“Build your imagination with Naratass Software”

Frontend JavaScript

Mengimplementasikan Backend ke Frontend

Pertemuan 19-20— Belajar JavaScript Tingkat Basic
Disampaikan oleh: Aries Indrabayu Yunandar

Struktur Frontend Modular

```
catatan-node-api/
├── frontend/
│   ├── index.html                ← Entry point utama (template SPA)
│   ├── assets/
│   │   ├── css/
│   │   │   └── style.css         ← Custom styling
│   │   ├── js/
│   │   │   ├── api/
│   │   │   │   └── catatanApi.js ← Semua akses ke API catatan
│   │   │   ├── pages/
│   │   │   │   ├── list.js       ← Halaman list catatan
│   │   │   │   ├── form.js       ← Halaman tambah/edit catatan
│   │   │   ├── components/
│   │   │   │   ├── navbar.js     ← Navbar SPA
│   │   │   │   ├── table.js      ← Table renderer reusable
│   │   │   ├── router.js         ← SPA Router (berbasis hash `#/list`, `#/form`)
│   │   │   └── main.js           ← Entry JS untuk SPA logic
│   │   └── libs/
│   │      └── bootstrap.min.css  ← CDN fallback atau manual import
│   └── templates/
│      ├── list.html              ← Template HTML list catatan
│      ├── form.html              ← Template HTML form catatan
│
```

Single Page Application (SPA) adalah **aplikasi web yang berjalan di satu halaman HTML saja**. Ketika pengguna berinteraksi atau berpindah "halaman", **tidak ada reload penuh ke server**, melainkan hanya bagian tertentu dari halaman yang diubah menggunakan JavaScript (biasanya dengan AJAX atau fetch API).

File: frontend/index.html

```html
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Catatan App - SPA</title>
  <!-- Bootstrap CDN -->
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
    rel="stylesheet"
  />
  <link rel="stylesheet" href="./assets/css/style.css" />
</head>
<body>
  <!-- 🧭 Navbar SPA -->
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
    <a class="navbar-brand" href="#">📒 CatatanApp</a>
    <div class="collapse navbar-collapse">
      <ul class="navbar-nav ms-auto">
        <li class="nav-item">
          <a class="nav-link" href="#/list">📄 Daftar Catatan</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#/form">✏️ Tambah Catatan</a>
        </li>
      </ul>
    </div>
  </nav>

  <!-- 📦 SPA Container -->
  <div class="container my-4" id="app">
    <!-- Konten akan dirender dinamis oleh JS -->
  </div>

  <!-- JS -->
  <script src="./assets/js/router.js"></script>
  <script src="./assets/js/main.js"></script>
</body>
</html>
```

File: frontend/assets/js/router.js

```
const routes = {
  '#/list': {
    template: './templates/list.html',
    script: './assets/js/pages/list.js',
  },
  '#/form': {
    template: './templates/form.html',
    script: './assets/js/pages/form.js',
  },
  '#/edit': {
    template: './templates/form.html', // Reuse form template for edit
    script: './assets/js/pages/form.js',
  },
};

async function loadTemplate(templatePath) {
  const response = await fetch(templatePath);
  return response.text();
}

async function loadScript(scriptPath) {
  return new Promise((resolve, reject) => {
    const script = document.createElement('script');
    script.src = scriptPath;
    script.type = 'module';
    script.onload = resolve;
    script.onerror = reject;
    document.body.appendChild(script);
  });
}

export async function router() {
  const hash = window.location.hash || '#/list';
  const route = routes[hash.split('?')[0]]; // strip query param if any

  if (!route) {
    document.getElementById('app').innerHTML = `<div class="alert alert-danger">Halaman
tidak ditemukan!</div>`;
    return;
  }

  try {
    const html = await loadTemplate(route.template);
    document.getElementById('app').innerHTML = html;
    await loadScript(route.script);
  } catch (err) {
    document.getElementById('app').innerHTML = `<div class="alert alert-danger">Gagal
memuat halaman.</div>`;
    console.error('Router Error:', err);
  }
}
```

File: frontend/assets/js/pages/list.js

```
import { getAllCatatan, softDeleteCatatan, restoreCatatan, forceDeleteCatatan } from '../api/catatanApi.js';

let currentPage = 1;

document.addEventListener('DOMContentLoaded', () => {
  document.getElementById('filterBtn').addEventListener('click', () => {
    currentPage = 1;
    loadData();
  });

  loadData();
});

async function loadData() {
  const search = document.getElementById('searchInput').value.trim();
  const deleted = document.getElementById('deletedFilter').value;
  const sortBy = document.getElementById('sortBy').value;
  const sortDir = document.getElementById('sortDir').value;
  const size = 5;

  const response = await getAllCatatan({ page: currentPage, size, search, sortBy, sortDir, deleted });
  if (response.status === 'success') {
    renderTable(response.data);
    renderPagination(response);
  } else {
    document.getElementById('catatanTableBody').innerHTML = `<tr><td colspan="6" class="text-center text-danger">Gagal memuat data</td></tr>`;
  }
}

function renderTable(rows) {
  const tbody = document.getElementById('catatanTableBody');
  tbody.innerHTML = '';

  if (rows.length === 0) {
    tbody.innerHTML = `<tr><td colspan="6" class="text-center">Tidak ada data ditemukan</td></tr>`;
    return;
  }

  for (const row of rows) {
    const tr = document.createElement('tr');

    tr.innerHTML = `
      <td>${row.id}</td>
      <td>${row.isi}</td>
      <td>${row.kategori ?? '-'}</td>
      <td>${row.status ? 'Aktif' : 'Nonaktif'}</td>
      <td>${new Date(row.updated_at).toLocaleString()}</td>
      <td>
        <a href="#/edit?id=${row.id}" class="btn btn-sm btn-warning">Edit</a>
        ${row.deleted_at ? `
          <button class="btn btn-sm btn-success btn-restore" data-id="${row.id}">Restore</button>
          <button class="btn btn-sm btn-danger btn-force" data-id="${row.id}">Hapus Permanen</button>
        ` : `
          <button class="btn btn-sm btn-outline-danger btn-delete" data-id="${row.id}">Hapus</button>
        `}
      </td>
    `;

    tbody.appendChild(tr);
  }

  // Attach event listener ke tombol dinamis
  attachButtonListeners();
}

function renderPagination({ page, total_page }) {
  const container = document.getElementById('pagination');
  container.innerHTML = '';

  for (let i = 1; i <= total_page; i++) {
    const li = document.createElement('li');
    li.className = `page-item ${i === page ? 'active' : ''}`;
    li.innerHTML = `<button class="page-link">${i}</button>`;
    li.addEventListener('click', () => {
      currentPage = i;
      loadData();
    });
    container.appendChild(li);
  }
}

function attachButtonListeners() {
  document.querySelectorAll('.btn-delete').forEach(btn => {
    btn.addEventListener('click', async () => {
      if (confirm('Yakin ingin menghapus catatan ini?')) {
        await softDeleteCatatan(btn.dataset.id);
        loadData();
      }
    });
  });

  document.querySelectorAll('.btn-restore').forEach(btn => {
    btn.addEventListener('click', async () => {
      await restoreCatatan(btn.dataset.id);
      loadData();
    });
  });

  document.querySelectorAll('.btn-force').forEach(btn => {
    if (confirm('Yakin ingin menghapus PERMANEN? Ini tidak bisa dikembalikan!')) {
      btn.addEventListener('click', async () => {
        await forceDeleteCatatan(btn.dataset.id);
        loadData();
      });
    }
  });
}
```

File: frontend/assets/js/api/catatanApi.js

```
const BASE_URL = 'http://localhost:7000/api/v1/catatan';

export async function getAllCatatan(params = {}) {
  const query = new URLSearchParams({
    page: params.page || 1,
    size: params.size || 10,
    search: params.search || '',
    sort_by: params.sortBy || 'updated_at',
    sort_dir: params.sortDir || 'desc',
    deleted: params.deleted || '0'
  });

  try {
    const res = await fetch(`${BASE_URL}?${query.toString()}`);
    return await res.json();
  } catch (err) {
    console.error('getAllCatatan Error:', err);
    return { status: 'error', message: 'Gagal mengambil data catatan.' };
  }
}

export async function getCatatanById(id) {
  try {
    const res = await fetch(`${BASE_URL}/${id}`);
    return await res.json();
  } catch (err) {
    console.error('getCatatanById Error:', err);
    return { status: 'error', message: 'Gagal mengambil data.' };
  }
}

export async function createCatatan(data) {
  try {
    const res = await fetch(BASE_URL, {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(data)
    });
    return await res.json();
  } catch (err) {
    console.error('createCatatan Error:', err);
    return { status: 'error', message: 'Gagal menambahkan catatan.' };
  }
}

export async function updateCatatan(id, data) {
  try {
    const res = await fetch(`${BASE_URL}/${id}`, {
      method: 'PUT',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(data)
    });
    return await res.json();
  } catch (err) {
    console.error('updateCatatan Error:', err);
    return { status: 'error', message: 'Gagal mengubah catatan.' };
  }
}

export async function softDeleteCatatan(id) {
  try {
    const res = await fetch(`${BASE_URL}/${id}`, { method: 'DELETE' });
    return await res.json();
  } catch (err) {
    console.error('softDeleteCatatan Error:', err);
    return { status: 'error', message: 'Gagal menghapus catatan.' };
  }
}

export async function restoreCatatan(id) {
  try {
    const res = await fetch(`${BASE_URL}/${id}/restore`, { method: 'PATCH' });
    return await res.json();
  } catch (err) {
    console.error('restoreCatatan Error:', err);
    return { status: 'error', message: 'Gagal merestore catatan.' };
  }
}

export async function forceDeleteCatatan(id) {
  try {
    const res = await fetch(`${BASE_URL}/${id}/force`, { method: 'DELETE' });
    return await res.json();
  } catch (err) {
    console.error('forceDeleteCatatan Error:', err);
    return { status: 'error', message: 'Gagal menghapus permanen catatan.' };
  }
}
```

File: frontend/assets/js/pages/form.js

```
import {
  getCatatanById,
  createCatatan,
  updateCatatan,
} from '../api/catatanApi.js';

// Ambil ID dari query string (jika ada)
const urlParams = new URLSearchParams(window.location.search);
const id = urlParams.get('id');

document.addEventListener('DOMContentLoaded', () => {
  if (id) {
    // Ubah judul form
    document.getElementById('formTitle').textContent = '📝 Edit Catatan';
    loadCatatan();
  }

  document.getElementById('catatanForm').addEventListener('submit', handleSubmit);
});

async function loadCatatan() {
  const res = await getCatatanById(id);
  if (res.status === 'success') {
    const { isi, kategori, keterangan, status } = res.data;
    document.getElementById('isi').value = isi;
    document.getElementById('kategori').value = kategori ?? '';
    document.getElementById('keterangan').value = keterangan ?? '';
    document.getElementById('status').checked = !!status;
  } else {
    alert('Gagal mengambil data catatan');
    window.location.hash = '#/list';
  }
}

async function handleSubmit(e) {
  e.preventDefault();

  const data = {
    isi: document.getElementById('isi').value.trim(),
    kategori: document.getElementById('kategori').value.trim(),
    keterangan: document.getElementById('keterangan').value.trim(),
    status: document.getElementById('status').checked,
  };

  if (!data.isi) {
    alert('Isi catatan wajib diisi!');
    return;
  }

  let res;
  if (id) {
    res = await updateCatatan(id, data);
  } else {
    res = await createCatatan(data);
  }

  if (res.status === 'success') {
    alert(`Catatan berhasil ${id ? 'diperbarui' : 'ditambahkan'}!`);
    window.location.hash = '#/list';
  } else {
    alert(`Gagal menyimpan catatan: ${res.message}`);
  }
}
```

File: frontend/assets/js/component/navbar.js

```
export function renderNavbar() {
  const navbar = `
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark px-3">
      <a class="navbar-brand" href="#">📒 CatatanApp</a>
      <div class="collapse navbar-collapse">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item">
            <a class="nav-link" href="#/list">📄 Daftar Catatan</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#/form">✏️ Tambah Catatan</a>
          </li>
        </ul>
      </div>
    </nav>
  `;
  document.body.insertAdjacentHTML('afterbegin', navbar);
}
```

File: frontend/assets/js/component/table.js

```
export function renderEmptyTable(selector, colCount, message = 'Tidak
ada data ditemukan') {
  const tbody = document.querySelector(selector);
  if (tbody) {
    tbody.innerHTML = `<tr><td colspan="${colCount}" class="text-center
text-muted">${message}</td></tr>`;
  }
}
```

File: frontend/templates/list.html

```html
<div class="d-flex justify-content-between align-items-center mb-3">
  <h2>📄 Daftar Catatan</h2>
  <a href="#/form" class="btn btn-primary">+ Tambah Catatan</a>
</div>

<!-- 🔍 Filter/Search Bar -->
<div class="row g-2 mb-3">
  <div class="col-md-4">
    <input type="text" class="form-control" id="searchInput" placeholder="Cari
isi/kategori/keterangan..." />
  </div>
  <div class="col-md-3">
    <select class="form-select" id="deletedFilter">
      <option value="0">Aktif</option>
      <option value="1">Terhapus</option>
      <option value="all">Semua</option>
    </select>
  </div>
  <div class="col-md-2">
    <select class="form-select" id="sortBy">
      <option value="updated_at">Sort by: Terakhir Diperbarui</option>
      <option value="id">Sort by: ID</option>
      <option value="kategori">Sort by: Kategori</option>
    </select>
  </div>
  <div class="col-md-2">
    <select class="form-select" id="sortDir">
      <option value="desc">⬇️ Descending</option>
      <option value="asc">⬆️ Ascending</option>
    </select>
  </div>
  <div class="col-md-1">
    <button id="filterBtn" class="btn btn-success w-100">🔍</button>
  </div>
</div>

<!-- 📄 Tabel Catatan -->
<div class="table-responsive">
  <table class="table table-bordered align-middle">
    <thead class="table-dark">
      <tr>
        <th>ID</th>
        <th>Isi</th>
        <th>Kategori</th>
        <th>Status</th>
        <th>Updated</th>
        <th>Aksi</th>
      </tr>
    </thead>
    <tbody id="catatanTableBody">
      <!-- Baris data akan dirender oleh JS -->
    </tbody>
  </table>
</div>

<!-- 🔢 Pagination -->
<nav>
  <ul class="pagination justify-content-center" id="pagination">
    <!-- Halaman akan di-render di sini -->
  </ul>
</nav>
```

File: frontend/templates/form.html

```
<h2 id="formTitle">✏️ Tambah Catatan</h2>

<form id="catatanForm" class="mt-3">
  <div class="mb-3">
    <label for="isi" class="form-label">Isi Catatan</label>
    <textarea class="form-control" id="isi" rows="3" required></textarea>
  </div>

  <div class="mb-3">
    <label for="kategori" class="form-label">Kategori</label>
    <input type="text" class="form-control" id="kategori"
placeholder="Misalnya: Belanja, Kerja, Ide" />
  </div>

  <div class="mb-3">
    <label for="keterangan" class="form-label">Keterangan</label>
    <input type="text" class="form-control" id="keterangan"
placeholder="Keterangan tambahan (opsional)" />
  </div>

  <div class="form-check form-switch mb-3">
    <input class="form-check-input" type="checkbox" id="status" checked />
    <label class="form-check-label" for="status">Aktif</label>
  </div>

  <button type="submit" class="btn btn-primary">💾 Simpan</button>
  <a href="#/list" class="btn btn-secondary">🔙 Kembali</a>
</form>
```

Refactor File: `frontend/assets/js/pages/list.js`

🔄 1. Impor modul di atas

Tambahkan di paling atas `list.js`:

```
import { showToast } from '../component/toast.js';
import { showConfirmModal } from '../component/modal.js';
import { renderEmptyTable } from '../component/table.js';
```

🔄 2. Refactor `renderTable()` bagian jika data kosong

Sebelum:

```
if (rows.length === 0) {
  tbody.innerHTML = `<tr><td colspan="6" class="text-center">Tidak ada data ditemukan</td></tr>`;
  return;
}
```

Sesudah:

```
if (rows.length === 0) {
  renderEmptyTable('#catatanTableBody', 6);
  return;
}
```

🔄 3. Refactor semua `confirm(...)`

Sebelum:

```
if (confirm('Yakin ingin menghapus catatan ini?')) {
  await softDeleteCatatan(id);
  loadData();
}
```

Sesudah:

```
showConfirmModal('Yakin ingin menghapus catatan ini?', async () => {
  const res = await softDeleteCatatan(id);
  if (res.status === 'success') {
    showToast('Catatan berhasil dihapus');
    loadData();
  } else {
    showToast(res.message || 'Gagal menghapus', 'error');
  }
});
```

Lakukan juga untuk **restore** dan **force delete**.

🔄 4. Refactor `loadData()` error handling

Sebelum:

```
document.getElementById('catatanTableBody').innerHTML =
  `<tr><td colspan="6">Gagal memuat data</td></tr>`;
```

Sesudah:

```
renderEmptyTable('#catatanTableBody', 6, '⚠️ Gagal memuat data');
showToast('Gagal memuat data catatan', 'error');
```

Refactor File: `frontend/assets/js/pages/form.js`

🔄 1. Impor toast

Tambahkan:

```
import { showToast } from '../component/toast.js';
```

🔄 2. Ganti alert() jadi showToast()

Contoh:

Sebelum:

```
alert('Isi catatan wajib diisi!');
```

Sesudah:

```
showToast('Isi catatan wajib diisi!', 'error');
```

Sebelum:

```
alert(`Catatan berhasil ${id ? 'diperbarui' : 'ditambahkan'}!`);
```

Sesudah:

```
showToast(`Catatan berhasil ${id ? 'diperbarui' : 'ditambahkan'}!`);
```

Sebelum:

```
alert(`Gagal menyimpan catatan: ${res.message}`);
```

Sesudah:

```
showToast(`Gagal menyimpan catatan: ${res.message}`, 'error');
```

Tambahkan komponen loader animasi modular

📁 Lokasi File

```
frontend/
└── assets/
    └── js/
        └── component/
            └── loader.js ✅
```

frontend/assets/js/component/loader.js

```
export function showLoader() {
  let loader = document.getElementById('global-loader');
  if (!loader) {
    loader = document.createElement('div');
    loader.id = 'global-loader';
    loader.innerHTML = `
      <div class="loader-overlay d-flex justify-content-center align-items-center">
        <div class="spinner-border text-primary" role="status">
          <span class="visually-hidden">Loading...</span>
        </div>
      </div>
    `;
    document.body.appendChild(loader);
  }
  loader.style.display = 'flex';
}

export function hideLoader() {
  const loader = document.getElementById('global-loader');
  if (loader) loader.style.display = 'none';
}
```

Tambahkan CSS di `index.html`

```
<style>
  .loader-overlay {
    position: fixed;
    inset: 0;
    z-index: 1050;
    background-color: rgba(0, 0, 0, 0.3);
    display: none;
  }
</style>
```

Contoh Pemakaian Loader

Contoh Pemakaian di list.js

```
import { showLoader, hideLoader } from '../component/loader.js';

async function loadData() {
  showLoader();
  try {
    const res = await getAllCatatan(params);
    renderTable(res.data);
  } catch (err) {
    showToast('Gagal memuat data', 'error');
  } finally {
    hideLoader();
  }
}
```

Contoh Pemakaian di form.js

```
import { showLoader, hideLoader } from '../component/loader.js';

async function handleSubmit(e) {
  e.preventDefault();
  showLoader();
  const data = { ... };

  try {
    const res = id
      ? await updateCatatan(id, data)
      : await createCatatan(data);

    if (res.status === 'success') {
      showToast('Berhasil disimpan');
      window.location.hash = '#/list';
    } else {
      showToast(res.message, 'error');
    }
  } catch (err) {
    showToast('Gagal menyimpan data', 'error');
  } finally {
    hideLoader();
  }
}
```

Langkah Uji Coba Sistem SPA Catatan

1. 🚀 Jalankan backend API Node.js

- Pastikan kamu sudah `npm run dev` atau `node index.js`
- Akses: `http://localhost:7000/api/v1/catatan`

2. 🧱 Buka `index.html` di browser

> 💡 Disarankan pakai **Live Server** VSCode atau http-server

3. 📄 Akses Halaman List (`#/list`)

- Lihat daftar catatan
- Coba filter, sort, dan pagination
- Cek apakah ada loading spinner (`showLoader`)
- Jika kosong → pastikan muncul `renderEmptyTable()`

4. ✏️ Tambah Catatan (`#/form`)

- Isi form
- Submit → muncul toast (`showToast`)
- Redirect ke `#/list`

5. 📝 Edit Catatan

- Klik tombol edit → URL `#/form?id=xxx`
- Form otomatis terisi
- Edit & simpan → muncul toast → redirect

6. 🗑️ Hapus (soft delete)

- Klik tombol hapus
- Muncul modal konfirmasi (`showConfirmModal`)
- Setelah sukses → muncul toast

7. ♻️ Restore / Force Delete

- Tampilkan catatan `deleted=1`
- Klik tombol Restore atau Force Delete
- Modal + toast jalan? Reload dan lihat data hilang atau kembali

8. 🔁 Refresh browser

- Pastikan router tetap bekerja dan tetap ke halaman terakhir (`#/list`, `#/form`, dll.)

Thank you