Berikut ini adalah **komponen Svelte lengkap** yang mereplikasi fungsionalitas modal Pilih Barang dari Blade dan JavaScript kamu:

**✅ PilihBarang.svelte**

svelte

CopyEdit

<script>

export let databarang = [];

let search\_filter = '0';

let filterNoBarang = '';

let filterNamaBarang = '';

let rowsPerPage = 10;

let currentPage = 1;

let selectedItems = new Set();

let filteredData = [];

$: filteredData = databarang.filter(item => {

return item.no\_barang.toLowerCase().includes(filterNoBarang.toLowerCase()) &&

item.nama\_barang.toLowerCase().includes(filterNamaBarang.toLowerCase());

});

$: paginatedData = filteredData.slice((currentPage - 1) \* rowsPerPage, currentPage \* rowsPerPage);

$: totalPages = Math.ceil(filteredData.length / rowsPerPage);

function changePage(page) {

currentPage = page;

}

function toggleAll(e) {

if (e.target.checked) {

filteredData.forEach(item => selectedItems.add(item.id));

} else {

selectedItems.clear();

}

}

function toggleItem(id) {

if (selectedItems.has(id)) {

selectedItems.delete(id);

} else {

selectedItems.add(id);

}

}

function tambahBarangTerpilih() {

const selected = databarang.filter(d => selectedItems.has(d.id));

console.log("Barang terpilih:", selected);

// Emit event ke parent jika diperlukan

// createEventDispatcher bisa digunakan

}

</script>

<!-- Modal -->

<div class="modal fade show d-block" tabindex="-1" role="dialog" aria-labelledby="modalBarangLabel">

<div class="modal-dialog modal-lg" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title">Pilih Barang</h5>

<button type="button" class="close" on:click={() => document.body.classList.remove('modal-open')}>

<span>&times;</span>

</button>

</div>

<div class="modal-body">

<!-- Filter Radio -->

<div class="form-inline mb-3">

<label class="form-check form-check-inline">

<input class="form-check-input" type="radio" bind:group={search\_filter} value="0">

<span class="form-check-label">Semua</span>

</label>

<label class="form-check form-check-inline">

<input class="form-check-input" type="radio" bind:group={search\_filter} value="1">

<span class="form-check-label">Kategori Barang</span>

</label>

<label class="form-check form-check-inline">

<input class="form-check-input" type="radio" bind:group={search\_filter} value="2">

<span class="form-check-label">Tipe Persediaan</span>

</label>

</div>

<!-- Filter Input -->

<div class="form-inline mb-3">

<input type="text" class="form-control form-control-sm mr-2" placeholder="Cari No Barang" bind:value={filterNoBarang}>

<input type="text" class="form-control form-control-sm mr-2" placeholder="Cari Nama Barang" bind:value={filterNamaBarang}>

<button class="btn btn-sm btn-primary" on:click={() => { currentPage = 1; }}>Filter</button>

</div>

<!-- Table -->

<div class="table-responsive">

<table class="table table-striped table-bordered table-hover table-center mb-0">

<thead class="thead-dark">

<tr>

<th class="text-center" style="width: 10%">

<input type="checkbox" on:change={toggleAll}>

</th>

<th style="padding: 4px;">No. Barang</th>

<th style="padding: 4px;">Nama Barang</th>

<th style="padding: 4px;">Satuan</th>

<th style="padding: 4px;">Kuantitas</th>

</tr>

</thead>

<tbody>

{#each paginatedData as item}

<tr>

<td class="text-center">

<input type="checkbox" checked={selectedItems.has(item.id)} on:change={() => toggleItem(item.id)}>

</td>

<td>{item.no\_barang}</td>

<td>{item.nama\_barang}</td>

<td>{item.satuan}</td>

<td>{item.kuantitas\_saldo\_awal}</td>

</tr>

{/each}

</tbody>

</table>

</div>

<!-- Pagination & Limit -->

<div class="d-flex justify-content-between align-items-center mt-3">

<div class="form-inline">

<label class="mr-2">Tampilkan</label>

<select bind:value={rowsPerPage} class="form-control form-control-sm" on:change={() => currentPage = 1}>

<option value="10">10</option>

<option value="25">25</option>

<option value="50">50</option>

<option value="100">100</option>

</select>

<span class="ml-2">data per halaman</span>

</div>

<nav>

<ul class="pagination pagination-sm mb-0">

{#each Array(totalPages) as \_, i}

<li class="page-item {i + 1 === currentPage ? 'active' : ''}">

<a class="page-link" href="#" on:click|preventDefault={() => changePage(i + 1)}>{i + 1}</a>

</li>

{/each}

</ul>

</nav>

</div>

</div>

<div class="modal-footer">

<button class="btn btn-success" on:click={tambahBarangTerpilih}>Tambah ke Form</button>

</div>

</div>

</div>

</div>

<style>

.modal {

background-color: rgba(0, 0, 0, 0.5);

}

</style>

**⚙️ Penyesuaian Penting:**

* Komponen ini **tidak tergantung jQuery atau Bootstrap JS**.
* Kamu bisa menggantikan modal show/hide menggunakan bind:this atau komponen modal terpisah jika lebih kompleks.
* Gunakan createEventDispatcher() dari Svelte jika kamu ingin mengirim data barang terpilih ke komponen parent.

Kalau kamu ingin saya bantu integrasikan komponen ini ke dalam form utama Svelte atau gabung dengan Laravel backend (misalnya lewat axios atau fetch), tinggal bilang ya.

**You said:**

kalau panggil dari javascript di html

**ChatGPT said:**

Kalau kamu ingin **memanggil atau membuka komponen Svelte dari JavaScript di HTML biasa**, kamu bisa lakukan ini dengan:

**✅ Langkah-langkah Memanggil Komponen Svelte dari HTML/JavaScript**

**1. Pastikan Komponen Svelte Sudah Di-*compile***

* Compile Svelte ke JavaScript menggunakan bundler seperti **Vite**, **Rollup**, atau **Webpack**.
* Misalnya output dari kompilasi adalah PilihBarang.js.

**2. Tambahkan Tag Script di HTML**

html

CopyEdit

<!-- Tambahkan setelah build -->

<script type="module" src="/path/to/PilihBarang.js"></script>

**3. Siapkan Tempat untuk Render Komponen**

Letakkan div kosong di tempat komponen akan dirender:

html

CopyEdit

<div id="modalBarangContainer"></div>

**4. Instansiasi Komponen Svelte dari JS**

Misalnya di dalam script tag:

html

CopyEdit

<script type="module">

import PilihBarang from '/path/to/PilihBarang.svelte';

// Data bisa datang dari backend (dari Blade misalnya)

const dataBarang = @json($databarang); // Blade -> Laravel inject

// Render Svelte component ke dalam div

const modalApp = new PilihBarang({

target: document.getElementById('modalBarangContainer'),

props: {

databarang: dataBarang

}

});

// Optional: trigger fungsi dari luar

window.showModalBarang = () => {

document.getElementById('modalBarangContainer').style.display = 'block';

};

</script>

**5. Panggil dari Tombol HTML Biasa**

html

CopyEdit

<button onclick="showModalBarang()">Pilih Barang</button>

**📌 Catatan Penting**

* Svelte **tidak otomatis menampilkan modal** seperti Bootstrap. Kamu bisa:
  + Gunakan komponen visible → true/false untuk show/hide modal
  + Atur CSS display/visibility manual
* Kalau ingin memakai Bootstrap-style modal, kamu bisa *mix* dengan Bootstrap CSS dan buat binding showModal di komponen.

Kalau kamu ingin:  
✅ Modal bisa ditutup dari dalam komponen  
✅ Emit data barang terpilih ke JS luar atau ke Laravel  
✅ Pakai Alpine atau Livewire untuk integrasi  
Aku bisa bantu juga buatkan contohnya. Mau lanjut ke salah satu?

**You said:**

triger tampilkan modalnya untuk panggil dari app.js

**ChatGPT said:**

Untuk men-trigger tampilan modal dari app.js (atau JavaScript eksternal lainnya) dan memanggil komponen Svelte PilihBarang, kamu cukup ikuti pola berikut:

**✅ 1. Tambahkan Binding visible di Komponen Svelte**

Pertama, di PilihBarang.svelte, buat **prop visible** untuk mengontrol tampilan modal:

svelte

CopyEdit

<script>

export let databarang = [];

export let visible = false; // kontrol tampil/tidaknya modal

</script>

{#if visible}

<div class="modal fade show d-block" tabindex="-1" role="dialog" aria-labelledby="modalBarangLabel">

<!-- isi modal -->

</div>

{/if}

**✅ 2. Render Komponen dan Simpan Instance-nya**

Di app.js, render komponen ke elemen #modalBarangContainer dan simpan ke variabel global agar bisa diakses di mana saja:

js

CopyEdit

import PilihBarang from './PilihBarang.svelte';

const dataBarang = window.dataBarang || []; // data bisa dari Laravel

// Simpan instance komponen ke window agar bisa diakses global

window.modalBarangComponent = new PilihBarang({

target: document.getElementById('modalBarangContainer'),

props: {

databarang: dataBarang,

visible: false

}

});

**✅ 3. Buat Fungsi Trigger dari app.js**

Masih di app.js, tambahkan fungsi trigger global:

js

CopyEdit

window.showModalBarang = function () {

// set ulang komponen agar visible = true

modalBarangComponent.$set({ visible: true });

};

**✅ 4. HTML untuk Memicu dari Tombol**

html

CopyEdit

<button onclick="showModalBarang()">Pilih Barang</button>

<div id="modalBarangContainer"></div>

**🔄 Kalau Ingin Tutup Modal dari Svelte**

Di PilihBarang.svelte, tambahkan tombol close untuk set visible = false:

svelte

CopyEdit

<button type="button" class="close" on:click={() => visible = false}>

<span>&times;</span>

</button>

Atau buatkan event keluar ke JS luar kalau ingin komponen tidak dikelola internal saja.

**🔁 Alternatif: Trigger lewat Event Dispatch (Opsional)**

Kalau kamu ingin PilihBarang.svelte memanggil fungsi JS eksternal setelah memilih barang, gunakan:

svelte

CopyEdit

import { createEventDispatcher } from 'svelte';

const dispatch = createEventDispatcher();

function tambahBarangTerpilih() {

const selected = databarang.filter(d => selectedItems.has(d.id));

dispatch('barangDipilih', { selected });

}

Dan di app.js:

js

CopyEdit

modalBarangComponent.$on('barangDipilih', (e) => {

console.log('Barang dipilih dari Svelte:', e.detail.selected);

});

Kalau kamu ingin komponen ini **terintegrasi dengan Laravel Blade dan Livewire**, atau mau fetch data databarang dari API, tinggal beri tahu aja.