hash Password

Password tidak disimpan langsung (plain text), melainkan di-*hash* terlebih dahulu menggunakan **SHA256** agar lebih aman.

kode (PasswordHelper.cs):

```
1. public static string Hash(string password)
2. {
3.     using (SHA256 sha = SHA256.Create())
4.     {
5.         byte[] bytes = sha.ComputeHash(Encoding.UTF8.GetBytes(password));
6.         return Convert.ToBase64String(bytes);
7.     }
8. }
```

### Penjelasan:

- Mengubah password menjadi bentuk hash (irreversible).
  - Hash disimpan ke file JSON, sehingga meskipun file terbuka, password asli tidak diketahui.
  - Ini adalah bagian dari **secure coding practices**.
- 

## 3. Simpan dan Baca dari JSON

Data user (username dan password yang sudah di-hash) disimpan dalam file `users.json`.

penyimpanan (AuthService.cs):

```
1. private void Save()
2. {
3.     string json = JsonSerializer.Serialize(users, new JsonSerializerOptions { WriteIndented = true });
4.     File.WriteAllText(FilePath, json);
5. }
```

Contoh pembacaan:

```
1. if (File.Exists(FilePath))
2. {
3.     string json = File.ReadAllText(FilePath);
```

```
4. users = JsonSerializer.Deserialize<list<user>>(json) ?? new List<user>();  
5. }
```

### Penjelasan:

- Data user disimpan dalam format JSON agar bisa dibaca/diedit kapan saja.
- Menggunakan System.Text.Json untuk serialisasi (konversi object  $\rightleftharpoons$  string JSON).
- File akan berisi daftar user dengan username dan password hash.

---

## 4. Cek Login Valid

Ketika user login, program memverifikasi apakah:

- Username ditemukan
- Password yang dimasukkan setelah di-hash **cocok** dengan hash yang disimpan

Contoh kode:

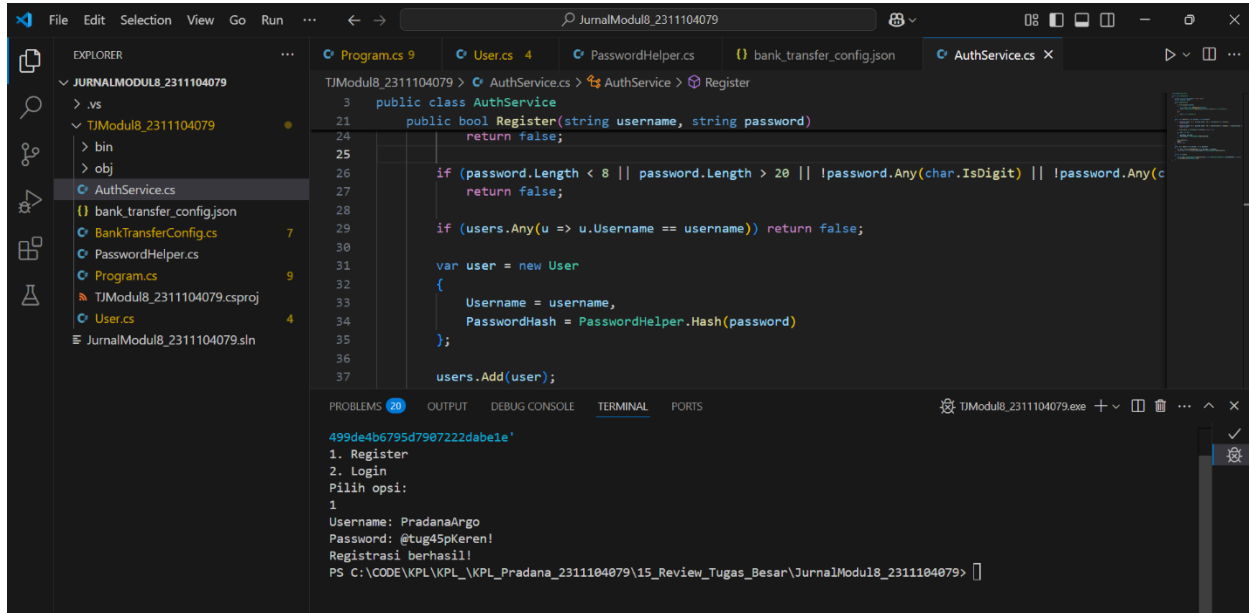
```
1. public bool Login(string username, string password)  
2. {  
3.     var user = users.FirstOrDefault(u => u.Username == username);  
4.     return user != null && user.PasswordHash == PasswordHelper.Hash(password);  
5. }
```

### Penjelasan:

- Jika username tidak ditemukan  $\rightarrow$  login gagal.
- Jika ditemukan, password di-hash lalu dibandingkan dengan hash tersimpan.
- Jika cocok  $\rightarrow$  login berhasil, jika tidak  $\rightarrow$  gagal.

### III. Output

#### Regis Berhasil

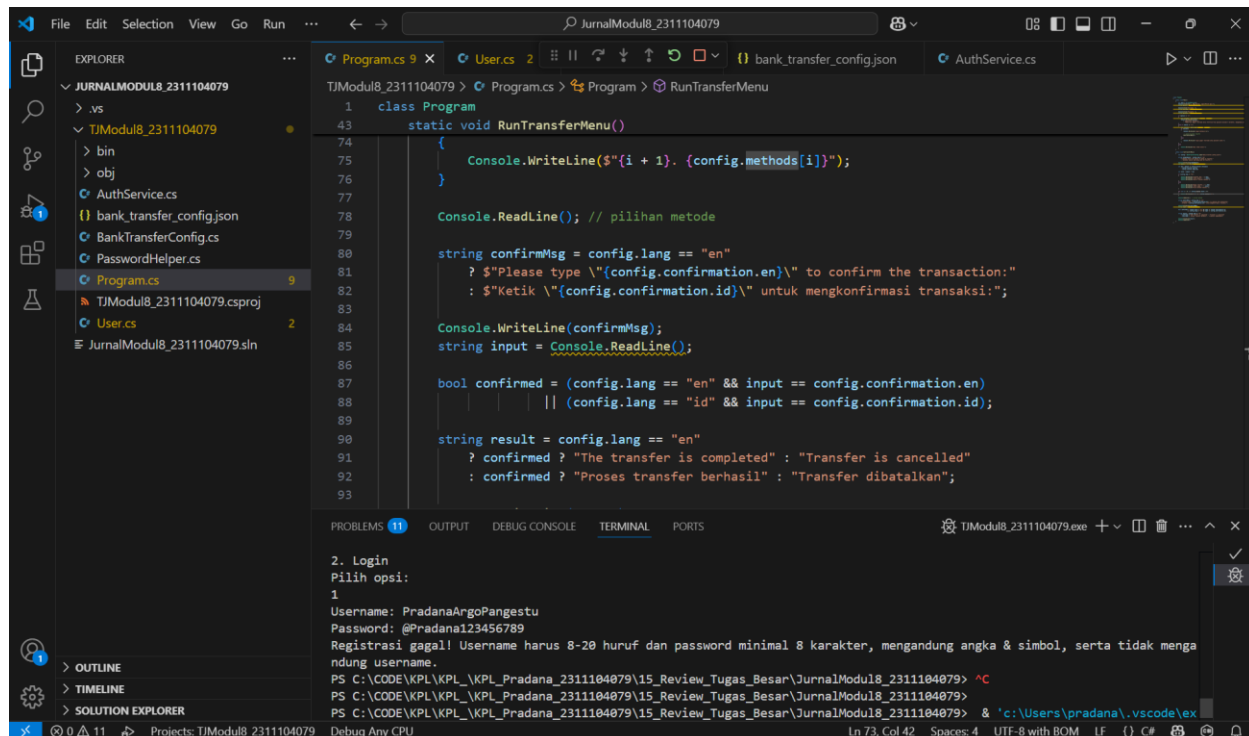


The screenshot shows the Visual Studio IDE with the `AuthService.cs` file open. The `Register` method is visible, which checks password length and complexity, verifies if the username already exists, and then adds the new user to the `users` list. The terminal output shows the following sequence of events:

```

499de4b6795d7907222dabe1e'
1. Register
2. Login
Pilih opsi:
1
Username: PradanaArgo
Password: @tug4SpKeren!
Registrasi berhasil!!
PS C:\CODE\KPL\KPL_Pradana_2311104079\15_Review_Tugas_Besar\JurnalModul8_2311104079>
  
```

#### Regis gagal

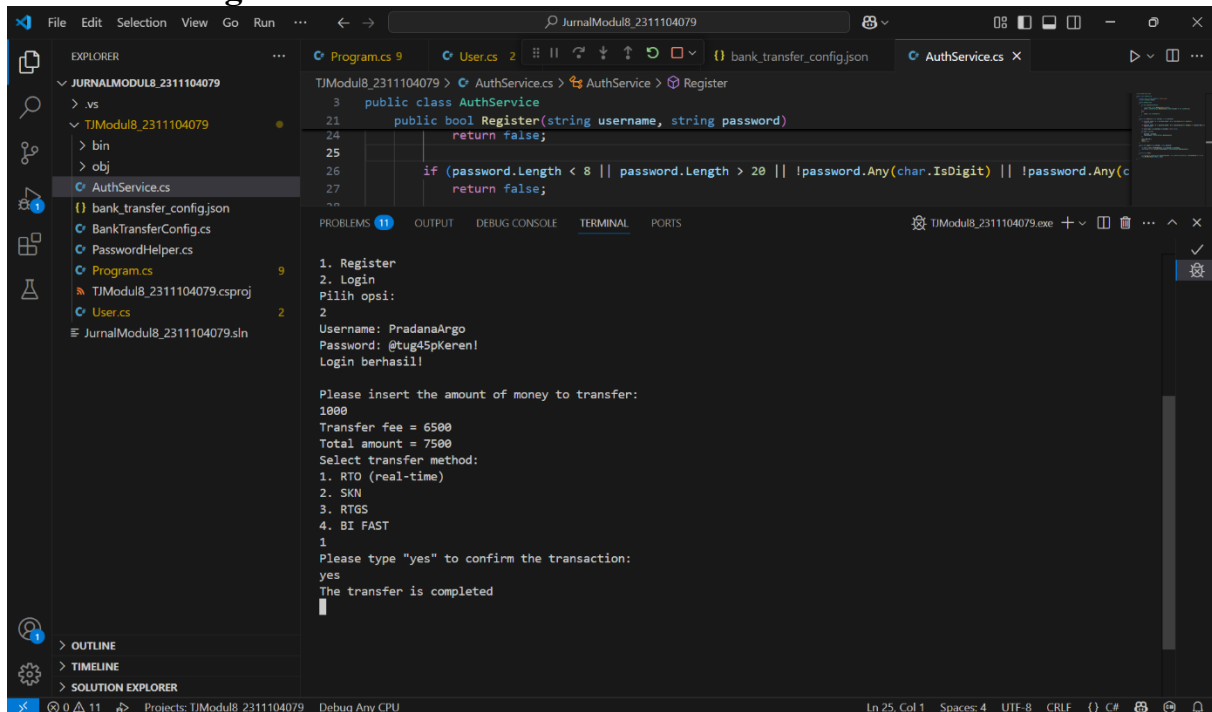


The screenshot shows the Visual Studio IDE with the `Program.cs` file open. The `RunTransferMenu` method is visible, which handles the registration process. The terminal output shows the following sequence of events:

```

2. Login
Pilih opsi:
1
Username: PradanaArgoPangestu
Password: @Pradana123456789
Registrasi gagal! Username harus 8-20 huruf dan password minimal 8 karakter, mengandung angka & simbol, serta tidak mengandung username.
PS C:\CODE\KPL\KPL_Pradana_2311104079\15_Review_Tugas_Besar\JurnalModul8_2311104079> ^C
PS C:\CODE\KPL\KPL_Pradana_2311104079\15_Review_Tugas_Besar\JurnalModul8_2311104079>
PS C:\CODE\KPL\KPL_Pradana_2311104079\15_Review_Tugas_Besar\JurnalModul8_2311104079> & 'c:\Users\pradana\.vscode\ex
  
```

## Login Berhasil



```

JurnalModul8_2311104079
├── .vs
│   ├── bin
│   ├── obj
│   ├── AuthService.cs
│   ├── bank_transfer_config.json
│   ├── BankTransferConfig.cs
│   ├── PasswordHelper.cs
│   ├── Program.cs
│   ├── TJModul8_2311104079.csproj
│   ├── User.cs
│   └── JurnalModul8_2311104079.sln
├── OUTLINE
├── TIMELINE
└── SOLUTION EXPLORER

```

```

TJModul8_2311104079 > AuthService > Register
3 public class AuthService
21 public bool Register(string username, string password)
24 return false;
25
26 if (password.Length < 8 || password.Length > 20 || !password.Any(char.IsDigit) || !password.Any(c
27 return false;

```

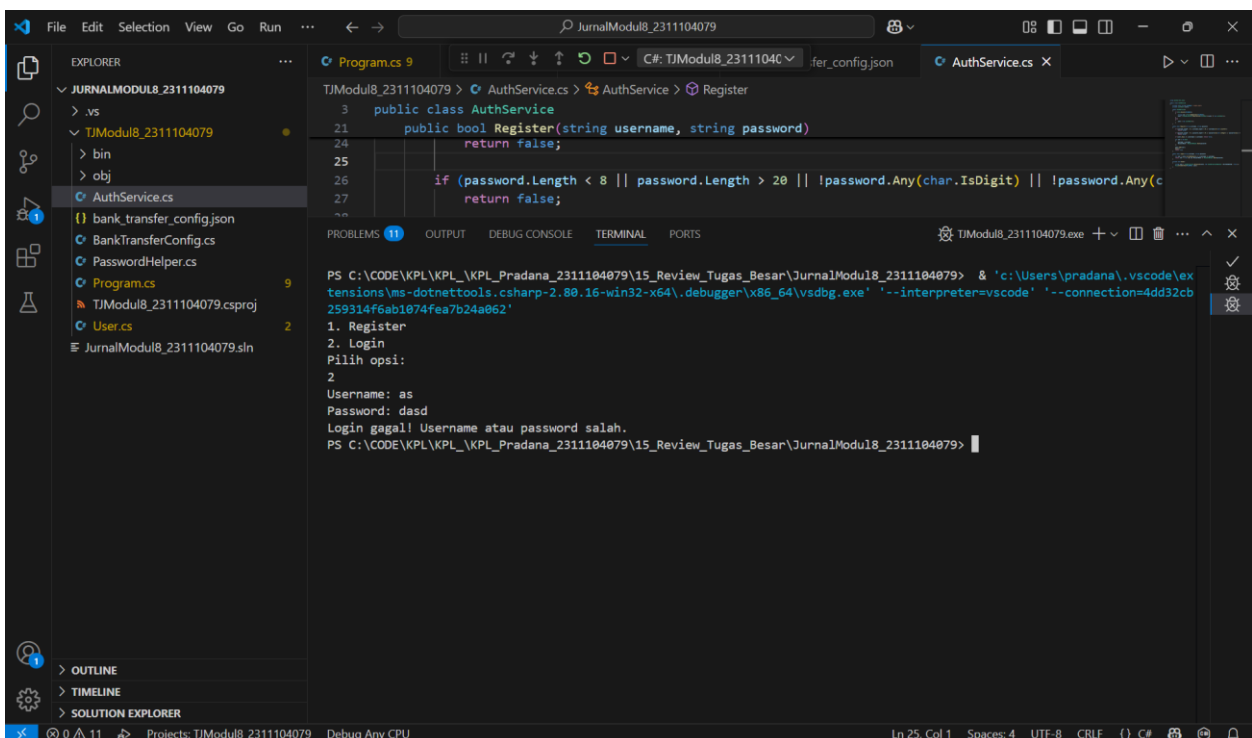
```

1. Register
2. Login
Pilih opsi:
2
Username: PradanaArgo
Password: @tug4SpKeren!
Login berhasil!

Please insert the amount of money to transfer:
1000
Transfer fee = 6500
Total amount = 7500
Select transfer method:
1. RTO (real-time)
2. SKN
3. RTGS
4. BI FAST
1
Please type "yes" to confirm the transaction:
yes
The transfer is completed

```

## Login gagal



```

JurnalModul8_2311104079
├── .vs
│   ├── bin
│   ├── obj
│   ├── AuthService.cs
│   ├── bank_transfer_config.json
│   ├── BankTransferConfig.cs
│   ├── PasswordHelper.cs
│   ├── Program.cs
│   ├── TJModul8_2311104079.csproj
│   ├── User.cs
│   └── JurnalModul8_2311104079.sln
├── OUTLINE
├── TIMELINE
└── SOLUTION EXPLORER

```

```

TJModul8_2311104079 > AuthService > Register
3 public class AuthService
21 public bool Register(string username, string password)
24 return false;
25
26 if (password.Length < 8 || password.Length > 20 || !password.Any(char.IsDigit) || !password.Any(c
27 return false;

```

```

PS C:\CODE\KPL\KPL_Pradana_2311104079\15_Review_Tugas_Besar\JurnalModul8_2311104079> & 'c:\Users\pradana\.vscode\extensions\ms-dotnettools.csharp-2.80.16-win32-x64\debugger\x86_64\vsdbg.exe' '--interpreter=vscode' '--connection=4dd32cb259314f6ab1074fea7b24a062'
1. Register
2. Login
Pilih opsi:
2
Username: as
Password: dasd
Login gagal! Username atau password salah.
PS C:\CODE\KPL\KPL_Pradana_2311104079\15_Review_Tugas_Besar\JurnalModul8_2311104079>

```

#### **IV. Kesimpulan**

Validasi Input memastikan bahwa data yang dimasukkan pengguna sesuai dengan aturan keamanan (panjang, karakter yang diperbolehkan, dan struktur password). Password Hashing menggunakan algoritma SHA256 menjamin bahwa password tidak disimpan dalam bentuk asli, sehingga lebih aman jika file bocor. Penyimpanan JSON membuat data user dapat disimpan secara persisten dalam format yang mudah dikelola dan dibaca ulang oleh program. Verifikasi Login dilakukan dengan mencocokkan hasil hash password input dengan yang ada di file, bukan password asli — ini mencegah pencurian data secara langsung.