**Introduction**



Ada beberapa cara untuk menyimpan data pengguna seperti pencarian, artikel berita, dan lain-lain ke lokal (*browser*) menggunakan *web storage* seperti *cookies*, *local storage*, dan *session storage*. Data ini dimanfaatkan oleh situs web tersebut untuk merekam kebiasaan pengguna agar dapat memberikan rekomendasi sesuai preferensi si pengguna tersebut.

Manakah di antara ketiganya yang harus kita gunakan? Apakah kelebihan dan kekurangannya? Lalu bagaimana cara menggunakannya?

**Apa itu *Cookies*?**

*Cookies* adalah data kecil yang dikirim dari situs web dan disimpan di komputer kita oleh *web browser* saat kita menjelajah. Disebut data kecil karena maksimum data yang dapat disimpan dalam *cookies* adalah 4096 *bytes* (4 KB).

Biasanya data yang disimpan di *cookies* adalah *access token* pengguna saat login atau data pencarian saat melakukan pencarian pada situs web tertentu. Hal ini yang biasanya dilakukan oleh situs pencarian untuk melacak pencarian kita dan menampilkan iklan yang berhubungan dengan pencarian kita sebelumnnya.

Namun ada beberapa kekurangan yang perlu kita perhatikan mengenai *cookies* di antaranya:

1. Setiap kita mengakses situs web, *cookies* juga kembali dikirim sehingga memperlambat aplikasi web kamu dengan mengirimkan data yang sama.
2. *Cookies* disertakan pada setiap *HTTP request*, sehingga mengirimkan data yang tidak dienkripsi melalui internet, maka saat kita ingin menyimpan data dalam *cookies* kita harus mengenkripsinya terlebih dahulu.
3. *Cookies* hanya dapat menyimpan data sebanyak 4KB.
4. Lalu *cookies* juga memiliki tanggal kadaluarsa. Tanggal ini telah ditentukan sehingga *web browser* bisa menghapus *cookies* jika tanggal sudah kadaluarsa atau tidak dibutuhkan.

Kita dapat memanfaatkan jenis *web storage* yang lain untuk mengatasi kekurangan yang dimiliki *cookies*.

***Local Storage* dan *Session Storage***

Dengan memanfaatkan *local storage* dan *session storage*, kita dapat menyimpan data lebih besar yaitu 5MB *per page* tanpa mempengaruhi kinerja situs web. Namun, penting untuk diketahui agar kita tidak menyimpan data sensitif seperti *password* ke dalam *local storage* ataupun *session storage* untuk menghindari serangan pencurian data. Pada *lesson* ini kita akan fokus mempelajari keduanya.

# Local Storage - Menyimpan Data

Pernahkah kita saat melakukan pencarian pada sebuah situs lalu situs tersebut menampilkan riwayat pencarian kita? Iya, data pencarian tersebut disimpan ke dalam local storage untuk diolah menjadi riwayat pencarian. Itulah salah satu contoh penerapan dari local storage pada aplikasi web.

Local storage memiliki karakteristik sebagai berikut:

1. Menyimpan data tanpa tanggal kadaluarsa.
2. Data tidak akan dihapus ketika web browser ditutup dan akan tersedia seterusnya selama kita tidak menghapus data local storage pada web browser.
3. Dapat menyimpan data hingga 5MB.
4. Hanya dapat menyimpan data string.

Untuk menyimpan data pada local storage, kita menggunakan method setItem() yang membutuhkan 2 parameter. Parameter pertama adalah key yang ingin kita simpan dan parameter kedua adalah data (value) dari key yang akan disimpan.

localStorage.setItem('key', value);

Agar dapat dipahami dengan baik, kita akan membuat aplikasi web sederhana untuk melakukan pencarian.

### Langkah 1 - Buat file index.html

Berikut kode yang ada pada file index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

body {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

}

form {

display: flex;

flex-direction: row;

}

form input {

padding: 5px 10px;

}

</style>

</head>

<body>

<form>

<input type="text" id="searchkey" name="searchkey" placeholder="Search Something"><br>

<input type="submit" value="Search" onclick="onSearch()">

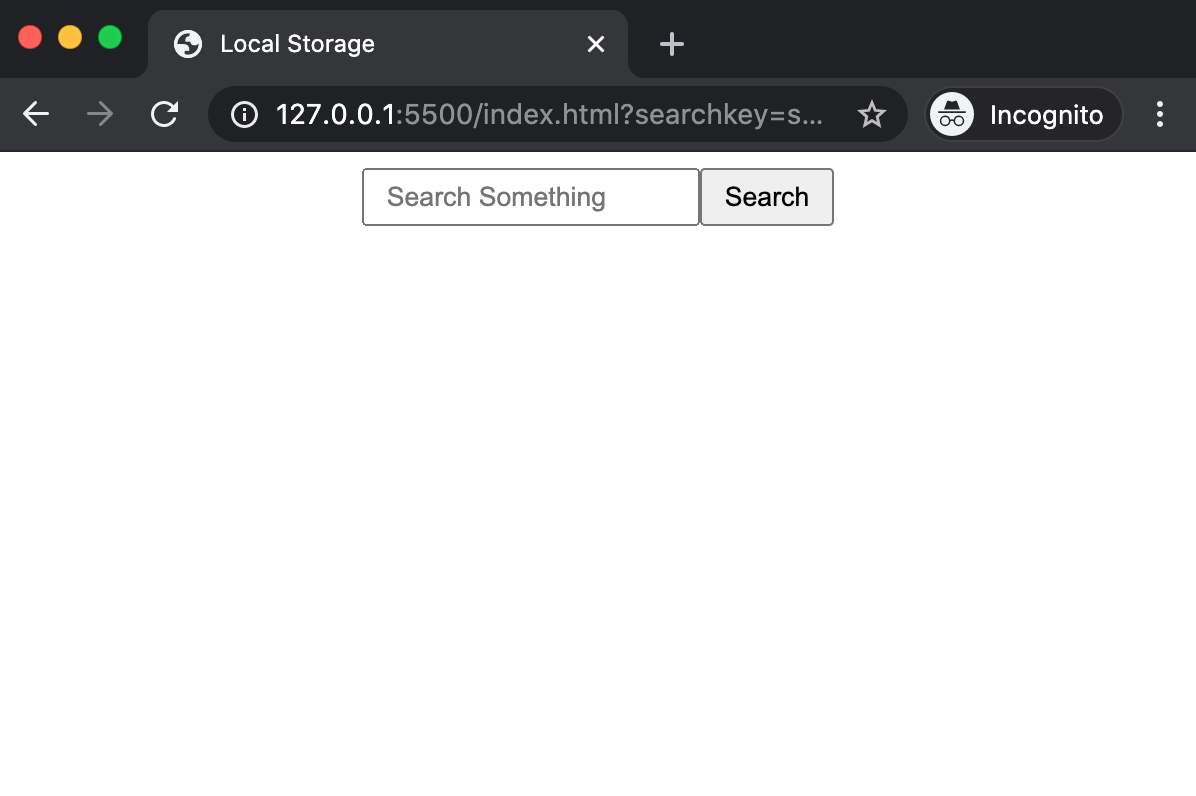
</form>

<script></script>

</body>

</html>

Jika kode di atas dijalankan, maka akan terlihat tampilan web seperti gambar di bawah ini:



### Langkah 2 - Isi tag <script> dengan fungsi onSearch()

Berikutnya kita akan membuat kode JavaScript di dalam tag <script>. Jika user klik tombol Search, maka function onSearch() akan dipanggil dan menjalankan kode di dalamnya. Berikut kode yang ada pada tag script:

var searchList = [];

function onSearch() {

var searchValue = document.getElementById('searchkey').value;

searchList.push(searchValue) // memasukan kata pencarian ke dalam array

var searchListString = JSON.stringify(searchList); // mengubah array menjadi string

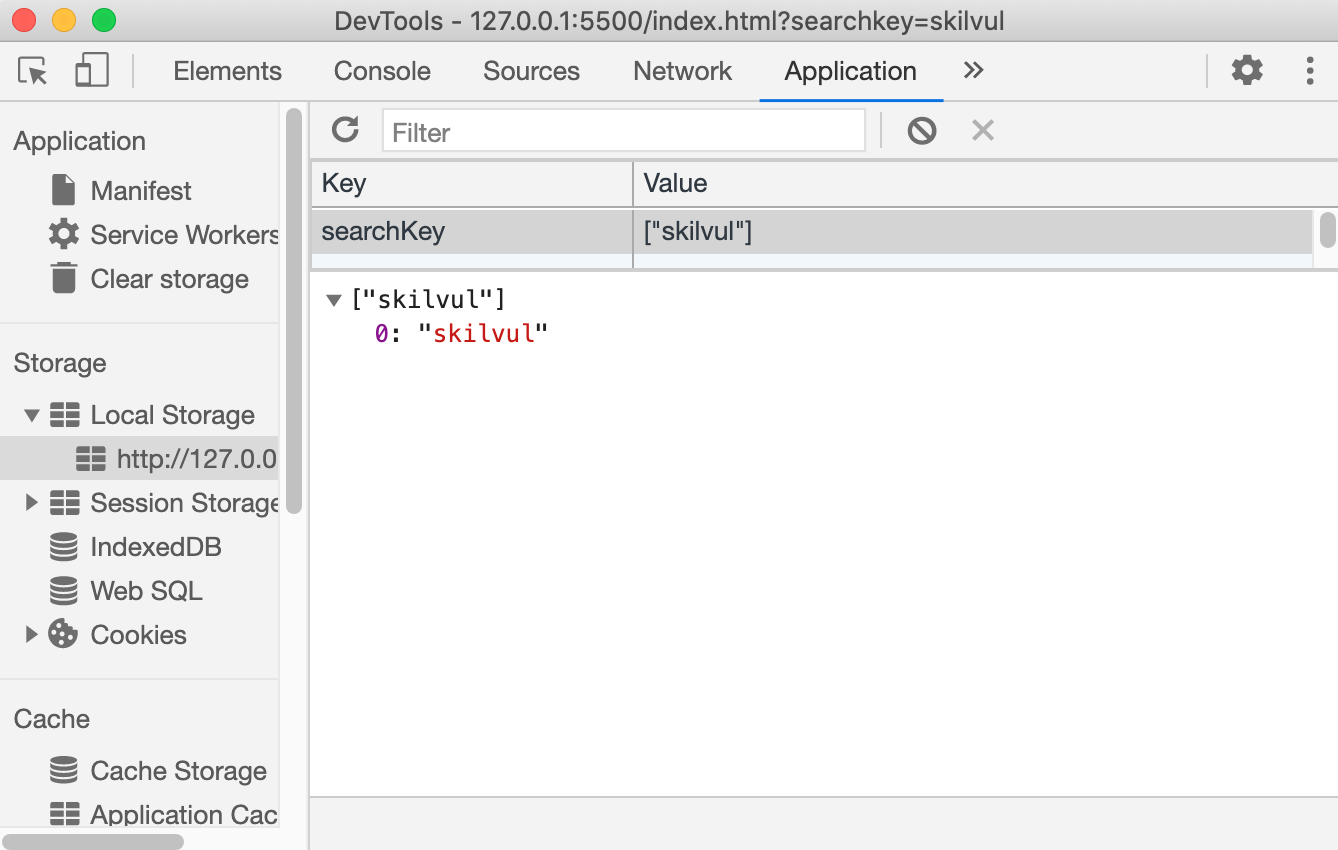
localStorage.setItem('searchKey', searchListString); // menyimpan pencarian dengan key 'searchKey'

}

Dari kode di atas dapat kita lihat bahwa kata yang telah dicari dimasukan ke dalam array. Hal ini bertujuan agar kita dapat menyimpan kata yang dicari pengguna sebanyak mungkin. Kemudian kita juga **menggunakan** JSON.stringify() untuk merubah array searchList menjadi string, hal ini dilakukan karena data yang dapat disimpan ke dalam local storage **hanya** tipe data string.

Maka pada saat kita mengetik skilvul pada input, lalu klik tombol Search, kita dapat melihat data yang tersimpan dalam local storage browser kita dengan cara:

1. Klik kanan pada halaman aplikasi web yang sedang diakses. Pilih ***inspect***.
2. Pilih dan klik ***Application***. Lalu klik ***Local Storage*** di sisi tab sebelah kiri. Hasil yang didapatkan adalah seperti ini:



Kamu dapat melihat gambar di atas, bahwa kita telah berhasil menyimpan data pada local storage.

# Local Storage - Mengambil Data

Pada topik sebelumnya kita telah mempelajari cara menyimpan data pada local storage menggunakan setItem(). Pada topik ini kita akan melanjutkan pembahasan bagaimana mengambil data yang telah tersimpan pada local storage.

Untuk mengambil data yang telah tersimpan pada local storage, kita dapat menggunakan method getItem() yang membutuhkan 1 parameter. Parameter tersebut adalah key dari data yang kita inginkan.

localStorage.getItem('key');

Untuk lebih memahami, kita akan melanjutkan aplikasi yang telah dibuat pada topik sebelumnya yaitu aplikasi situs pencarian. Pada topik sebelumnya, kita berhasil menyimpan pencarian kita ke dalam local storage. Pada topik ini, kita akan menampilkan riwayat pencarian.

### Langkah 1 - Modifikasi file index.html dari topik sebelumnya

Kita akan menambahkan beberapa elemen untuk menampilkan riwayat pencarian:

...

...

</form>

<h4>Riwayat Pencarian</h4>

<div id="search-history"></div>

<script>

...

</body>

</html>

Dari kode di atas, kita menambahkan teks Riwayat Pencarian di bawah tag </form> diikuti dengan elemen <div id="search-history"></div>.

### Langkah 2 - Membuat fungsi getSearchHistory()

Selanjutnya kita akan membuat function getSearchHistory() di dalam tag <script> yang digunakan untuk mengambil data pada local storage. Kita akan menggunakan method getItem() untuk menampilkan data searchKey pada halaman index.html.

<script>

var searchList = JSON.parse(localStorage.getItem('searchKey')) || []; // jika searchKey bernilai undefined, maka set searchList sebagai empty array

...

...

function getSearchHistory() {

var list = '';

for (var i = 0; i < searchList.length; i++) {

list += `<div>${searchList[i]}</div>`;

}

document.getElementById('search-history').innerHTML = list;

}

// memanggil fungsi getSearchHistory

if (searchList.length > 0) { // Jika panjang array searchList > 0

getSearchHistory(); // panggil fungsi getSearchHistory

}

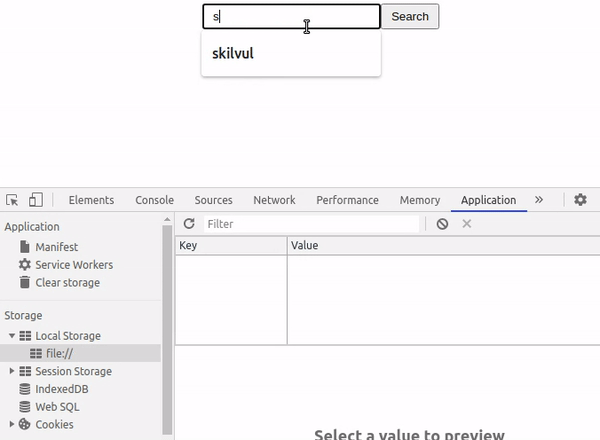
</script>

Dari kode di atas, kita dapat lihat bahwa kita memodifikasi variabel searchList menjadi:

var searchList = JSON.parse(localStorage.getItem("searchKey")) || [];

Maksud dari kode ini adalah variabel searchList nilai default-nya adalah data dari local storage dengan key searchKey. Namun jika data pada local storage dengan key searchKey tidak ada, maka isi variabel searchList dengan array kosong []. Lalu disini kita menggunakan JSON.parse() yang merupakan kebalikan dari JSON.stringify(), yaitu untuk mengubah string menjadi bentuk array kembali.

Sehingga kita mendapatkan hasil seperti ini:



📝**Catatan**:

Riwayat pencarian akan tampil setelah halaman direfresh.

# Local Storage - Menghapus Data

Pada topik sebelumnya kita telah mempelajari cara mengambil data pada local storage menggunakan getItem(). Pada topik ini kita akan melanjutkan pembahasan bagaimana **menghapus** data yang telah tersimpan pada local storage .

Untuk menghapus data yang telah tersimpan pada local storage, kita dapat menggunakan method removeItem() yang membutuhkan 1 parameter. Parameter tersebut adalah key dari data yang ingin kita hapus.

// menghapus key tertentu

localStorage.removeItem("key");

// menghapus semua key

localStorage.clear();

Untuk lebih memahami cara penggunaan dari removeItem(), kita akan melanjutkan pengembangan aplikasi web pencarian sederhana yang telah dibuat pada topik sebelumnya. Pada topik ini kita akan menghapus data pada local storage saat user klik tombol Hapus Riwayat pada halaman index.html.

### Langkah 1 - Buat <button> Hapus Riwayat

Berikut detail kode yang ada pada file index.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

...

...

<h4>Riwayat Pencarian</h4>

<div id="search-history"></div>

<button onclick="clearSearchHistory()" style="margin-top: 20px;">Hapus Riwayat</button>

</body>

</html>

Pada file index.html, kita menambahkan <button> Hapus Riwayat yang memanggil fungsi clearSearchHistory().

### Langkah 2 - Buat fungsi clearSearchHistory()

Kita tambahkan fungsi clearSearchHistory() dalam tag <script>

function clearSearchHistory() {

localStorage.removeItem("searchKey"); // menghapus data pada localStorage dengan key "searchKey"

document.getElementById('search-history').innerHTML = ""; // mengosongkan riwayat pencarian

}

Maka hasilnya akan menjadi seperti ini:

