



COACHING PROGRAM Week 3

Percabangan Perulangan

Bagian 1: Percabangan: Membuat Keputusan di Game

Bayangkan game di mana karaktermu kebal dan musuh tak bisa dikalahkan. Membosankan, kan? Percabangan membuat game-mu lebih menarik dengan memberikan kemampuan untuk mengambil keputusan berdasarkan kondisi tertentu.

- **if:** Jika kondisi terpenuhi, jalankan aksi ini.

```
...  
  
if (nyawa <= 0)  
{  
    GameOver(); // Panggil fungsi GameOver()  
}
```

- **else if:** Jika kondisi pertama tidak terpenuhi, cek kondisi lain.

```
...  
  
if (skor > 1000)  
{  
    BonusStage(); // Panggil fungsi BonusStage()  
}  
else if (skor > 500)  
{  
    PowerUp(); // Panggil fungsi PowerUp()  
}
```

COACHING PROGRAM Week 3

Percabangan Perulangan

- **else:** Jika semua kondisi tidak terpenuhi, jalankan aksi ini.

```
•••  
  
if (kunciTerambil)  
{  
    BukaPintu(); // Panggil fungsi BukaPintu()  
}  
else  
{  
    TampilkanPesan("Kunci dibutuhkan!");  
}
```

Bagian 2: Perulangan: Aksi Berulang Tanpa Henti

Dalam game, banyak aksi yang dilakukan berulang kali, seperti menggerakkan musuh, menghasilkan kotak item, atau menghitung mundur waktu. Perulangan membuat kodemu lebih efisien dan mudah dibaca.

- **for:** Ulangi aksi sejumlah kali yang ditentukan.

```
•••  
  
for (int i = 0; i < 3; i++)  
{  
    BuatMusuh(); // Panggil fungsi BuatMusuh() 3 kali  
}
```

COACHING PROGRAM Week 3

Percabangan Perulangan

- **while:** Ulangi aksi selama kondisi terpenuhi.

```
•••  
  
while (waktu > 0)  
{  
    waktu--; // Kurangi waktu setiap detik  
    UpdateWaktu(); // Panggil fungsi UpdateWaktu()  
}
```

- **do-while:** Ulangi aksi minimal sekali, lalu cek kondisi.

```
•••  
  
do  
{  
    GerakkanKarakter(); // Panggil fungsi GerakkanKarakter()  
} while (inputKeyboard != "berhenti");
```



COACHING PROGRAM Week 3

Percabangan Perulangan

Bagian 3: Fungsi: Kode yang Terstruktur dan Reusable

Fungsi adalah blok kode yang diberi nama dan dapat dipanggil berulang kali. Fungsi membuat kodemu lebih terstruktur, mudah dibaca, dan *reusable*.

- Mendeklarasikan Fungsi:

```
•••  
  
void GerakkanKarakter(float kecepatan)  
{  
    // Kode untuk menggerakkan karakter  
}
```

- Memanggil Fungsi:

```
•••  
  
GerakkanKarakter(5.0f); // Panggil fungsi dengan kecepatan 5.0
```



COACHING PROGRAM Week 3

Percabangan Perulangan

Bagian 4: *Nested Loop* dan *Nested if*: Logika yang Lebih Kompleks

- ***Nested Loop***: Perulangan di dalam perulangan. Contoh: membuat *grid* peta dalam game.

```
● ● ●

for (int i = 0; i < 10; i++)
{
    for (int j = 0; j < 10; j++)
    {
        BuatTile(i, j); // Panggil fungsi BuatTile() untuk setiap posisi
    }
}
```

- ***Nested if***: Percabangan di dalam percabangan. Contoh: cek *collision* antar objek dengan kondisi tambahan.

```
● ● ●

if (objek1.Bertabrakan(objek2))
{
    if (objek1.isPlaying)
    {
        KurangiNyawa(); // Panggil fungsi KurangiNyawa()
    }
    else
    {
        TambahSkor(); // Panggil fungsi TambahSkor()
    }
}
```



COACHING PROGRAM Week 3

Percabangan Perulangan

Tugas:

- Buatlah program C# yang menampilkan tabel perkalian dari 1 sampai 10 menggunakan *nested loop*.
- Buatlah program C# game "batu gunting kertas" sederhana yang menerima input dari *user* dan menentukan pemenangnya menggunakan percabangan.

Semoga materi ini membuatmu semakin mahir dalam C# dan siap membangun logika game yang lebih kompleks! 🎮