**TAGGED TEMPLATES** adalah bentuk yang lebih kompleks dari Template Literals, memungkinkan kita membaca template literal melalui sebuah function

Contoh:

const nama = 'Sandhika Galih';

const umur = 33;

function coba(strings, ...values){

let result = '';

strings.forEach((str,i) => {

result += `${str}${values[i]||''}`;

});

return result;

}

const str = coba `Hello, nama saya ${nama}, umur saya ${umur} tahun. `;

console.log (str);

atau bisa juga menggunakan reduce sbb:

const nama = 'Sandhika Galih';

const umur = 33;

function coba(strings, ...values){

return strings.reduce((result,str,i) => `${result}${str}${values[i] || ''}`,'');

}

const str = coba `Hello, nama saya ${nama}, umur saya ${umur} tahun. `;

console.log (str);

Penggunaan dari Tagged Template:

1. Escaping HTML Tags -> Sanitasi HTML Tags (menghindari script yang tidak diinginkan)

function sanitize (strings, …values) {

return DOMPurify.sanitize(aboutMe);

}

const name = ‘petyr baelish’;

const aboutMe = `I love to do evil <img src=”<http://unplash.it/100/100?random>”

onload=”alert (‘I hacked you hahaha…’);” />`;

const html = sanitize `

<h3>${name}</h3>

<p>${aboutMe}</p>`;

1. Translation & Internationalization -> alih bahasa

const html = i18n`Hello ${name}, you have ${amount}:c in your bank account`

1. Styled Component

const Title = styled.h1`

font-size: 1.5 em;

text-align: center;

color: palevioletred;`