

Anda tidak memerlukan jQuery

Dewasa ini perkembangan environment frontend sangatlah pesat, dimana banyak browser sudah mengimplementasikan DOM/BOM APIs dengan baik. Kita tidak perlu lagi belajar jQuery dari nol untuk keperluan manipulasi DOM atau events. Disaat yang sama; dengan berterimakasih kepada library frontend terkini seperti React, Angular dan Vue; Manipulasi DOM secara langsung telah menjadi anti-pattern alias sesuatu yang tidak perlu dilakukan. Dengan kata lain, jQuery sekarang menjadi semakin tidak diperlukan. Proyek ini memberikan informasi mengenai metode alternatif dari jQuery untuk implementasi Native dengan support untuk browser IE 10+.

Daftar Isi

1. [Query Selector](#)
2. [CSS & Style](#)
3. [DOM Manipulation](#)
4. [Ajax](#)
5. [Events](#)
6. [Utilities](#)
7. [Translation](#)
8. [Browser Support](#)

Query Selector

Untuk selector-selector umum seperti class, id atau attribute, kita dapat menggunakan

`document.querySelector` atau `document.querySelectorAll` sebagai pengganti. Perbedaan diantaranya adalah:

- `document.querySelector` mengembalikan elemen pertama yang cocok
- `document.querySelectorAll` mengembalikan semua elemen yang cocok sebagai NodeList. Hasilnya bisa dikonversikan menjadi Array `[]` `.slice.call (document.querySelectorAll (selector) || [])`;
- Bila tidak ada hasil pengembalian elemen yang cocok, jQuery akan mengembalikan `[]` sedangkan DOM API akan mengembalikan `null`. Mohon diperhatikan mengenai Null Pointer Exception. Anda juga bisa menggunakan operator `||` untuk set nilai awal jika hasil pencarian tidak ditemukan :
`document.querySelectorAll (selector) || []`

*Perhatian: `document.querySelector` dan `document.querySelectorAll` sedikit **LAMBAT**. Silahkan menggunakan `getElementById`, `document.getElementsByClassName` atau `document.getElementsByTagName` jika anda menginginkan tambahan performa.*

- [1.0](#) Query by selector

```
// jQuery
$('selector');
```

```
// Native
document.querySelectorAll('selector');
```

- [1.1](#) Query by class

```
// jQuery
$('.class');
```

```
// Native
document.querySelectorAll('.class');

// or
document.getElementsByClassName('class');
```

- [1.2](#) Query by id

```
// jQuery
$('#id');

// Native
document.querySelector('#id');

// or
document.getElementById('id');
```

- [1.3](#) Query menggunakan attribute

```
// jQuery
$('a[target=_blank]');

// Native
document.querySelectorAll('a[target=_blank]');
```

- [1.4](#) Pencarian.

- Mencari nodes

```
// jQuery
$el.find('li');

// Native
el.querySelectorAll('li');
```

- Mencari body

```
// jQuery
$('body');

// Native
document.body;
```

- Mencari Attribute

```
// jQuery
$el.attr('foo');
```

```
// Native
e.getAttribute('foo');
```

- Mencari data attribute

```
// jQuery
$el.data('foo');

// Native
// gunakan getAttribute
el.getAttribute('data-foo');
// anda juga bisa menggunakan `dataset` bila anda perlu support IE 11+
el.dataset['foo'];
```

- [1.5](#) Elemen-elemen Sibling/Previous/Next

- Elemen Sibling

```
// jQuery
$el.siblings();

// Native
[].filter.call(el.parentNode.children, function(child) {
    return child !== el;
});
```

- Elemen Previous

```
// jQuery
$el.prev();

// Native
el.previousElementSibling;
```

- Elemen Next

```
// next
$el.next();
el.nextElementSibling;
```

- [1.6](#) Closest

Mengembalikan elemen pertama yang cocok dari selector yang digunakan, dengan cara mencari mulai dari elemen-sekarang sampai ke document.

```
// jQuery
$el.closest(queryString);

// Native
```

```
function closest(el, selector) {
  const matchesSelector = el.matches || el.webkitMatchesSelector ||
    el.mozMatchesSelector || el.msMatchesSelector;

  while (el) {
    if (matchesSelector.call(el, selector)) {
      return el;
    } else {
      el = el.parentElement;
    }
  }
  return null;
}
```

- [1.7](#) Parents Until

Digunakan untuk mendapatkan "ancestor" dari setiap elemen yang ditemukan. Namun tidak termasuk elemen-sekarang yang didapat dari pencarian oleh selector, DOM node, atau object jQuery.

```
// jQuery
$.parentsUntil(selector, filter);

// Native
function parentsUntil(el, selector, filter) {
  const result = [];
  const matchesSelector = el.matches || el.webkitMatchesSelector ||
    el.mozMatchesSelector || el.msMatchesSelector;

  // match start from parent
  el = el.parentElement;
  while (el && !matchesSelector.call(el, selector)) {
    if (!filter) {
      result.push(el);
    } else {
      if (matchesSelector.call(el, filter)) {
        result.push(el);
      }
    }
    el = el.parentElement;
  }
  return result;
}
```

- [1.8](#) Form

- Input/Textarea

```
// jQuery
$('#my-input').val();
```

```
// Native
document.querySelector('#my-input').value;
```

- Get index of e.currentTarget between `.radio`

```
// jQuery
$(e.currentTarget).index('.radio');

// Native
[].indexOf.call(document.querySelectorAll('.radio'), e.currentTarget);
```

- [1.9](#) Iframe Contents

`$('iframe').contents()` mengembalikan `contentDocument`

- Iframe contents

```
// jQuery
$iframe.contents();

// Native
iframe.contentDocument;
```

- Iframe Query

```
// jQuery
$iframe.contents().find('.css');

// Native
iframe.contentDocument.querySelectorAll('.css');
```

[↑ back to top](#)

CSS Style

- [2.1](#) CSS

- Get style

```
// jQuery
$el.css("color");

// Native
// PERHATIAN: ada bug disini, dimana fungsi ini akan mengembalikan
// nilai 'auto' bila nilai dari atribut style adalah 'auto'
const win = el.ownerDocument.defaultView;
// null artinya tidak mengembalikan pseudo styles
win.getComputedStyle(el, null).color;
```

- Set style

```
// jQuery
$el.css({ color: "#ff0011" });

// Native
el.style.color = '#ff0011';
```

- Get/Set Styles

Mohon dicatat jika anda ingin men-set banyak style bersamaan, anda dapat menemukan referensi di metode [setStyles](#) pada package oui-dom-utils

- Add class

```
// jQuery
$el.addClass(className);

// Native
el.classList.add(className);
```

- Remove class

```
// jQuery
$el.removeClass(className);

// Native
el.classList.remove(className);
```

- has class

```
// jQuery
$el.hasClass(className);

// Native
el.classList.contains(className);
```

- Toggle class

```
// jQuery
$el.toggleClass(className);

// Native
el.classList.toggle(className);
```

- [2.2](#) Width & Height

Secara teori, width dan height identik, contohnya Height:

- Window height

```
// window height
$(window).height();
// without scrollbar, behaves like jQuery
window.document.documentElement.clientHeight;
// with scrollbar
window.innerHeight;
```

- Document height

```
// jQuery
$(document).height();

// Native
document.documentElement.scrollHeight;
```

- Element height

```
// jQuery
$el.height();

// Native
function getHeight(el) {
  const styles = this.getComputedStyle(el);
  const height = el.offsetHeight;
  const borderTopWidth = parseFloat(styles.borderTopWidth);
  const borderBottomWidth = parseFloat(styles.borderBottomWidth);
  const paddingTop = parseFloat(styles.paddingTop);
  const paddingBottom = parseFloat(styles.paddingBottom);
  return height - borderBottomWidth - borderTopWidth - paddingTop -
paddingBottom;
}
// accurate to integer (when `border-box`, it's `height`; when
`content-box`, it's `height + padding + border`)
el.clientHeight;
// accurate to decimal (when `border-box`, it's `height`; when
`content-box`, it's `height + padding + border`)
el.getBoundingClientRect().height;
```

- [2.3](#) Position & Offset

- Position

```
// jQuery
$el.position();

// Native
{ left: el.offsetLeft, top: el.offsetTop }
```

- Offset

```
// jQuery
$el.offset();

// Native
function getOffset (el) {
  const box = el.getBoundingClientRect();

  return {
    top: box.top + window.pageYOffset -
document.documentElement.clientTop,
    left: box.left + window.pageXOffset -
document.documentElement.clientLeft
  }
}
```

- [2.4](#) Scroll Top

```
// jQuery
$(window).scrollTop();

// Native
(document.documentElement && document.documentElement.scrollTop) ||
document.body.scrollTop;
```

[↑ back to top](#)

DOM Manipulation

- [3.1](#) Remove

```
// jQuery
$el.remove();

// Native
el.parentNode.removeChild(el);
```

- [3.2](#) Text

- Get text

```
// jQuery
$el.text();

// Native
el.textContent;
```

- Set text


```
// jQuery
$el.text(string);

// Native
el.textContent = string;
```

- [3.3](#) HTML

- Get HTML

```
// jQuery
$el.html();

// Native
el.innerHTML;
```

- Set HTML

```
// jQuery
$el.html(htmlString);

// Native
el.innerHTML = htmlString;
```

- [3.4](#) Append

Menambahkan elemen-anak setelah anak terakhir dari elemen-parent

```
// jQuery
$el.append("<div id='container'>hello</div>");

// Native
let newEl = document.createElement('div');
newEl.setAttribute('id', 'container');
newEl.innerHTML = 'hello';
el.appendChild(newEl);
```

- [3.5](#) Prepend

```
// jQuery
$el.prepend("<div id='container'>hello</div>");

// Native
let newEl = document.createElement('div');
newEl.setAttribute('id', 'container');
newEl.innerHTML = 'hello';
el.insertBefore(newEl, el.firstChild);
```

- [3.6 insertBefore](#)

Memasukkan node baru sebelum elemen yang dipilih.

```
// jQuery
$newEl.insertBefore(queryString);

// Native
const target = document.querySelector(queryString);
target.parentNode.insertBefore(newEl, target);
```

- [3.7 insertAfter](#)

Memasukkan node baru sesudah elemen yang dipilih.

```
// jQuery
$newEl.insertAfter(queryString);

// Native
const target = document.querySelector(queryString);
target.parentNode.insertBefore(newEl, target.nextSibling);
```

[↑ back to top](#)

Ajax

Gantikan dengan [fetch](#) dan [fetch-jsonp](#)

[↑ back to top](#)

Events

Untuk penggantian secara menyeluruh dengan namespace dan delegation, rujuk ke <https://github.com/oneuijs/oui-dom-events>

- [5.1](#) Bind event dengan menggunakan on

```
// jQuery
$el.on(eventName, eventHandler);

// Native
el.addEventListener(eventName, eventHandler);
```

- [5.2](#) Unbind event dengan menggunakan off

```
// jQuery
$el.off(eventName, eventHandler);

// Native
el.removeEventListener(eventName, eventHandler);
```

- [5.3](#) Trigger

```
// jQuery
$(el).trigger('custom-event', {key1: 'data'});

// Native
if (window.CustomEvent) {
  const event = new CustomEvent('custom-event', {detail: {key1: 'data'}});
} else {
  const event = document.createEvent('CustomEvent');
  event.initCustomEvent('custom-event', true, true, {key1: 'data'});
}

el.dispatchEvent(event);
```

[↑ back to top](#)

Utilities

- [6.1](#) isArray

```
// jQuery
$.isArray(range);

// Native
Array.isArray(range);
```

- [6.2](#) Trim

```
// jQuery
$.trim(string);

// Native
string.trim();
```

- [6.3](#) Object Assign

Extend, use object.assign polyfill <https://github.com/ljharb/object.assign>

```
// jQuery
$.extend({}, defaultOpts, opts);

// Native
Object.assign({}, defaultOpts, opts);
```

- [6.4](#) Contains

```
// jQuery
$.contains(el, child);
```

```
// Native
el !== child && el.contains(child);
```

[↑ back to top](#)

Terjemahan

- [한국어](#)
- [简体中文](#)
- [Bahasa Melayu](#)
- [Bahasa Indonesia](#)
- [Português\(PT-BR\)](#)
- [Tiếng Việt Nam](#)
- [Русский](#)
- [Türkçe](#)

Browser yang di Support

 Chrome	 Firefox	 IE	 Opera	 Safari
Latest ✓	Latest ✓	10+ ✓	Latest ✓	6.1+ ✓

License

MIT