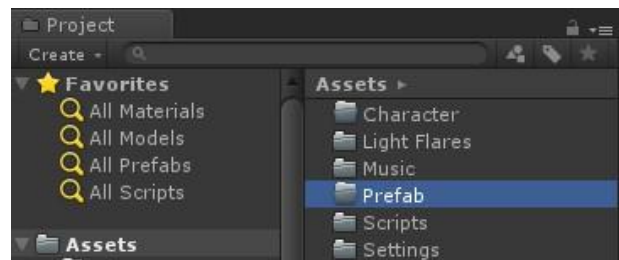


MEMBUAT SISTEM GAME (Prefab dan Sistem Damage)

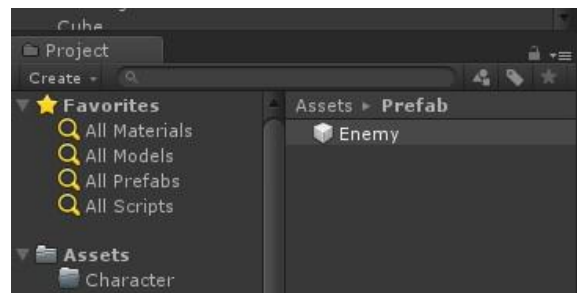
A. Pembuatan Prefab

Langkah-langkah:

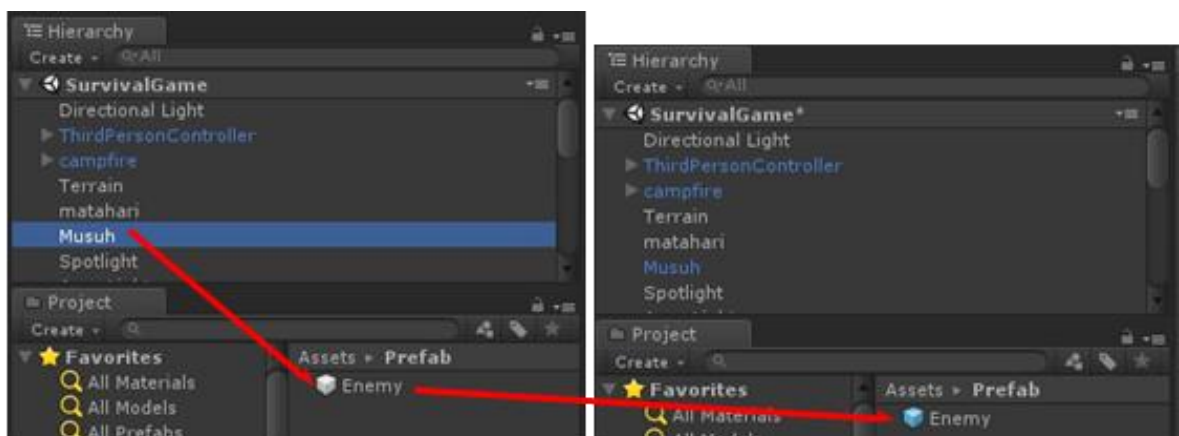
1. Buat folder **“Prefab”** dalam Asset, caranya: klik kanan Assets → **Create** → **Folder**



2. Klik kanan folder **Prefab** → **Create** → **Prefab**. Selanjutnya ubah namanya menjadi **“Enemy”**.



3. Drag Game object **“Musuh”** yang ada di hierarchy ke dalam prefab **“Enemy”**. Setelah itu prefab Enemy akan terlihat menjadi berwarna biru.



Dengan membuat prefab, maka kita dapat memperbanyak objek game secara cepat tanpa harus membuatnya dari awal. Kita cukup drag-drop preafb ke dalam Scene View sebanyak yang kita inginkan. Prefab yang telah kita simpan juga masih dapat kita tambah atau kita kurangi jumlah komponen yang ada di dalamnya.

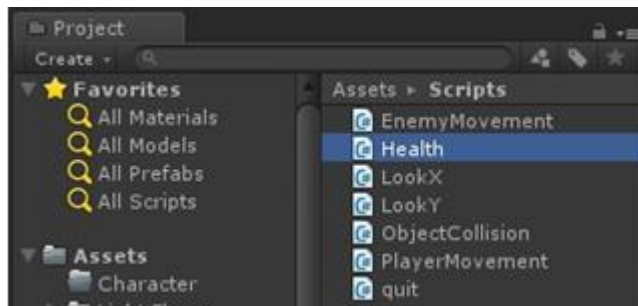
B. Membuat Sistem Damage

Damage (kerusakan) merupakan nilai kerusakan yang dihasilkan oleh serangan karakter atau musuh. Sistem damage akan membuat karakter dapat membunuh musuh atau sebaliknya. Hal penting yang menentukan sistem damage adalah *health point* (nyawa) dan kekuatan serangan.

C. Membuat Health Point

Health point merupakan nyawa yang menjadi batas karakter atau musuh dapat menerima damage akibat sebuah serangan. Health point akan semakin berkurang jika karakter/musuh terkena damage. Jika nilai dari health point 0, maka karakter atau musuh dikatakan “**mati**”. Adapun langkah pembuatan health point adalah sebagai berikut:

1. Buat skrip baru di dalam folder “**Scripts**” dengan nama “**Health**”.



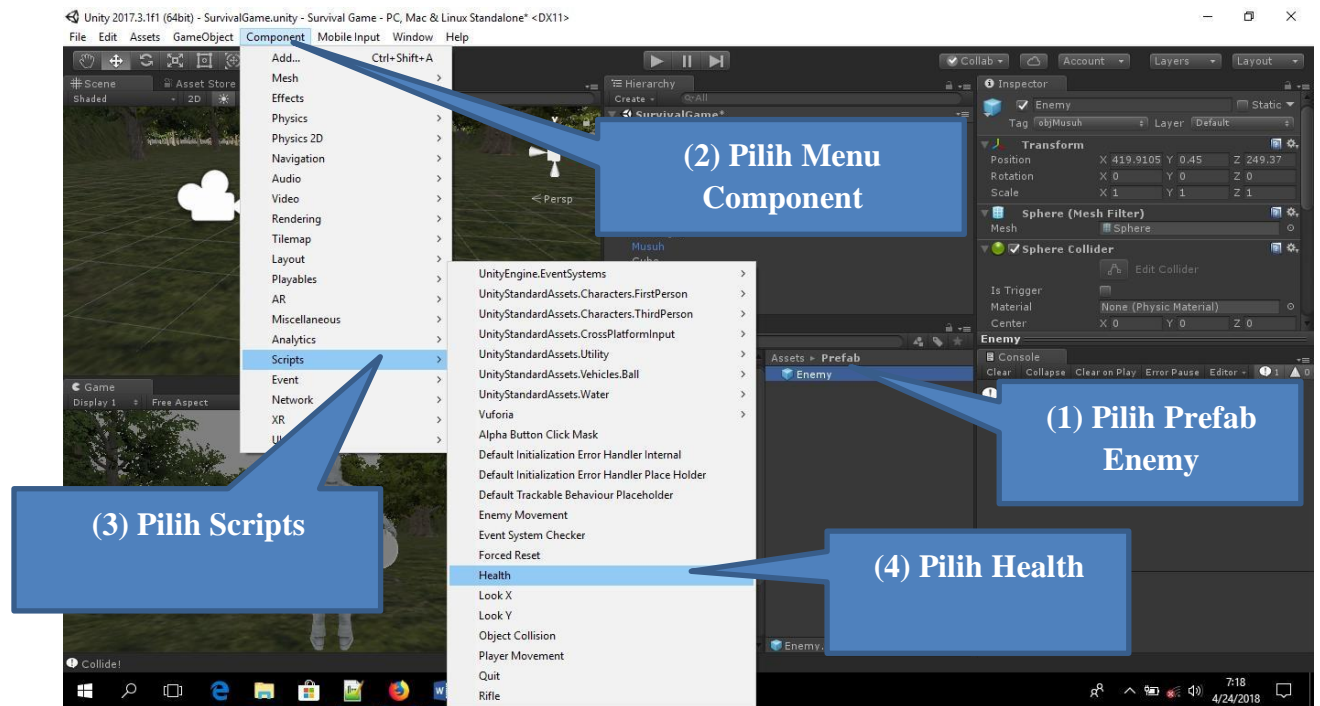
2. Ubah skrip menjadi seperti berikut:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Health : MonoBehaviour {
6     [SerializeField]
7     int _maximumHealth = 100;
8
9     int _currentHealth = 0;
10
11     void Start () {
12         _currentHealth = _maximumHealth;
13     }
14
15     void Damage (int damageValue) {
16         _currentHealth -= damageValue;
17     }
18 }
```

Keterangan:

Variabel **_maximumHealth** menentukan nilai maksimum dari health point. Variabel **_currentHealth** menentukan nilai dari health point saat ini. Ketika objek game menerima serangan, secara otomatis variabel akan berkurang nilainya (sesuai dengan nilai dari parameter damageValue).

3. Berikutnya tambahkan skrip “**Health**” ke dalam Prefab “**Enemy**”.



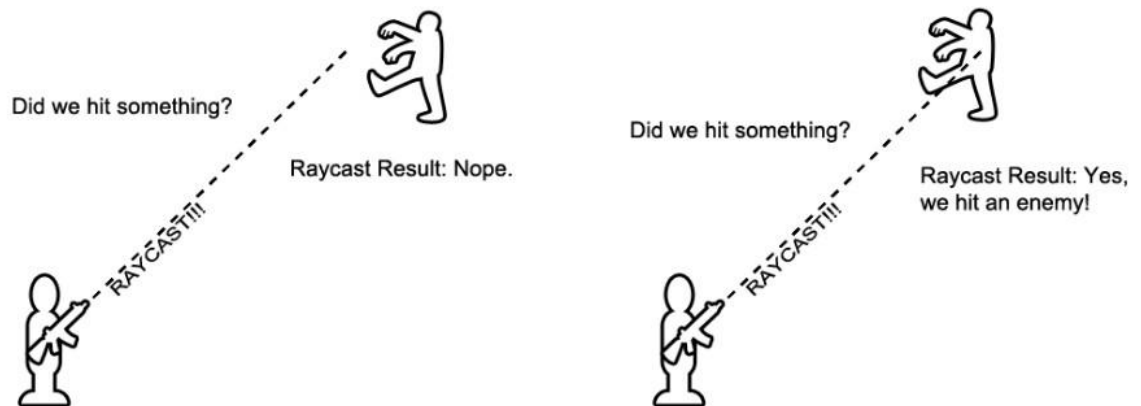
D. Membuat Senjata

Untuk membuat damage yang mengurangi health point, karakter utama akan dibeikan senjata. Pada praktik ini karakter akan menggunakan sebuah pistol. Cara kerja pistol akan ditembakkan secara lurus. Jika tembakan ini mengenai game objek yang memiliki health point, secara otomatis health point dari game objek akan berkurang. Pistol ini tidak akan diisi oleh peluru, melainkan sistem menemukan game objek dalam jalur tembakan akan bereaksi terhadap skrip health. Buat skrip C# dengan nama “Rifle” sebagai skrip senjata. Selanjutnya ubah isi skrip menjadi seperti berikut:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Rifle : MonoBehaviour {
6
7     void Update () {
8         Ray mouseRay = Camera.main.ViewportPointToRay (new Vector3 (0.5f, 0.5f, 0));
9         RaycastHit hitInfo;
10        if (Physics.Raycast (mouseRay, out hitInfo)) {
11            Debug.Log (hitInfo.transform.name);
12        }
13    }
14 }
```

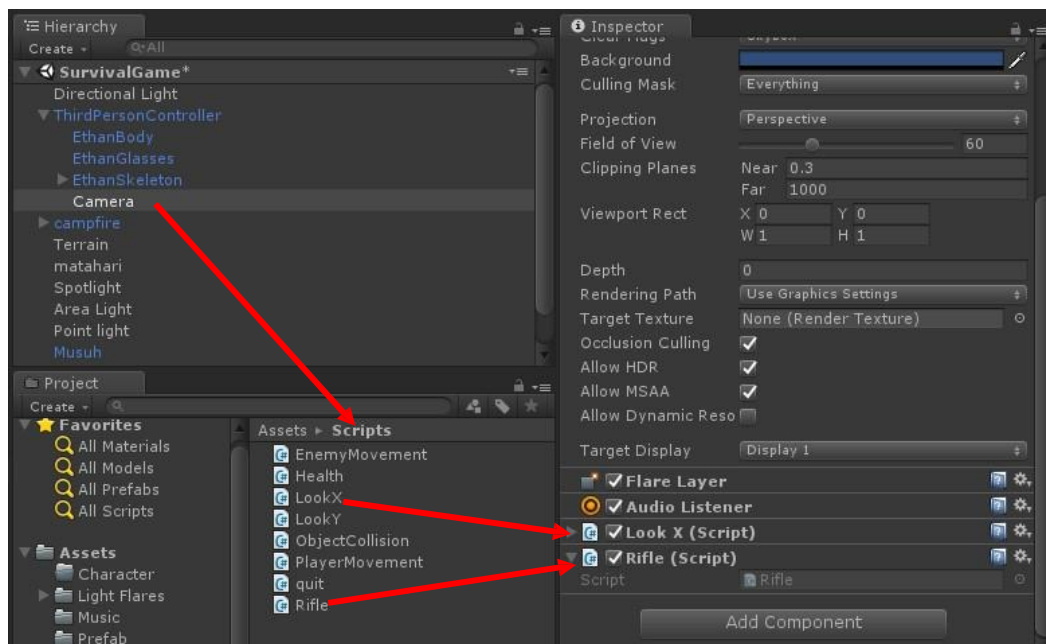
Keterangan:

- **Ray** mouseRay menentukan posisi serangan dimulai. Nilai (0.5f, 0.5f, 0) berarti posisi serangan berdasarkan sumbu x, y, dan z. nilai negatif akan membuat posisi serangan dimulai dari belakang kamera, sedangkan nilai positif akan membuat serangan di depan kamera.

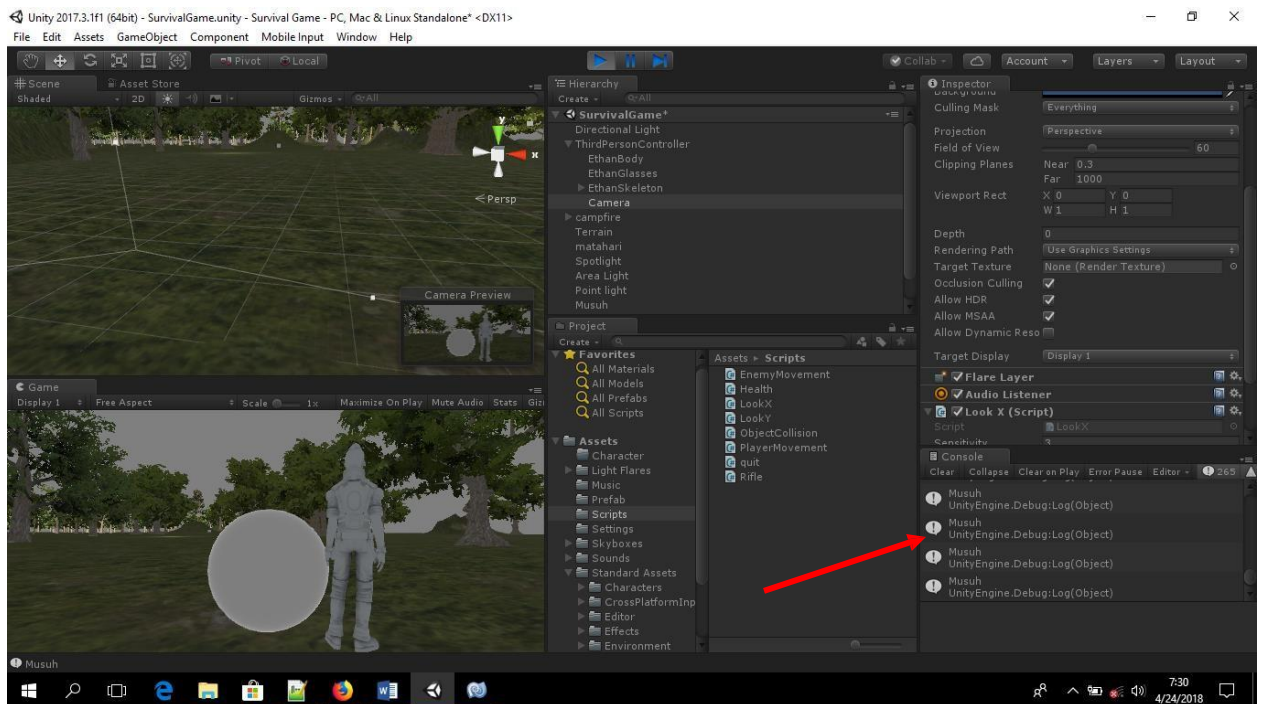


- **RaycastHit** hitInfo akan memberikan sejauh mana posisi target yang terkena serangan.
- **if** (**Physics.Raycast** (mouseRay, **out** hitInfo)) jika senjata berhasil mengenai objek yang memiliki health point, maka akan memberikan damage.

Untuk melakukan simulasi, tambahkan skrip “**Rifle**” dan “**LookX**” ke dalam objek “**Camera**” (pastikan kamera yang ada sudah ditandai sebagai “**Main Camera**”).



Selanjutnya tampilkan jendela console kemudian jalankan game, maka anda akan melihat debug komponen apa saja yang diarahi cursor akan ditampilkan dalam console.



Untuk membuat karakter dapat menembakkan pistol di saat yang dibutuhkan, ubah skrip "Rifle" menjadi seperti berikut:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Rifle : MonoBehaviour {
6
7     void Update () {
8         if (Input.GetButtonDown ("Fire1")) {
9             Ray mouseRay = Camera.main.ScreenPointToRay (Input.mousePosition);
10            RaycastHit hitInfo;
11            if (Physics.Raycast (mouseRay, out hitInfo)) {
12                Debug.Log (hitInfo.transform.name);
13            }
14        }
15    }
16 }
```

Keterangan:

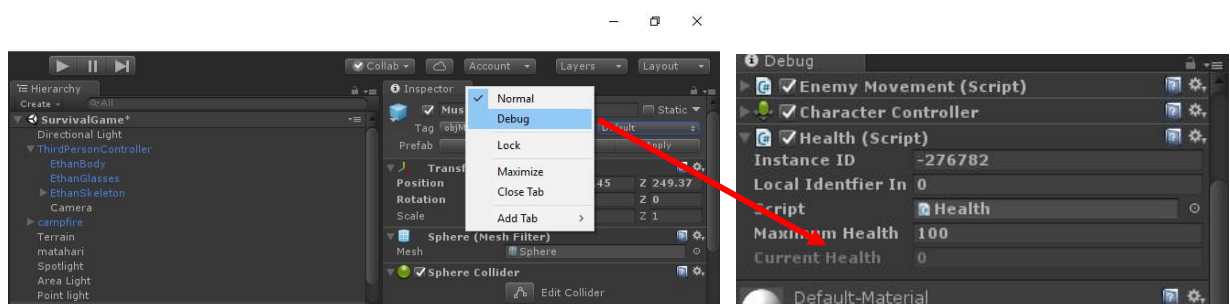
`if (Input.GetButtonDown("Fire1"))` adalah suatu kondisi dimana ketika ditekan tombol tersebut maka akan menembakkan senjata. Secara default, "Fire1" adalah klik kiri pada mouse.

E. Membuat Damage

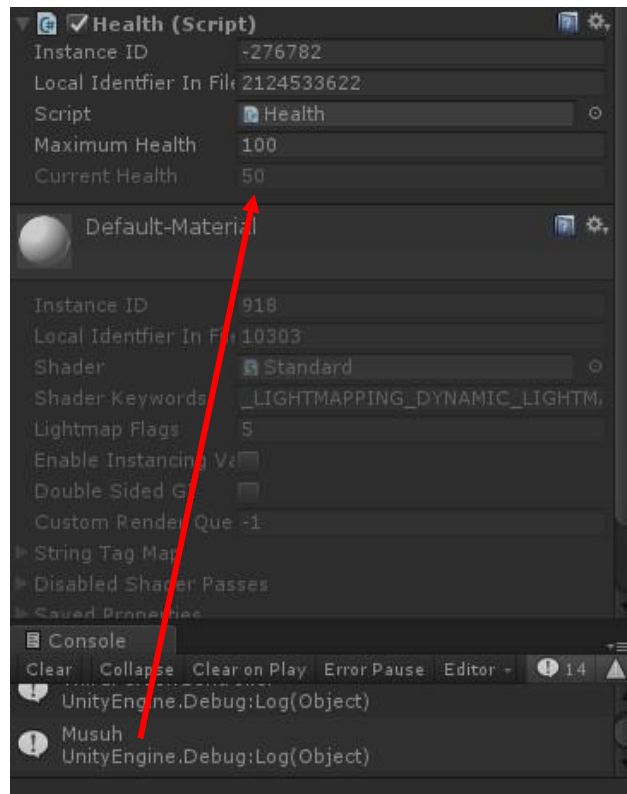
Setelah membuat senjata, langkah selanjutnya adalah membuat damage. Damage akan mengurangi health point lawan. Damage akan disatukan dengan senjata yang digunakan karakter untuk membuat damage, ubah skrip “**Rifle**” menjadi seperti berikut:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Rifle : MonoBehaviour {
6     [SerializeField]
7     int _damageDealt = 50;
8
9     void Update () {
10         if (Input.GetButtonDown ("Fire1")) {
11             Ray mouseRay = Camera.main.ScreenPointToRay (Input.mousePosition);
12             RaycastHit hitInfo;
13             if (Physics.Raycast (mouseRay, out hitInfo)) {
14                 Debug.Log (hitInfo.transform.name);
15                 Health enemyHealth = hitInfo.transform.GetComponent<Health>();
16                 if (enemyHealth != null) {
17                     enemyHealth.Damage (_damageDealt);
18                 }
19             }
20         }
21     }
22 }
```

Untuk melihat efek serangan terhadap musuh, pada kotak inspector objek “Musuh” klik kanan dan pilih Debug. Anda akan melihat “Current Health” pada kotak inspector menjadi read only. Sehingga kita tidak diperbolehkan merubah nilainya.



Untuk memudahkan dalam simulasi, anda dapat memberi nama pada musuh seperti Enemy1, Enemy2, Enemy3 dan seterusnya. Saat simulasi, jika lakukan klik kiri saat musuh berada di crosshair, maka kita akan melihat nilai “**Current Health**” musuh berkurang.



F. Menghilangkan Musuh

Musuh yang nilai current health = 0 seharusnya dianggap mati. Oleh karena itu, musuh yang mati tersebut seharusnya dihilangkan dari peta. Untuk melakukannya, ubah skrip “Health” menjadi seperti berikut:

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Health : MonoBehaviour {
6     [SerializeField]
7     int _maximumHealth = 100;
8
9     int _currentHealth = 0;
10
11     public void Start () {
12         _currentHealth = _maximumHealth;
13     }
14
15     public void Damage (int damageValue) {
16         currentHealth -= damageValue;
17         if (_currentHealth <= 0) {
18             Destroy (gameObject);
19         }
20     }
21 }

```

G. Pengaturan Mouse

Saat melakukan simulasi, gerakan mouse terasa sangat sulit atau sangat merepotkan. Hal ini dikarenakan mouse bergerak keluar dari layar Game View. Kita dapat mengatasi hal ini dengan mengubah skrip “Rifle” menjadi seperti berikut:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Rifle : MonoBehaviour {
6     [SerializeField]
7     int _damageDealt = 50;
8
9     void Start(){
10         Screen.lockCursor = true;
11     }
12
13     void Update () {
14         if(Input.GetKey(KeyCode.K)) {
15             Screen.lockCursor = false;
16         }
17         if (Input.GetButtonDown ("Fire1")) {
18             Screen.lockCursor = true;
19             Ray mouseRay = Camera.main.ScreenPointToRay (Input.mousePosition);
20             RaycastHit hitInfo;
21             if (Physics.Raycast (mouseRay, out hitInfo)) {
22                 Debug.Log (hitInfo.transform.name);
23                 Health enemyHealth = hitInfo.transform.GetComponent<Health>();
24                 if (enemyHealth != null) {
25                     enemyHealth.Damage (_damageDealt);
26                 }
27             }
28         }
29     }
30 }
```

Catatan:

1. Jalankan game. Ketika game pertama kali dijalankan, maka cursor akan di kunci (cursor tidak nampak pada layar game).
2. Untuk menampilkannya kembali, tekan ulang tombol K.
3. Ketika kita melakukan hit pada objek dalam game maka cursor akan dikunci kembali kunci (cursor tidak nampak pada layar game).
4. Untuk mengembalikan ke mode normal, berikan komentar pada kode yang diberi tanda kotak merah.


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Rifle : MonoBehaviour {
6     [SerializeField]
7     int _damageDealt = 50;
8
9     void Start(){
10         //Screen.LockCursor = true;
11     }
12
13     void Update () {
14         if(Input.GetKey(KeyCode.K)) {
15             //Screen.LockCursor = false;
16         }
17         if (Input.GetButtonDown ("Fire1")) {
18             //Screen.LockCursor = true;
19             Ray mouseRay = Camera.main.ScreenPointToRay (Input.mousePosition);
20             RaycastHit hitInfo;
21             if (Physics.Raycast (mouseRay, out hitInfo)) {
22                 Debug.Log (hitInfo.transform.name);
23                 Health enemyHealth = hitInfo.transform.GetComponent<Health>();
24                 if (enemyHealth != null) {
25                     enemyHealth.Damage (_damageDealt);
26                 }
27             }
28         }
29     }
30 }
```