

Promise Methods

all - Promise.all()

Penjelasan: Mengembalikan satu Promise yang resolved ketika semua promise dalam iterable resolved, atau rejected jika ada yang rejected.

Syntax: `Promise.all(iterable)`

Contoh:

```
Promise.all([Promise.resolve(1), Promise.resolve(2)]).then(console.log);
```

Output:

```
[1, 2]
```

allSettled - Promise.allSettled()

Penjelasan: Mengembalikan promise yang selesai ketika semua promise dalam iterable selesai (fulfilled atau rejected).

Syntax: `Promise.allSettled(iterable)`

Contoh:

```
Promise.allSettled([Promise.resolve(1), Promise.reject('err')])  
  .then(console.log);
```

Output:

```
[{status: 'fulfilled', value: 1}, {status: 'rejected', reason: 'err'}]
```

any - Promise.any()

Penjelasan: Mengembalikan promise yang fulfilled pertama, atau rejected jika semua promise rejected.

Syntax: `Promise.any(iterable)`

Contoh:

```
Promise.any([Promise.reject('x'), Promise.resolve('y')]).then(console.log);
```

Output:

```
'y'
```

race - Promise.race()

Penjelasan: Mengembalikan promise pertama yang selesai (fulfilled atau rejected).

Syntax: `Promise.race(iterable)`

Contoh:

```
Promise.race([Promise.resolve(1), Promise.reject('err')]);
```

Output:

```
1
```

reject - Promise.reject()

Penjelasan: Mengembalikan promise yang langsung ditolak dengan alasan yang diberikan.

Syntax: `Promise.reject(reason)`

Contoh:

```
Promise.reject('error');
```

Output:

```
Promise {<rejected>: 'error'}
```

resolve - Promise.resolve()

Penjelasan: Mengembalikan promise yang langsung diselesaikan dengan nilai yang diberikan.

Syntax: `Promise.resolve(value)`

Contoh:

```
Promise.resolve(42);
```

Output:

```
Promise {<fulfilled>: 42}
```

Promise Methods

then - Promise.prototype.then()

Penjelasan: Menentukan callback untuk saat promise fulfilled atau rejected.

Syntax: `promise.then(onFulfilled, onRejected)`

Contoh:

```
Promise.resolve(1).then(val => val + 1);
```

Output:

```
2
```

catch - Promise.prototype.catch()

Penjelasan: Menangani penolakan (rejection) dari promise.

Syntax: `promise.catch(onRejected)`

Contoh:

```
Promise.reject('err').catch(e => 'caught: ' + e);
```

Output:

```
'caught: err'
```

finally - Promise.prototype.finally()

Penjelasan: Menentukan callback untuk dipanggil saat promise selesai (fulfilled atau rejected).

Syntax: `promise.finally(onFinally)`

Contoh:

```
Promise.resolve('done').finally(() => console.log('selesai'));
```

Output:

```
'selesai' (dari finally), 'done' (dari resolve)
```