



DISKUSIKAN MATERI

LAPORKAN MATERI

Daftar Modul

Masukkan kata kunci

| | |
|---------------------------------------|---|
| Persetujuan Hak Cipta | ✓ |
| Modul 1: Introduction Course (Gratis) | ✓ |
| Prasyarat Kelas (Gratis) | ✓ |
| Apa yang Akan Kita Pelajari? (Gratis) | ✓ |
| Tools Requirement | ✓ |
| Modul 2: ECMAScript 6 (Gratis) | ✓ |
| Running Snippet Code (Gratis) | ✓ |
| Persiapan Project Latihan (Gratis) | ✓ |

Template Literals

Potongan kode untuk materi ini: <https://repl.it/@dicodingacademy/163-02-template-literals?lite=true>

ES6 menawarkan kemudahan bagi developer dalam mengelola nilai string. Sebelum ES6, cara lama dalam menggabungkan nilai string adalah dengan menggunakan operator (+).

```
1. const user = {
2.   firstName: "Dimas",
3.   lastName: "Saputra",
4.   age: 18
5. }
6.
7. console.log("Nama: " + user.firstName + " " + user.lastName + ", Umur: " + user.age);
8.
9. /* output
10. Nama: Dimas Saputra, Umur: 18
11. */
```

Mungkin hal tersebut tidak terlalu repot, tapi jika dalam proses penggabungan tersebut terdapat garis baru, ini jadi sangat merepotkan.

```
1. const user = {
2.   firstName: "Dimas",
3.   lastName: "Saputra",
4.   age: 18
5. }
6.
7. const admin = {
8.   name: "Jane",
9.   position: "Marketing"
10. }
11.
12. const message = "Dear, " + user.firstName + " " + user.lastName + "\n\n" +
13. "Selamat ulang tahun yang ke-" + user.age + " semoga selalu diberikan kesehatan." + "\n\n" +
14. "Best Regards,\n" +
15. admin.name + "\n" +
16. admin.position;
17.
18. console.log(message);
19.
20. /* output ->
21. Dear, Dimas Saputra
22.
23. Selamat ulang tahun yang ke-18 semoga selalu diberikan kesehatan.
```

kita bisa lihat betapa repotnya membuat pesan tersebut. Entah berapa banyak operator (+) yang perlu kita gunakan.

Jangan khawatir karena ada template literals yang mampu menyederhanakan proses tersebut. Bagaimana menggunakannya? Gunakanlah tanda *backticks* (``) untuk menggantikan tanda *single quotes* (') atau *double quotes* (") yang digunakan sebelumnya. Template literals juga dapat mengandung sebuah *expression* dengan menuliskannya di dalam tanda *\$(expression)*. Hal ini sungguh membantu sekali dalam pembuatan nilai string yang kompleks menjadi jauh lebih mudah.

```
1. const user = {
2.   firstName: "Dimas",
3.   lastName: "Saputra",
4.   age: 18
5. }
6.
7. console.log(`Nama: ${user.firstName} ${user.lastName}, Umur: ${user.age}`);
8.
9. /* output:
10. Nama: Dimas Saputra, Umur: 18
11. */
```

Template literals juga dapat membaca *line space* secara langsung tanpa harus menggunakan *escape character*.

```
1. const user = {
2.   firstName: "Dimas",
3.   lastName: "Saputra",
4.   age: 18
5. }
6.
7. const admin = {
8.   name: "Jane",
9.   position: "Marketing"
10. }
11.
12. const message = `Dear, ${user.firstName} ${user.lastName}
13.
14. Selamat ulang tahun yang ke-${user.age} semoga selalu diberikan kesehatan.
15.
16. Best Regards,
17. ${admin.name}
18. ${admin.position}`;
19.
20. console.log(message);
21.
22. /* output ->
23. Dear, Dimas Saputra
```

[← KEMBALI KE MATERI SEBELUMNYA](#)[LANJUTKAN KE MATERI BERIKUTNYA →](#)



PERUSAHAAN

Tentang Kami

Blog

Berita Terbaru



PROGRAM

Academy

Challenge

Event

Job

Rewards

SUPPORT

Bantuan

FAQ

Hubungi Kami

