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.



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



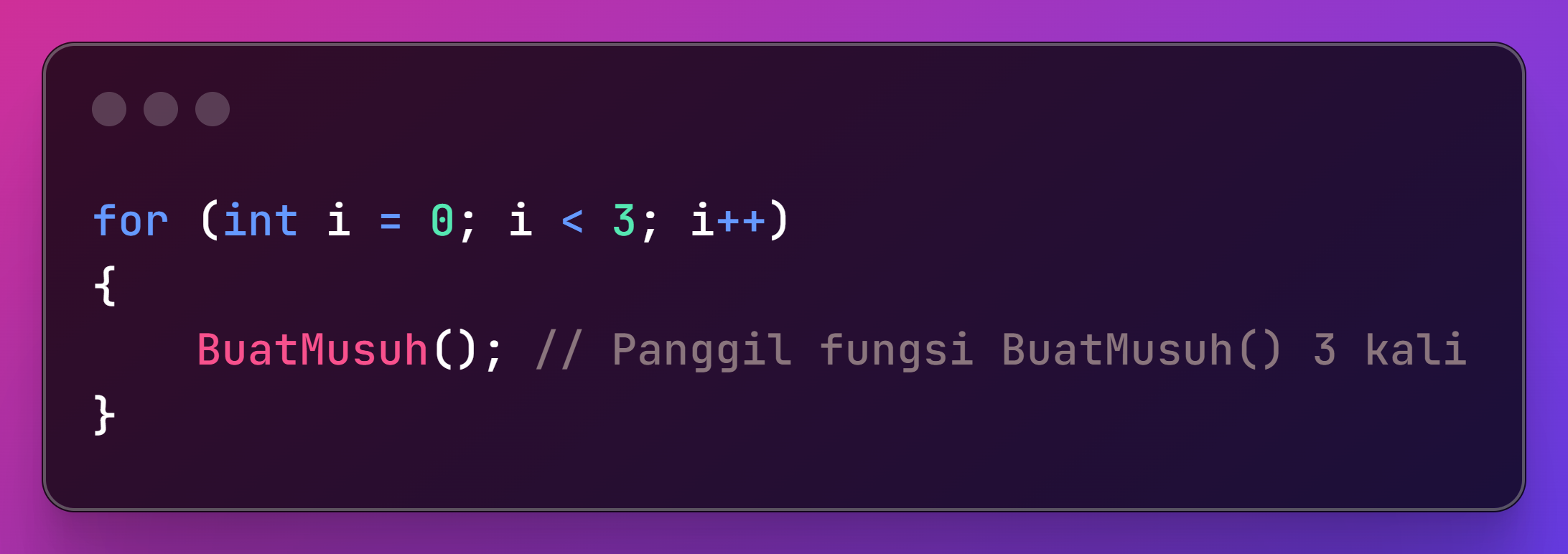
* **else**: Jika semua kondisi tidak terpenuhi, jalankan aksi ini.



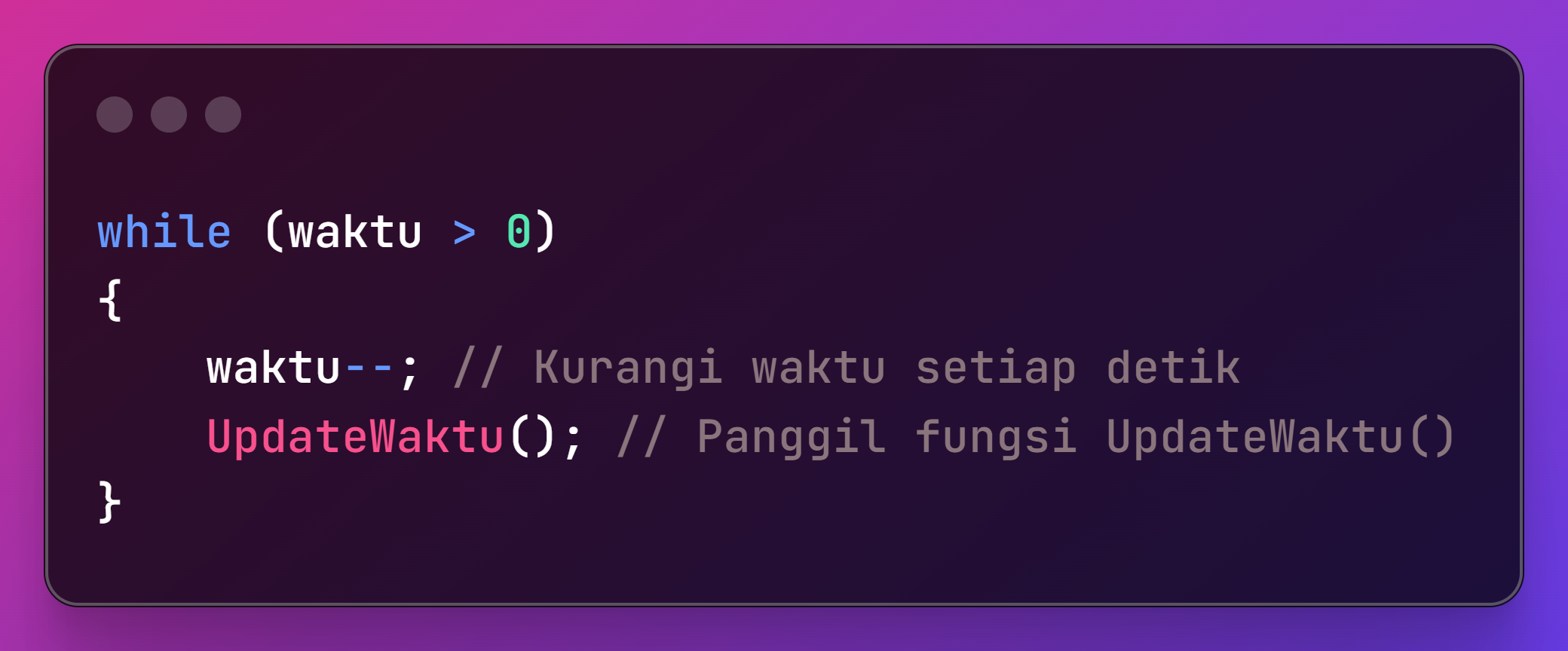
Bagian 2: Perulangan: Aksi Berulang Tanpa Henti

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.



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



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

****

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:**



* **Memanggil Fungsi:**

****

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

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



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



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! 🎮