Table of Contents

- Solusi yang sudah kamu terapkan di akhir PDF:
- - Langkah-langkah:
- 1. Update DatasetController@index
 - 2. Update Blade View
- Best Practice: Simpan notifikasi upload di database per user
 - **Langkah implementasinya:**
 - 1. Buat Migration:
- 2. Simpan data upload saat proses import:
 - 3. Ambil notifikasi di controller dataset:
 - 4. Tampilkan di Blade:
- Reuntungan:
- 1. Migration (buat file upload_logs)
- **2.** Model UploadLog
- **3.** Simpan log di Controller saat proses upload
- 4. Ambil log di controller dataset
- **7** 5. Tampilkan di dataset.blade.php
- \begin{align*} \begin{align*} \begin{align*} \begin{align*} \text{Hasilnya:} \end{align*}
- Perbedaan:
 - ✓ Cara Alternatif (tanpa database)
 - 1. 🔁 Ubah dari flash() ke put()
 - 2. Tampilkan di Blade (tetap sama)
 - 3. X Tambahkan tombol untuk "hapus notifikasi"

- **@** Keunggulan metode ini:
- Solusi Paling Mudah: Pindahkan logika ke controller index()
 - 🔧 Jadi yang perlu kamu lakukan:
- 1. Tambahkan ke controller index():
 - 2. Di Blade (dataset.blade.php) ubah:
- Opsional: Hapus riwayat saat diinginkan
- Kesimpulan:



chat baca seluruh isi dari pdf ini, lalu baca pesan terakhirnya

Terima kasih sudah memberikan semua detail dan file pendukung. Aku sudah membaca seluruh isi dari PDF *DeepSeek - Into the Unknown* dan juga pesan terakhirnya. Berdasarkan isinya, kamu sudah menyusun dan mengimplementasikan sistem impor data Excel untuk kebutuhan klasifikasi Naive Bayes di Laravel, termasuk:

- Struktur tabel imported_data
- Form upload multiple file
- Preview nama file, ukuran, ekstensi, dan ikon
- Mapping dinamis kolom Excel
- Transformasi data enum dan diskritisasi
- Controller dengan error handling dan logging
- View dengan modal dan JavaScript interaktif

Pesan terakhir di PDF adalah:

"ok, aku sudah mencoba untuk mengupload tapi masih tidak bisa, tidak ada error notifikasi yang keluar, apa karena nama kolom pada excel yang kapital?"

Dan kamu benar—penyebabnya karena nama kolom tidak dikenali akibat tidak sesuai format mapping. Misalnya: RWKriteria (tanpa spasi) tidak cocok dengan RW dan Kriteria yang dipisah.

Solusi yang sudah kamu terapkan di akhir PDF:

- Menambahkan mapping alternatif untuk berbagai variasi penulisan kolom.
- Normalisasi header kolom menggunakan:

```
php

preg_replace('/\s+/', ' ', trim(strtoupper($header)));
```

- Validasi kolom wajib (RW , USIA , KETERANGAN)
- Logging dan notifikasi error bila file tidak valid atau kolom tidak dikenali.

K Rekomendasi tambahan agar lebih stabil:

1. Pisahkan kata dalam kolom Excel jika tanpa spasi (misal RWKriteria → RW dan Kriteria) — bisa diatasi dengan:

```
splitHeaders = preg_split('/(?=[A-Z])/', $header, -1, PREG_SPLIT_NO_EN
```

Tapi ini hanya jika memang kamu ingin otomatis membagi header tanpa spasi. Kalau tidak, pastikan user pakai template Excel yang benar.

2. Tambahkan contoh file template download di tampilan agar user tidak salah format:

```
blade
<a href="{{ asset('template/template_dataset.xlsx') }}" class="text-blue"</pre>
```

3. Di controller, kamu bisa tambahkan alert jika tidak ada baris berhasil diimport:

```
if ($importedCount === 0) {
    return back()->with('error', 'Tidak ada data yang berhasil diimport. }
}
```

Kalau kamu mau, aku bisa bantu juga:

- Menambahkan tombol "Download Template"
- Menambahkan fitur "preview isi data" sebelum confirm upload
- Membuat notifikasi lebih informatif di Blade

Mau lanjut ke mana nih? Mau aku bantu cek file Excel kamu langsung juga boleh, tinggal upload saja.

```
tidak perlu itu semua sudah berhasil,
sekarang bagaimana cara
menampilkannya dalam bentuk table
pada resources/views/pages/naive-
bayes/dataset.blade.php
rapikan ini:
@extends('layout.app')
@section('title', 'Dataset')
@section('styles')
  <style>
    /* File Upload Dropzone */
    .file-upload {
      transition: all 0.3s ease;
    }
    .file-upload:hover {
      background-color: #f9fafb;
    }
    /* File Preview Container */
    #file-preview-container {
      transition: all 0.3s ease;
    /* File List Scrollbar */
    #file-list::-webkit-scrollbar {
      width: 6px;
    }
    #file-list::-webkit-scrollbar-track {
      background: #f1f1f1;
    }
    #file-list::-webkit-scrollbar-thumb {
      background: #c1c1c1;
      border-radius: 3px;
    }
    #file-list::-webkit-scrollbar-
thumb:hover {
      background: #a8a8a8;
    }
  </style>
@endsection
@section('content')
  <header class="bg-white shadow flex</pre>
flex-col">
    <div class="max-w-7xl w-full mx-auto</pre>
py-6 px-4 sm:px-6 lg:px-8 flex flex-row
justify-between">
```

```
<h2 class="font-semibold text-xl
text-gray-800 flex items-center">
        Dataset
      </h2>
      <but
        class="flex items-center justify-
center w-1/2 px-5 py-2 text-sm text-gray-
50 transition-colors duration-200 bg-blue-
500 border rounded-lg gap-x-2 sm:w-auto
dark:hover:bg-gray-800 dark:bg-gray-900
hover:bg-blue-700 dark:text-gray-200
dark:border-gray-700"
onclick="openModal('importModal')">
        <svg width="20" height="20"</pre>
viewBox="0 0 20 20" fill="none"
xmlns="http://www.w3.org/2000/svg">
          <g clip-
path="url(#clip0_3098_154395)">
            <path
              d="M13.3333
13.3332L9.99997 9.9999M9.99997
9.9999L6.66663 13.3332M9.99997
9.9999V17.4999M16.9916 15.3249C17.8044
14.8818 18.4465 14.1806 18.8165
13.3321C19.1866 12.4835 19.2635 11.5359
19.0351 10.6388C18.8068 9.7417 18.2862
8.94616 17.5555 8.37778C16.8248 7.80939
15.9257 7.50052 15 7.4999H13.95C13.6977
6.52427 13.2276 5.61852 12.5749
4.85073C11.9222 4.08295 11.104 3.47311
10.1817 3.06708C9.25943 2.66104 8.25709
2.46937 7.25006 2.50647C6.24304 2.54358
5.25752 2.80849 4.36761 3.28129C3.47771
3.7541 2.70656 4.42249 2.11215
5.23622C1.51774 6.04996 1.11554 6.98785
0.935783 7.9794C0.756025 8.97095
0.803388 9.99035 1.07431 10.961C1.34523
11.9316 1.83267 12.8281 2.49997 13.5832"
              stroke="currentColor"
stroke-width="1.67" stroke-
linecap="round" stroke-linejoin="round"
/>
          </g>
          <defs>
            <clipPath
id="clip0_3098_154395">
              <rect width="20"
height="20" fill="white" />
            </clipPath>
          </defs>
        </svg>
        <span>Import Dataset</span>
```

```
</button>
    </div>
    {{-- resources/views/partials/upload-
notification.blade.php --}}
    @if (session('uploaded_files'))
      <div class="mb-4 p-4 bg-blue-50</pre>
border border-blue-200 rounded-lg mx-
4">
        <h3 class="font-medium text-
blue-800 mb-2">File yang berhasil di
upload:</h3>
        ul class="list-disc list-inside
space-y-1">
          @foreach
(session('uploaded_files') as $file)
            {{
$file }}
          @endforeach
        </div>
    @endif
    @if (session('failed_files'))
      <div class="bg-yellow-50 border-l-4</pre>
border-yellow-400 p-4 mb-4">
        <div class="flex">
          <div class="flex-shrink-0">
            <svg class="h-5 w-5 text-
yellow-400"
xmlns="http://www.w3.org/2000/svg"
viewBox="0 0 20 20"
              fill="currentColor">
              <path fill-rule="evenodd"</p>
                 d="M8.257 3.099c.765-
1.36 2.722-1.36 3.486 0l5.58 9.92c.75
1.334-.213 2.98-1.742 2.98H4.42c-1.53 0-
2.493-1.646-1.743-2.98l5.58-9.92zM11
13a1 1 0 11-2 0 1 1 0 012 0zm-1-8a1 1 0
00-1 1v3a1 1 0 002 0V6a1 1 0 00-1-1z"
                 clip-rule="evenodd" />
            </svg>
          </div>
          <div class="ml-3">
            <h3 class="text-sm font-
medium text-yellow-800">
               Beberapa file gagal
diproses
            </h3>
            <div class="mt-2 text-sm text-
yellow-700">
              ul class="list-disc pl-5"
space-y-1">
                 @foreach
(session('failed_files') as $file)
```

```
<
                     <strong>{{
$file['name'] }}</strong>: {{ $file['error'] }}
                   @endforeach
              </div>
          </div>
        </div>
      </div>
    @endif
  </header>
  <main class="bg-gray-100 flex-grow">
    <div class="flex flex-col m-6">
      <div class="overflow-hidden border
border-gray-200 dark:border-gray-700
md:rounded-lg overflow-x-scroll ">
        @if ($importedData->isEmpty())
          <!-- Tampilkan pesan jika data
kosong -->
          <div class="p-6 text-center text-
gray-500 dark:text-gray-400">
            Tidak ada data yang
diimport.
          </div>
        @else
          <table class="min-w-full
divide-y divide-gray-200 dark:divide-gray-
700 table-auto">
            <thead class="bg-gray-50</pre>
dark:bg-gray-800">
                @if ($importedData-
>isNotEmpty())
                  {{-- @foreach
(json_decode($importedData->first()-
>row_data, true) as $key => $value) --}}
                  {{--@foreach
(is_array($importedData->first()-
>row_data) ? $importedData->first()-
>row_data:json_decode($importedData-
>first()->row_data, true) as $key =>
$value) --}}
                   @foreach
($importedData->first()->row_data as
$key => $value)
                     <th scope="col"
                       class="py-3.5 px-4
text-sm font-normal text-left text-gray-500
dark:text-gray-400">
                       {{ $key }}
```

```
@endforeach
                @endif
              </thead>
            <tbody class="bg-white
divide-y divide-gray-200 dark:divide-gray-
700 dark:bg-gray-900">
              @foreach ($importedData
as $index => $data)
                {{-- @foreach
(json_decode($data->row_data, true) as
$value) --}}
                  {{-- @foreach
(is_array($importedData->first()-
>row_data) ? $importedData->first()-
>row_data:json_decode($importedData-
>first()->row_data, true) as $key =>
$value) --}}
                  @foreach
($importedData->first()->row_data as
$key => $value)
                    <td class="px-4 py-4
text-sm text-gray-500 dark:text-gray-300
whitespace-nowrap">
                      {{ $value }}
                    @endforeach
                @endforeach
            <!-- Tampilkan navigasi
pagination -->
          <div class="mt-4">
            {{ $importedData->links() }}
          </div>
        @endif
      </div>
    </div>
  </main>
@endsection
@section('modals')
  <div id="importModal"
    class="fixed inset-0 bg-black bg-
opacity-50 justify-center items-center
opacity-0 scale-95 hidden transition-all
duration-300 ease-in-out px-5">
    <div
      class="bg-white p-5 rounded-lg w-
```

```
full sm:w-full sm:max-w-md sm:p-6
sm:align-middle opacity-0 scale-95
transition-all duration-300 ease-in-out">
      <h2 class="text-lg font-medium"
leading-6 text-gray-800 capitalize"
id="title_modal">Import Dataset</h2>
      <form action="{{ route('naive-</pre>
bayes.dataset.import') }}"
method="POST"
enctype="multipart/form-data"
        class="mt-4">
        @csrf
        <div class="flex items-center</pre>
justify-center w-full">
          <label for="dropzone-file"</pre>
            class="flex flex-col items-
center justify-center w-full h-64 border-2
border-gray-300 border-dashed rounded-
lg cursor-pointer bg-gray-50 hover:bg-
gray-100">
            <div class="flex flex-col
items-center justify-center pt-5 pb-6"
id="file-upload-placeholder">
              <svg class="w-10 h-10 mb-
4 text-green-500"
xmlns="http://www.w3.org/2000/svg"
                viewBox="0 0 24 24"
fill="currentColor">
                <path
                  d="M19.5 14.5v-
5.379a3 3 0 0 0-.879-2.121l-3.62-3.62A3 3 0
0 0 12.879 2.5H5.5A1.5 1.5 0 0 0 4 4v16a1.5
1.5 0 0 0 1.5 1.5h13a1.5 1.5 0 0 0 1.5-
1.5V14.5zM12 4v5h5v-1.5a1.5 1.5 0 0 0-1.5-
1.5H12z" />
              </svg>
              text-gray-500"><span class="font-
semibold">Click to upload</span>
              500">Excel files only (.xlsx, .xls)
            </div>
            <div id="file-preview-
container" class="hidden w-full px-4">
              <div class="border
rounded-lg divide-y divide-gray-200">
                <div class="px-4 py-2 bg-
gray-50 font-medium text-sm text-gray-
500">
                   Selected Files
                </div>
                <div id="file-list"
class="max-h-48 overflow-y-auto"></div>
              </div>
```

</div>

```
<input id="dropzone-file"
type="file" name="files[]" multiple
accept=".xlsx,.xls"
               class="hidden" />
           </label>
        </div>
        <div class="mt-5 flex flex-row
justify-between">
           <button type="button"</pre>
onclick="closeModal('importModal')"
             class="bg-gray-500 text-
white px-4 py-2
rounded">Cancel</button>
          <button type="submit"</pre>
id="upload-button"
             class="bg-blue-600 rounded-
lg hover:bg-blue-700 text-white px-4 py-2
hidden">Upload All</button>
        </div>
      </form>
    </div>
  </div>
@endsection
@section('scripts')
  <script>
    function openModal(id) {
      let modal =
document.getElementById(id);
      let modalBox =
modal.querySelector('div');
      modal.classList.remove('hidden');
      modal.classList.add('flex');
      setTimeout(() => {
        modal.classList.remove('opacity-
0', 'scale-95');
modalBox.classList.remove('opacity-0',
'scale-95');
        modalBox.classList.add('scale-
100');
      }, 10);
      modal.addEventListener('click',
function(event) {
        if (event.target === modal) {
           closeModal(id);
        }
      });
    }
    function closeModal(id) {
      let modal =
document.getElementById(id);
```

```
let modalBox =
modal.querySelector('div');
      modalBox.classList.add('opacity-0',
'scale-95');
      modalBox.classList.remove('scale-
100');
      setTimeout(() => {
        modal.classList.add('hidden',
'opacity-0');
        resetFileInput();
      }, 300);
    }
    function resetFileInput() {
document.getElementById('dropzone-
file').value = ";
      document.getElementById('file-
preview-
container').classList.add('hidden');
      document.getElementById('file-
upload-
placeholder').classList.remove('hidden');
document.getElementById('upload-
button').classList.add('hidden');
      document.getElementById('file-
list').innerHTML = ";
    }
    function formatFileSize(bytes) {
      if (bytes === 0) return '0 Bytes';
      const k = 1024;
      const sizes = ['Bytes', 'KB', 'MB', 'GB'];
      const i = Math.floor(Math.log(bytes)
/ Math.log(k));
      return parseFloat((bytes /
Math.pow(k, i)).toFixed(2)) + ' ' + sizes[i];
    }
document.getElementById('dropzone-
file').addEventListener('change',
function(event) {
      const files = event.target.files;
      const fileList =
document.getElementById('file-list');
      const placeholder =
document.getElementById('file-upload-
placeholder');
      const previewContainer =
document.getElementById('file-preview-
container');
```

```
const uploadButton =
document.getElementById('upload-
button');
      if (files.length > 0) {
placeholder.classList.add('hidden');
previewContainer.classList.remove('hidde
n');
uploadButton.classList.remove('hidden');
        fileList.innerHTML = ";
        Array.from(files).forEach(file => {
          const fileItem =
document.createElement('div');
          fileItem.className = 'px-4 py-2
flex items-center';
          fileItem.innerHTML =
          <div class="flex-shrink-0 text-
green-500">
            <svg
xmlns="http://www.w3.org/2000/svg"
class="h-6 w-6" viewBox="0 0 24 24"
fill="currentColor">
              <path d="M19.5 14.5v-
5.379a3 3 0 0 0-.879-2.121l-3.62-3.62A3 3 0
0 0 12.879 2.5H5.5A1.5 1.5 0 0 0 4 4v16a1.5
1.5 0 0 0 1.5 1.5h13a1.5 1.5 0 0 0 1.5-
1.5V14.5zM12 4v5h5v-1.5a1.5 1.5 0 0 0-1.5-
1.5H12z"/>
            </svg>
          </div>
          <div class="ml-3 flex-1 min-w-
0">
            <p class="text-sm font-
medium text-gray-900
truncate">${file.name}
            500">${formatFileSize(file.size)} •
${file.name.split('.').pop().toUpperCase()}
</div>
          fileList.appendChild(fileItem);
        });
      } else {
        resetFileInput();
      }
    });
  </script>
```

```
Dataset View

@extends('layout.app')

@section('title', 'Dataset')

@section('styles')

style>
.file-upload {
    transition: all 0.3s ease;

.file-upload:hover {
    background-color: #f9fafb;
    }

#file-preview-container {
    transition: all 0.3s ease;
}

#file-preview-container {
    transition: all 0.3s ease;
}
```

Tampilan file dataset.blade.php milikmu sekarang sudah dirapikan, dengan:

- Struktur header dan tombol upload yang bersih
- Tabel data yang hanya menampilkan kolom yang relevan (id , user_email , created_at , dan updated_at disembunyikan)
- Warna latar belakang berbeda untuk kolom keterangan (kuning) dan lainnya (biru muda)
- Penanganan pagination Laravel
- Style dan struktur HTML yang lebih konsisten dan readable

Kalau kamu ingin menambahkan fitur filter, export, atau tampilan lebih dinamis (misal pakai Livewire atau DataTables), tinggal bilang aja ya! Mau aku bantu?

Great, kita bisa batasi data yang ditampilkan langsung dari controller dengan memilih hanya kolom yang dibutuhkan menggunakan select() pada query Eloquent. Jadi Blade tidak perlu lagi menyaring kolom seperti id , user_email , created_at , dll.

Langkah-langkah:

1. Update DatasetController@index

Buka app/Http/Controllers/DatasetController.php lalu ubah method index() jadi seperti ini:

```
public function index()
{
    $columns = [
        'rw', 'kriteria', 'usia', 'nama',
        'tanggungan_kepala_keluarga', 'lansia', 'anak_wajib_sekolah',
        'penghasilan_kepala_keluarga', 'status_bpjs', 'tipe_kendaraan',
        'sumber_air', 'tipe_jamban', 'status_kepemilikan_bangunan',
        'bahan_lantai', 'bahan_dinding', 'kategori_luas_bangunan', 'keteranga'
];

$importedData = ImportedData::select($columns)->paginate(10);
    return view('pages.naive-bayes.dataset', compact('importedData'));
}
```

Ini akan memastikan hanya kolom tersebut yang diambil dari database, dan otomatis tidak ada id, created_at, dsb.

2. Update Blade View

Sekarang kamu bisa hapus kondisi @if (!in_array(...)) di Blade, dan ganti jadi seperti ini:

Dan untuk row-nya:

Kamu juga bisa tetap kasih warna khusus untuk kolom keterangan kalau mau, tinggal tambahkan kondisi if (\$key === 'keterangan') di loop baris.

Kalau kamu ingin menjadikan daftar kolom itu sebagai variabel global (biar reusable), bisa juga aku bantu atur di config atau constant di model. Mau begitu?

> ok ok, sekarang mari ke bagian resources/views/pages/naive-bayes/initialprocess.blade.php, sebenarnya untuk bagian ini hanya menampilkan hal yang sama seperti resources/views/pages/naivebayes/dataset.blade.php hanya saja dengan memisahkan mana bagian "atribut pendukung" dengan mana "label target" untuk kolom 'rw', 'kriteria', 'usia', 'nama', 'tanggungan_kepala_keluarga', 'lansia', 'anak_wajib_sekolah', 'penghasilan_kepala_keluarga', 'status_bpjs', 'tipe_kendaraan', 'sumber_air', 'tipe_jamban', 'status_kepemilikan_bangunan', 'bahan_lantai', 'bahan_dinding', 'kategori_luas_bangunan' masukkan kedalam bagiann dari "atribut pendukung" sedangkan untuk 'keterangan' masukkan kedalam "label target"

⟨/> Initial Process View

- Beres! Tampilan untuk initial-process.blade.php sudah dibuat berdasarkan struktur yang kamu minta:
 - Data terbagi menjadi dua bagian:
 - Atribut Pendukung: menampilkan semua kolom kecuali keterangan
 - Label Target: hanya menampilkan kolom keterangan
- Ditampilkan dalam dua kolom (grid-cols-2) agar responsif dan terorganisir

Kalau kamu ingin menambahkan filter, sort, atau tombol aksi selanjutnya (misal: "Proses Naive Bayes"), tinggal bilang aja. Mau oh bukan begitu maksud ku, harusnya seperti ini :

Rw Kriteria Usia Nama Tanggungan Kepala Keluarga Lansia Anak Wajib Sekolah Penghasilan Kepala Keluarga Status Bpjs Tipe Kendaraan Sumber Air Tipe Jamban Status Kepemilikan Bangunan Bahan Lantai Bahan Dinding Kategori Luas Bangunan Keterangan Atribut Pendukung Label Target RW 3 usia_produktif 54 AC Sedikit ada tidak_ada Tinggi Punya Asuransi Sepeda Air Ledeng Tidak Ada Milik Sendiri Ubin Kayu Kecil penerima RW 3 usia_produktif 38 AC Sedikit tidak ada ada Rendah Tidak Ada Asuransi Mobil Pribadi Pompa Air Jamban Bersama Lainnya Tanah Kayu Besar penerima RW 3 usia_produktif 34 AG Sedang tidak_ada ada Sedang Tidak Ada Asuransi Motor Pompa Air Jamban Keluarga Kontrak Keramik Tembok Kecil penerima

Sudah diperbarui! Sekarang tampilan tabel di initialprocess.blade.php menampilkan:

- Semua kolom atribut pendukung di sebelah kiri
- Kolom keterangan di sebelah kanan sebagai Label Target
- Baris header kedua memisahkan dengan label "Atribut Pendukung" dan "Label Target"

Siap untuk proses analisis atau klasifikasi berikutnya. Mau lanjut ke fitur Naive Bayes selanjutnya?

menarik, tapi aku lebih suka style yang seperti ini : @extends('layout.app')

@section('title', 'Initial Process')

```
gray-800">Initial Process</h2>
  </div>
</header>
<main class="bg-gray-100 p-6">
  <div class="overflow-x-auto border</pre>
border-gray-200 rounded-lg bg-white">
    @if ($importedData->isEmpty())
     <div class="text-center text-gray-</pre>
500 p-6">Tidak ada data yang diimport.
</div>
   @else
     <table class="min-w-full table-auto
divide-y divide-gray-200">
        <thead class="bg-gray-50">
          @foreach
(array_keys($importedData->first()-
>toArray()) as $col)
              <th class="px-4 py-2 text-
left text-xs font-semibold text-gray-600
uppercase">
               {{ Str::headline($col) }}
             @endforeach
          <th colspan="16" class="px-4"
py-2 text-left text-xs font-medium text-
blue-700">Atribut Pendukung
            <th class="px-4 py-2 text-left
text-xs font-medium text-yellow-
700">Label Target
         </thead>
        <tbody class="divide-y divide-
gray-200 bg-white">
         @foreach ($importedData as
$data)
              @foreach ($data->toArray()
as $key => $value)
               <td class="px-4 py-2 text-
sm text-gray-700 whitespace-nowrap">
                 {{ $value }}
               @endforeach
            @endforeach
        <div class="mt-4">
        {{ $importedData->links() }}
     </div>
    @endif
```

```
</div>
</main>
@endsection
```

apakah kamu bisa memperbaikinya agar keterangan memiliki background berwarna kuning?

- ✓ Sudah diperbarui sesuai preferensimu! Sekarang:
- Struktur tetap ringkas dan natural seperti sebelumnya.
- Kolom keterangan ditandai dengan background kuning (bg-yellow-100) dan teks ditebalkan agar tampak sebagai Label Target.

Kalau kamu mau tambahkan fitur "highlight baris berdasarkan label" atau "export data ini ke Excel/CSV", tinggal bilang aja!

oh iya aku baru ingat, masih ingat bagian resources/views/pages/naive-bayes/dataset.blade.php.
menurut mu bagaiman cara agar notifikasi file apa saja yang sudah di upload tidak terhapus ketika user menganti halaman, dan semisal user mengunjungi halaman lain, lalu mengakses resources/views/pages/naive-bayes/dataset.blade.php maka notifikasi tentang informasi file apa saja yang di upload masih ada. menurut bagaiamna cara best practice untuk melakukan hal tersebu t?

Pertanyaan kamu bagus banget — ini menyentuh persistensi notifikasi lintas halaman yang tidak bergantung hanya pada sesi satu request. Karena session()->flash() (seperti session('uploaded_files')) hanya bertahan 1 request saja, maka dia hilang begitu user pindah halaman.

☑ Best Practice: Simpan notifikasi upload di database per user

Supaya info file yang berhasil diupload tetap muncul walau halaman direfresh atau user kembali ke halaman itu lain waktu, kamu bisa simpan data notifikasi di tabel khusus, misalnya upload_logs, yang isinya bisa diload berdasarkan user login.

🔧 Langkah implementasinya:

1. Buat Migration:

```
bash

php artisan make:migration create_upload_logs_table
```

Isi migrasinya:

```
Schema::create('upload_logs', function (Blueprint $table) {
    $table->id();
    $table->string('filename');
    $table->string('status')->default('success'); // atau failed
    $table->string('user_email'); // atau user_id kalau ada auth
    $table->text('error_message')->nullable();
    $table->timestamps();
});
```

2. Simpan data upload saat proses import:

Di controller ketika upload berhasil:

```
php

UploadLog::create([
   'filename' => $file->getClientOriginalName(),
   'status' => 'success',
   'user_email' => auth()->user()->email,
]);
```

Atau saat gagal:

```
php

UploadLog::create([
   'filename' => $file->getClientOriginalName(),
   'status' => 'failed',
   'error_message' => $e->getMessage(),
   'user_email' => auth()->user()->email,
]);
```

3. Ambil notifikasi di controller dataset:

```
suploadLogs = UploadLog::where('user_email', auth()->user()->email)
    ->orderByDesc('created_at')
    ->take(10)
    ->get();
```

Lalu kirim ke Blade:

```
php
```

4. Tampilkan di Blade:

```
blade
@if ($uploadLogs->isNotEmpty())
  <div class="mb-4 p-4 bg-blue-50 border border-blue-200 rounded-lg m
    <h3 class="font-medium text-blue-800 mb-2">Riwayat Upload File:</h
    ul class="list-disc list-inside space-y-1">
      @foreach ($uploadLogs as $log)
        status === 'failed' ? 'text-red-600' : 'text-blue-7
           {{ $log->filename }}
           @if ($log->status === 'failed')
             - {{ $log->error_message }}
           @endif
        @endforeach
    </div>
@endif
```

Keuntungan:

- User bisa selalu lihat file yang pernah diupload.
- Riwayat bisa dipantau admin juga.
- Bisa dibersihkan periodik (misalnya auto delete setelah 30 hari).
- Kalau mau, bisa tampilkan hanya hari ini, atau bulan ini.

Kalau kamu ingin solusi lebih ringan dan cepat tanpa database, bisa juga:

- Simpan ke session secara manual dan persist (session()->put())
 lalu forget hanya ketika dihapus manual.
- Atau simpan ke localStorage via JavaScript (tapi ini cuma di sisi klien, bisa hilang kalau cache dibersihkan).

Mau aku bantu implementasikan model, migration, dan bagian Blade-nya juga?

bisa tampilkan kode secara keseluruhan, dan dibagian manakah aku harus menempatkan kode tersebut Oke! Berikut ini implementasi lengkap persistensi notifikasi upload lintas halaman menggunakan database upload_logs .

1. Migration (buat file upload_logs)

Jalankan:

```
php artisan make:migration create_upload_logs_table
```

Isi migration:

```
Schema::create('upload_logs', function (Blueprint $table) {
    $table->id();
    $table->string('filename');
    $table->string('status')->default('success'); // success atau failed
    $table->string('user_email'); // atau user_id kalau pakai auth bawaan
    $table->text('error_message')->nullable();
    $table->timestamps();
});
```

Tempatkan ini di folder: database/migrations/

2. Model UploadLog

Buat model:

```
bash
php artisan make:model UploadLog
```

Isi model app/Models/UploadLog.php:

```
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class UploadLog extends Model
{
   protected $fillable = [
     'filename', 'status', 'user_email', 'error_message'
   ];
}
```

3. Simpan log di Controller saat proses upload

Letakkan kode ini di bagian upload dataset (misalnya NaiveBayesController@import):

```
php
use App\Models\UploadLog;
try {
  // proses penyimpanan/import data...
  UploadLog::create([
    'filename' => $file->getClientOriginalName(),
    'status' => 'success',
    'user_email' => auth()->user()->email,
  ]);
} catch (\Exception $e) {
  UploadLog::create([
    'filename' => $file->getClientOriginalName(),
     'status' => 'failed',
    'user_email' => auth()->user()->email,
    'error_message' => $e->getMessage(),
  ]);
}
```

4. Ambil log di controller dataset

Di NaiveBayesController@index atau controller untuk dataset.blade.php:

```
php

use App\Models\UploadLog;

public function index()
{
    $columns = [ /* kolom yang kamu perlukan untuk tabel dataset */ ];

$importedData = ImportedData::select($columns)->paginate(10);

$uploadLogs = UploadLog::where('user_email', auth()->user()->email)
    ->orderByDesc('created_at')
    ->limit(10)
    ->get();

return view('pages.naive-bayes.dataset', compact('importedData', 'uploa')
}
```

5. Tampilkan di dataset.blade.php

Ganti bagian ini:

```
@if (session('uploaded_files'))
...
@endif

@if (session('failed_files'))
...
@endif
```

Dengan:

```
blade
@if ($uploadLogs->isNotEmpty())
  <div class="mb-4 p-4 bg-blue-50 border border-blue-200 rounded-lg m
    <h3 class="font-medium text-blue-800 mb-2">Riwayat Upload File:
    ul class="list-disc list-inside space-y-1">
      @foreach ($uploadLogs as $log)
        status === 'failed' ? 'text-red-600' : 'text-blue-7
          {{ $log->filename }}
          @if ($log->status === 'failed')
             - {{ $log->error_message }}
          @endif
        @endforeach
    </div>
@endif
```

🔄 Hasilnya:

- Riwayat file yang diupload tetap tampil walau user refresh halaman, atau kembali ke halaman lain.
- Bisa dikembangkan untuk fitur arsip riwayat upload, atau admin log monitoring.

Kalau kamu butuh versi yang auto-hapus setelah sekian waktu, tinggal tambahkan scheduler artisan + soft delete.

Mau aku bantu setup seeder, soft delete, atau fitur hapus riwayat upload juga?

apakah ada cara lain? cara ini terlalu rumit, dan sepertinya prosesnya tidak berjalan dengan semestinya

Yes, kalau solusi dengan database terasa terlalu kompleks, maka ada alternatif yang lebih ringan dan langsung jalan: yaitu menggunakan session()->put() alih-alih session()->flash().

🧠 Perbedaan:

- session()->flash() hanya bertahan 1 request.
- session()->put() akan bertahan sampai kamu hapus secara
 eksplisit, jadi cocok untuk menyimpan notifikasi lintas halaman.

Cara Alternatif (tanpa database)

```
1.  Ubah dari flash() ke put()
```

Di controller import (misalnya saat upload berhasil):

```
$uploadedFiles = session()->get('uploaded_files', []);
$uploadedFiles[] = $file->getClientOriginalName();
session()->put('uploaded_files', $uploadedFiles);
```

Untuk yang gagal:

```
sfailedFiles = session()->get('failed_files', []);
sfailedFiles[] = [
    'name' => $file->getClientOriginalName(),
    'error' => $e->getMessage(),
];
session()->put('failed_files', $failedFiles);
```

Ini akan menyimpan info file upload dalam session dan tidak hilang sampai kamu hapus manual.

2. 🔁 Tampilkan di Blade (tetap sama)

Di dataset.blade.php , ini tetap kamu bisa pakai:

```
<h3 class="text-sm font-medium text-yellow-800">Beberapa file gaga

    @foreach (session('failed_files') as $file)
    <strong>{{ $file['name'] }}</strong>: {{ $file['error'] }}
    @endforeach

</div>
@endif
```

3. X Tambahkan tombol untuk "hapus notifikasi"

Karena put() akan tersimpan, kamu bisa beri opsi di UI untuk menghapusnya:

```
class="text-sm text-red-600 underline hover:text-red-800"> blade

cform method="POST" action="{{ route('naive-bayes.dataset.clear-session
    @csrf
    <button type="submit"
    class="text-sm text-red-600 underline hover:text-red-800"> Bersihkan
    </form>
```

Di route:

```
php

Route::post('/naive-bayes/dataset/clear-session', [NaiveBayesController::cl
```

Di controller:

```
public function clearSession()
{
    session()->forget(['uploaded_files', 'failed_files']);
    return back();
}
```

◎ Keunggulan metode ini:

- Sangat simpel, tidak perlu buat model/migrasi.
- Notifikasi akan muncul sampai user menghapusnya atau logout.
- Bisa langsung jalan tanpa modifikasi besar.

Kalau kamu ingin, bisa juga digabung dengan penambahan timestamp agar tahu kapan file itu diupload.

Mau aku bantu generate semua kode yang dibutuhkan di Laravel?

```
belum bekerja, coba kamu cek
app/Http/Controllers/DatasetController.p
hp:
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\ImportedData;
use App\Models\ImportedDataHistory;
use Maatwebsite\Excel\Facades\Excel;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Str;
use App\Models\UploadLog;
class DatasetController extends Controller
  // Mapping kolom Excel ke kolom
database
  const COLUMN_MAPPING = [
    'RW' => 'rw',
    'R W' => 'rw',
    'KRITERIA' => 'kriteria',
    'Kriteria' => 'kriteria',
    'USIA' => 'usia',
    'Usia' => 'usia',
    'NAMA' => 'nama',
    'Nama' => 'nama',
    'JUMLAH TANGGUNGAN KEPALA
KELUARGA' =>
'tanggungan_kepala_keluarga',
    'Jumlah Tanggungan Kepala
Keluarga' =>
'tanggungan_kepala_keluarga',
    'Tanggungan' =>
'tanggungan_kepala_keluarga',
    'LANSIA' => 'lansia',
    'Lansia' => 'lansia',
    'ADA LANSIA' => 'lansia',
    'ANAK WAJIB SEKOLAH' =>
'anak_wajib_sekolah',
    'Anak Wajib Sekolah' =>
'anak_wajib_sekolah',
    'Anak Sekolah' =>
'anak_wajib_sekolah',
    'PENGHASILAN KEPALA KELUARGA' =>
'penghasilan_kepala_keluarga',
    'Penghasilan Kepala Keluarga' =>
'penghasilan_kepala_keluarga',
    'Penghasilan' =>
'penghasilan_kepala_keluarga',
    'STATUS BPJS ANGGOTA KELUARGA'
=> 'status_bpjs',
```

```
'Status BPJS Anggota Keluarga' =>
'status_bpjs',
    'BPJS' => 'status_bpjs',
    'TIPE KENDARAAN' =>
'tipe_kendaraan',
    'Tipe Kendaraan' => 'tipe_kendaraan',
    'Kendaraan' => 'tipe_kendaraan',
    'SUMBER AIR BERSIH' => 'sumber_air',
    'Sumber Air Bersih' => 'sumber_air',
    'Air Bersih' => 'sumber_air',
    'TIPE JAMBAN' => 'tipe_jamban',
    'Tipe Jamban' => 'tipe_jamban',
    'Jamban' => 'tipe_jamban',
    'STATUS KEPEMILIKAN BANGUNAN'
=> 'status_kepemilikan_bangunan',
    'Status Kepemilikan Bangunan' =>
'status_kepemilikan_bangunan',
    'Kepemilikan Bangunan' =>
'status_kepemilikan_bangunan',
    'BAHAN DASAR LANTAI' =>
'bahan_lantai',
    'Bahan Dasar Lantai' =>
'bahan_lantai',
    'Lantai' => 'bahan_lantai',
    'BAHAN DASAR DINDING' =>
'bahan_dinding',
    'Bahan Dasar Dinding' =>
'bahan_dinding',
    'Dinding' => 'bahan_dinding',
    'KATEGORI LUAS BANGUNAN' =>
'kategori_luas_bangunan',
    'Kategori Luas Bangunan' =>
'kategori_luas_bangunan',
    'Luas Bangunan' =>
'kategori_luas_bangunan',
    'KETERANGAN' => 'keterangan',
    'Keterangan' => 'keterangan',
    'Status' => 'keterangan'
  ];
  public function index()
  {
    $columns = [
      'rw',
      'kriteria',
      'usia',
      'tanggungan_kepala_keluarga',
      'lansia',
      'anak_wajib_sekolah',
      'penghasilan_kepala_keluarga',
      'status_bpjs',
      'tipe_kendaraan',
      'sumber_air',
      'tipe_jamban',
```

```
'status_kepemilikan_bangunan',
      'bahan_lantai',
      'bahan_dinding',
      'kategori_luas_bangunan',
      'keterangan'
    ];
    $importedData =
ImportedData::select($columns)-
>paginate(50);
    $uploadLogs =
UploadLog::where('user_email',
Auth::user()->email)
      ->orderByDesc('created_at')
      ->limit(10)
      ->get();
    return view('pages.naive-
bayes.dataset', compact('importedData',
'uploadLogs'));
  }
  public function import(Request
$request)
  {
    $request->validate([
      'files.*' =>
'required|mimes:xlsx,xls|max:2048'
    ]);
    try {
      $this->moveExistingDataToHistory();
      $importedCount = 0;
      $failedFiles = [];
      foreach ($request->file('files') as
$file) {
        try {
           $data = Excel::toArray([], $file)
[0];
           $headers = array_shift($data);
          // Normalisasi header - hapus
spasi berlebih dan ubah ke uppercase
           $normalizedHeaders =
array_map(function ($header) {
             return preg_replace('/\s+/', ' ',
trim(strtoupper($header)));
          }, $headers);
          // Validasi header minimal
           $requiredColumns = ['RW',
'USIA', 'KETERANGAN'];
```

```
$missingColumns =
array_diff($requiredColumns,
$normalizedHeaders);
          if (!empty($missingColumns)) {
            throw new \Exception("File
{$file->getClientOriginalName()} tidak
memiliki kolom wajib: " . implode(', ',
$missingColumns));
          }
          foreach ($data as $rowIndex =>
$row) {
            $mappedData = [];
            $errorMessages = [];
            foreach
($normalizedHeaders as $index =>
$header) {
               $header =
preg_replace('/\s+/', ' ', trim($header));
               if
(array_key_exists($header,
self::COLUMN_MAPPING)) {
                 $dbColumn =
self::COLUMN_MAPPING[$header];
                 $value = $row[$index] ??
null;
                 try {
$mappedData[$dbColumn] = $this-
>transformValue($dbColumn, $value);
                } catch (\Exception $e) {
                   $errorMessages[] =
"Kolom {$header} baris {$rowIndex}: ".
$e->getMessage();
              }
            }
            // Validasi data minimal
            if (empty($mappedData['rw'])
|| empty($mappedData['usia']) ||
empty($mappedData['keterangan'])) {
               $errorMessages[] = "Baris
{$rowIndex} tidak memiliki data RW, Usia,
atau Keterangan";
               continue;
            }
            if (!empty($errorMessages)) {
               Log::warning('Import
partial error in file '. $file-
```

```
>getClientOriginalName() . ': ' . implode(', ',
$errorMessages));
             }
             // Set kriteria berdasarkan
usia
             $mappedData['kriteria'] =
$this-
>determineKriteria($mappedData['usia']);
             $mappedData['user_email'] =
Auth::user()->email;
             $mappedData['user_id'] =
Auth::id(); // Tambahkan ini
ImportedData::create($mappedData);
             $importedCount++;
          }
           $uploadedFiles = session()-
>get('uploaded_files', []);
           $uploadedFiles[] = $file-
>getClientOriginalName();
           session()->put('uploaded_files',
$uploadedFiles);
        } catch (\Exception $e) {
           $failedFiles = session()-
>get('failed_files', []);
           $failedFiles[] = [
             'name' => $file-
>getClientOriginalName(),
             'error' => $e->getMessage(),
          ];
           session()->put('failed_files',
$failedFiles);
           $failedFiles[] = [
             'name' => $file-
>getClientOriginalName(),
             'error' => $e->getMessage()
           Log::error("Error processing file
{$file->getClientOriginalName()}: ".$e-
>getMessage());
        }
      }
      $message = "Berhasil mengimport
{$importedCount} data";
      if (!empty($failedFiles)) {
        $failedNames =
array_column($failedFiles, 'name');
        $message .= ". Gagal memproses
".count($failedFiles)." file: ".implode(',
```

```
', $failedNames);
        return redirect()
           ->route('naive-
bayes.dataset.index')
           ->with('warning', $message)
           ->with('failed_files',
$failedFiles);
      }
      return redirect()
        ->route('naive-
bayes.dataset.index')
        ->with('success', $message);
    } catch (\Exception $e) {
      Log::error('Import error: ' . $e-
>getMessage());
      return back()
        ->with('error', 'Terjadi kesalahan
saat mengimport data: ' . $e-
>getMessage());
    }
  }
  protected function
moveExistingDataToHistory()
  {
    $existingData = ImportedData::all();
    foreach ($existingData as $data) {
      ImportedDataHistory::create([
        'user_email' => $data-
>user_email,
        'dataset_id' => $data->id,
        'row_data' => $data->toArray(),
        'modified_at' => now()
      ]);
    }
    ImportedData::truncate();
  }
  protected function
transformValue($column, $value)
  {
    $value = trim($value);
    if ($value === null || $value === '') {
      return null;
    }
    // Transformasi untuk kolom enum
    switch ($column) {
      case 'lansia':
      case 'anak_wajib_sekolah':
```

```
$value = strtolower($value);
        return (str_contains($value,
'tidak') || $value === 'no' ? 'tidak_ada' :
'ada');
      case 'keterangan':
        $value = strtolower($value);
        return (str_contains($value,
'tidak') || $value === 'no' ?
'bukan_penerima' : 'penerima');
      case
'penghasilan_kepala_keluarga':
        $value = strtolower($value);
        if (str_contains($value, 'rendah'))
return 'Rendah';
        if (str_contains($value, 'sedang'))
return 'Sedang';
        if (str_contains($value, 'tinggi'))
return 'Tinggi';
        return $value;
      case
'tanggungan_kepala_keluarga':
        $value = strtolower($value);
        if (str_contains($value, 'sedikit'))
return 'Sedikit';
        if (str_contains($value, 'sedang'))
return 'Sedang';
        if (str_contains($value, 'banyak'))
return 'Banyak';
        return $value;
      case 'kategori_luas_bangunan':
        $value = strtolower($value);
        if (str_contains($value, 'kecil'))
return 'Kecil';
        if (str_contains($value, 'sedang'))
return 'Sedang';
        if (str_contains($value, 'besar'))
return 'Besar';
        return $value;
      case 'usia':
        return is_numeric($value) ?
(int)$value: null;
      default:
        return $value;
    }
  }
  protected function
determineKriteria($usia)
  {
```

```
if ($usia > 55) return 'lansia';
    if ($usia <= 18) return 'anak_sekolah';</pre>
    return 'usia_produktif';
  }
  protected function
mapPenghasilan($value)
  {
    $value = strtolower($value);
    if (Str::contains($value, 'rendah'))
return 'Rendah';
    if (Str::contains($value, 'sedang'))
return 'Sedang';
    if (Str::contains($value, 'tinggi')) return
'Tinggi';
    return $value;
  }
  protected function
mapTanggungan($value)
  {
    $value = strtolower($value);
    if (Str::contains($value, 'sedikit'))
return 'Sedikit';
    if (Str::contains($value, 'sedang'))
return 'Sedang';
    if (Str::contains($value, 'banyak'))
return 'Banyak';
    return $value;
  }
  protected function
mapLuasBangunan($value)
  {
    $value = strtolower($value);
    if (Str::contains($value, 'kecil')) return
'Kecil';
    if (Str::contains($value, 'sedang'))
return 'Sedang';
    if (Str::contains($value, 'besar')) return
'Besar';
    return $value;
  }
}
```

```
Initial Process View

@extends('layout.app')

@section('title', 'Initial Process')

@section('content')

header class="bg-white shadow">

div class="max-w-7xl mx-auto py-6 px-4 sm:px-6 lg:px-8">
```

Terima kasih, aku sudah cek kode lengkap DatasetController.php kamu. Ternyata penyebab kenapa notifikasi file yang sudah diupload hilang setelah refresh atau pindah halaman adalah karena kamu masih menggunakan session()->with() di redirect alih-alih menampilkan notifikasi berdasarkan session()->get() biasa di controller index.

Solusi Paling Mudah: Pindahkan logika ke controller index()

Saat ini kamu hanya melakukan ini di import():

```
php

session()->put('uploaded_files', $uploadedFiles); // 🗹 Sudah benar
```

Tapi di index() kamu tidak mengambil session('uploaded_files'), melainkan:

```
php
return view('pages.naive-bayes.dataset', compact('importedData', 'upload
```

🔧 Jadi yang perlu kamu lakukan:

1. Tambahkan ke controller index():

```
$uploadedFiles = session()->get('uploaded_files', []);
$failedFiles = session()->get('failed_files', []);
```

Lalu ubah return menjadi:

```
php
return view('pages.naive-bayes.dataset', compact('importedData', 'upload
```

2. Di Blade (dataset.blade.php) ubah:

Dari:

```
blade
@if (session('uploaded_files'))
```

Menjadi:

```
blade
@if (!empty($uploadedFiles))
  <div class="mb-4 p-4 bg-blue-50 border border-blue-200 rounded-lg m
   <h3 class="font-medium text-blue-800 mb-2">File yang berhasil di ur
   ul class="list-disc list-inside space-y-1">
      @foreach ($uploadedFiles as $file)
        {{ $file }}
     @endforeach
    </div>
@endif
@if (!empty($failedFiles))
  <div class="bg-yellow-50 border-l-4 border-yellow-400 p-4 mb-4">
    <h3 class="text-sm font-medium text-yellow-800">Beberapa file gaga
    @foreach ($failedFiles as $file)
        <strong>{{ $file['name'] }}</strong>: {{ $file['error'] }}
     @endforeach
    </div>
@endif
```

Opsional: Hapus riwayat saat diinginkan

Tambahkan tombol di Blade:

Tambahkan route:

```
php

Route::post('/naive-bayes/dataset/clear-session', [DatasetController::class,
```

Tambahkan method:

```
php

public function clearSession()
{
```

```
session()->forget(['uploaded_files', 'failed_files']);
return back();
}
```

Kesimpulan:

Kamu sudah benar menyimpan notifikasi dengan session()->put(), hanya perlu ambil datanya di controller index(), bukan hanya di with() redirect. Dengan begitu, notifikasi akan terus tampil sampai dihapus manual.

Kalau kamu mau aku bantu kirim potongan controller dan blade versi lengkap setelah revisi ini, tinggal bilang aja.

```
ok, koreksi chat
app/Http/Controllers/DatasetController.p
hp:
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Models\ImportedData;
use App\Models\ImportedDataHistory;
use Maatwebsite\Excel\Facades\Excel;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Log;
use Illuminate\Support\Str;
class DatasetController extends Controller
  // Mapping kolom Excel ke kolom
database
  const COLUMN_MAPPING = [
    'RW' => 'rw',
    'R W' => 'rw',
    'KRITERIA' => 'kriteria',
    'Kriteria' => 'kriteria',
    'USIA' => 'usia',
    'Usia' => 'usia',
    'NAMA' => 'nama',
    'Nama' => 'nama',
    'JUMLAH TANGGUNGAN KEPALA
KELUARGA' =>
'tanggungan_kepala_keluarga',
    'Jumlah Tanggungan Kepala
Keluarga' =>
'tanggungan_kepala_keluarga',
    'Tanggungan' =>
'tanggungan_kepala_keluarga',
```

```
'LANSIA' => 'lansia',
    'Lansia' => 'lansia',
    'ADA LANSIA' => 'lansia',
    'ANAK WAJIB SEKOLAH' =>
'anak_wajib_sekolah',
    'Anak Wajib Sekolah' =>
'anak_wajib_sekolah',
    'Anak Sekolah' =>
'anak_wajib_sekolah',
    'PENGHASILAN KEPALA KELUARGA' =>
'penghasilan_kepala_keluarga',
    'Penghasilan Kepala Keluarga' =>
'penghasilan_kepala_keluarga',
    'Penghasilan' =>
'penghasilan_kepala_keluarga',
    'STATUS BPJS ANGGOTA KELUARGA'
=> 'status_bpjs',
    'Status BPJS Anggota Keluarga' =>
'status_bpjs',
    'BPJS' => 'status_bpjs',
    'TIPE KENDARAAN' =>
'tipe_kendaraan',
    'Tipe Kendaraan' => 'tipe_kendaraan',
    'Kendaraan' => 'tipe_kendaraan',
    'SUMBER AIR BERSIH' => 'sumber_air',
    'Sumber Air Bersih' => 'sumber_air',
    'Air Bersih' => 'sumber air',
    'TIPE JAMBAN' => 'tipe_jamban',
    'Tipe Jamban' => 'tipe_jamban',
    'Jamban' => 'tipe_jamban',
    'STATUS KEPEMILIKAN BANGUNAN'
=> 'status_kepemilikan_bangunan',
    'Status Kepemilikan Bangunan' =>
'status_kepemilikan_bangunan',
    'Kepemilikan Bangunan' =>
'status_kepemilikan_bangunan',
    'BAHAN DASAR LANTAI' =>
'bahan_lantai',
    'Bahan Dasar Lantai' =>
'bahan_lantai',
    'Lantai' => 'bahan lantai',
    'BAHAN DASAR DINDING' =>
'bahan_dinding',
    'Bahan Dasar Dinding' =>
'bahan_dinding',
    'Dinding' => 'bahan_dinding',
    'KATEGORI LUAS BANGUNAN' =>
'kategori_luas_bangunan',
    'Kategori Luas Bangunan' =>
'kategori_luas_bangunan',
    'Luas Bangunan' =>
'kategori_luas_bangunan',
    'KETERANGAN' => 'keterangan',
    'Keterangan' => 'keterangan',
    'Status' => 'keterangan'
```

```
];
  public function index()
    $columns = [
      'rw',
      'kriteria',
      'usia',
      'nama',
      'tanggungan_kepala_keluarga',
      'lansia',
      'anak_wajib_sekolah',
      'penghasilan_kepala_keluarga',
      'status_bpjs',
      'tipe_kendaraan',
      'sumber_air',
      'tipe_jamban',
      'status_kepemilikan_bangunan',
      'bahan_lantai',
      'bahan_dinding',
      'kategori_luas_bangunan',
      'keterangan'
    ];
    $importedData =
ImportedData::select($columns)-
>paginate(50);
    $uploadedFiles = session()-
>get('uploaded_files', []);
    $failedFiles = session()-
>get('failed_files', []);
    return view('pages.naive-
bayes.dataset', compact('importedData',
'uploadedFiles', 'failedFiles'));
  public function import(Request
$request)
  {
    $request->validate([
      'files.*' =>
'required|mimes:xlsx,xls|max:2048'
    ]);
    try {
      $this->moveExistingDataToHistory();
      $importedCount = 0;
      $failedFiles = [];
      foreach ($request->file('files') as
$file) {
```

```
try {
          $data = Excel::toArray([], $file)
[0];
          $headers = array_shift($data);
          // Normalisasi header - hapus
spasi berlebih dan ubah ke uppercase
          $normalizedHeaders =
array_map(function ($header) {
            return preg_replace('/\s+/', ' ',
trim(strtoupper($header)));
          }, $headers);
          // Validasi header minimal
          $requiredColumns = ['RW',
'USIA', 'KETERANGAN'];
          $missingColumns =
array_diff($requiredColumns,
$normalizedHeaders);
          if (!empty($missingColumns)) {
            throw new \Exception("File
{$file->getClientOriginalName()} tidak
memiliki kolom wajib: " . implode(', ',
$missingColumns));
          foreach ($data as $rowIndex =>
$row) {
            $mappedData = [];
            $errorMessages = [];
            foreach
($normalizedHeaders as $index =>
$header) {
               $header =
preg_replace('/\s+/', ' ', trim($header));
               if
(array_key_exists($header,
self::COLUMN MAPPING)) {
                 $dbColumn =
self::COLUMN_MAPPING[$header];
                 $value = $row[$index] ??
null;
                 try {
$mappedData[$dbColumn] = $this-
>transformValue($dbColumn, $value);
                } catch (\Exception $e) {
                   $errorMessages[] =
"Kolom {$header} baris {$rowIndex}: " .
$e->getMessage();
```

}

```
}
             // Validasi data minimal
             if (empty($mappedData['rw'])
|| empty($mappedData['usia']) ||
empty($mappedData['keterangan'])) {
               $errorMessages[] = "Baris
{$rowIndex} tidak memiliki data RW, Usia,
atau Keterangan";
               continue;
             }
             if (!empty($errorMessages)) {
               Log::warning('Import
partial error in file '. $file-
>getClientOriginalName() . ': ' . implode(', ',
$errorMessages));
             }
             // Set kriteria berdasarkan
usia
             $mappedData['kriteria'] =
$this-
>determineKriteria($mappedData['usia']);
             $mappedData['user_email'] =
Auth::user()->email;
             $mappedData['user_id'] =
Auth::id(); // Tambahkan ini
ImportedData::create($mappedData);
             $importedCount++;
           $uploadedFiles = session()-
>get('uploaded_files', []);
           $uploadedFiles[] = $file-
>getClientOriginalName();
           session()->put('uploaded_files',
$uploadedFiles);
        } catch (\Exception $e) {
           $failedFiles = session()-
>get('failed_files', []);
           $failedFiles[] = [
             'name' => $file-
>getClientOriginalName(),
             'error' => $e->getMessage(),
          session()->put('failed_files',
$failedFiles);
        }
      }
      $message = "Berhasil mengimport
```

```
{$importedCount} data";
      if (!empty($failedFiles)) {
        $failedNames =
array_column($failedFiles, 'name');
        $message .= ". Gagal memproses
".count($failedFiles)." file: ".implode(',
', $failedNames);
        return redirect()
           ->route('naive-
bayes.dataset.index')
           ->with('warning', $message)
           ->with('failed_files',
$failedFiles);
      }
      return redirect()
        ->route('naive-
bayes.dataset.index')
        ->with('success', $message);
    } catch (\Exception $e) {
      Log::error('Import error: ' . $e-
>getMessage());
      return back()
        ->with('error', 'Terjadi kesalahan
saat mengimport data: '. $e-
>getMessage());
    }
  }
  protected function
moveExistingDataToHistory()
    $existingData = ImportedData::all();
    foreach ($existingData as $data) {
      ImportedDataHistory::create([
        'user_email' => $data-
>user_email,
        'dataset_id' => $data->id,
        'row_data' => $data->toArray(),
        'modified_at' => now()
      ]);
    }
    ImportedData::truncate();
  }
  protected function
transformValue($column, $value)
  {
    $value = trim($value);
    if ($value === null || $value === '') {
```

```
return null;
    }
    // Transformasi untuk kolom enum
    switch ($column) {
      case 'lansia':
      case 'anak_wajib_sekolah':
        $value = strtolower($value);
        return (str_contains($value,
'tidak') || $value === 'no' ? 'tidak_ada' :
'ada');
      case 'keterangan':
        $value = strtolower($value);
        return (str_contains($value,
'tidak') || $value === 'no' ?
'bukan_penerima': 'penerima');
      case
'penghasilan_kepala_keluarga':
        $value = strtolower($value);
        if (str_contains($value, 'rendah'))
return 'Rendah';
        if (str_contains($value, 'sedang'))
return 'Sedang';
        if (str_contains($value, 'tinggi'))
return 'Tinggi';
        return $value;
      case
'tanggungan_kepala_keluarga':
        $value = strtolower($value);
        if (str_contains($value, 'sedikit'))
return 'Sedikit';
        if (str_contains($value, 'sedang'))
return 'Sedang';
        if (str_contains($value, 'banyak'))
return 'Banyak';
        return $value;
      case 'kategori_luas_bangunan':
        $value = strtolower($value);
        if (str_contains($value, 'kecil'))
return 'Kecil';
        if (str_contains($value, 'sedang'))
return 'Sedang';
        if (str_contains($value, 'besar'))
return 'Besar';
        return $value;
      case 'usia':
        return is_numeric($value)?
(int)$value : null;
      default:
```

```
return $value;
}

protected function
determineKriteria($usia)
{
   if ($usia > 55) return 'lansia';
   if ($usia <= 18) return 'anak_sekolah';
   return 'usia_produktif';
}

public function clearSession()
{
   session()->forget(['uploaded_files',
'failed_files']);
   return back();
}
```

(i) You've reached our limits of messages. Please try again later.