import React, { useEffect, useState, useRef } from "react";

import { useParams, Link } from "react-router-dom";

import axios from "axios";

import { Audio } from "react-loader-spinner";

import { Chart as ChartJS, CategoryScale, LinearScale, PointElement, LineElement, ArcElement, Tooltip, Legend, Title, DoughnutController } from "chart.js";

import { useNavigate, useLocation } from "react-router-dom";

import ChartDataLabels from "chartjs-plugin-datalabels";

import Breadcrumb from "./Breadcrumb";

import DoughnutChart from "./Chart/DoughnutChart";

import LineChart from "./Chart/LineChart"; // Import komponen LineChart

ChartJS.register(CategoryScale, LinearScale, PointElement, LineElement, ArcElement, Tooltip, Legend, Title, DoughnutController, ChartDataLabels);

const formatDate = (dateString) => {

  const months = ["Januari", "Februari", "Maret", "April", "Mei", "Juni", "Juli", "Agustus", "September", "Oktober", "November", "Desember"];

  const date = new Date(dateString);

  const day = date.getDate();

  const month = months[date.getMonth()];

  const year = date.getFullYear();

  return `${day} ${month} ${year}`;

};

const generateTimelineLabels = (startDate, endDate, points = 5) => {

  if (!startDate || !endDate || isNaN(new Date(startDate)) || isNaN(new Date(endDate))) {

    console.warn("⚠️ Skip generating timeline labels, invalid start or end date:", startDate, endDate);

    return [];

  }

  const start = new Date(startDate);

  const end = new Date(endDate);

  const timeline = [];

  for (let i = 0; i < points; i++) {

    const current = new Date(start.getTime() + ((end - start) / (points - 1)) \* i);

    const formattedDate = current.toISOString().split("T")[0]; // Format YYYY-MM-DD

    timeline.push(formattedDate);

  }

  return timeline;

};

const countDesaPerTimeline = (timeline, desaList, kategori) => {

  return timeline.map((date) => {

    const count = desaList.filter((desa) => {

      const desaStart = desa.tahun\_pembentukan ? new Date(desa.tahun\_pembentukan) : null;

      const timelineDate = new Date(date);

      if (!desaStart || isNaN(desaStart)) {

        console.warn(`🚨 Tahun pembentukan tidak valid untuk desa ID ${desa.id}:`, desa);

        return false;

      }

      return desa.kategori === kategori && desaStart <= timelineDate;

    }).length;

    return count;

  });

};

const mapKabupatenName = (namaKabupaten) => {

  const kabupatenMapping = {

    "Kulon Progo": "KAB. KULON PROGO",

    Sleman: "KAB. SLEMAN",

    Bantul: "KAB. BANTUL",

    "Kota Yogyakarta": "KOTA YOGYAKARTA",

    Gunungkidul: "KAB. GUNUNGKIDUL",

  };

  return kabupatenMapping[namaKabupaten] || namaKabupaten;

};

const mapKabupatenKategori = (namaKabupaten) => {

  const kabupatenMapping = {

    "Kulon Progo": "kulonprogo",

    Sleman: "sleman",

    Bantul: "bantul",

    "Kota Yogyakarta": "yogyakarta",

    Gunungkidul: "gunungkidul",

  };

  return kabupatenMapping[namaKabupaten] || namaKabupaten;

};

const KabupatenDetail = () => {

  const lineChartRef = useRef(null);

  const doughnutChartRef = useRef(null);

  const { id } = useParams();

  const [kabupaten, setKabupaten] = useState(null);

  const [loading, setLoading] = useState(true);

  const [error, setError] = useState("");

  const [desaList, setDesaList] = useState([]);

  const [timelineLabels, setTimelineLabels] = useState([]);

  useEffect(() => {

    const fetchKabupatenDetail = async () => {

      try {

        const kabupatenResponse = await axios.get(`http://localhost:5000/api/kabupaten/${id}`);

        const kabupatenData = kabupatenResponse.data;

        console.log("Kabupaten Data:", kabupatenData);

        // Cek nilai kabupatenData.nama\_kabupaten

        console.log("Nama Kabupaten:", kabupatenData.nama\_kabupaten);

        // Map nama kabupaten ke format yang diinginkan

        const mappedNamaKabupaten = mapKabupatenName(kabupatenData.nama\_kabupaten);

        console.log("Mapped Nama Kabupaten:", mappedNamaKabupaten);

        // Ambil data desa berdasarkan kabupaten

        // Ambil data desa berdasarkan kabupaten

        const desaResponse = await axios.get(`http://localhost:5000/api/desa?kabupaten=${mappedNamaKabupaten}`);

        const desaData = desaResponse.data;

        console.log("Desa Data:", desaData); // Cek data desa yang dikembalikan oleh API

        setDesaList(desaData);

        const validDates = desaData.map((desa) => (desa.tahun\_pembentukan ? new Date(desa.tahun\_pembentukan) : null)).filter((date) => date !== null && !isNaN(date.getTime()));

        let startDate, endDate;

        if (validDates.length > 0) {

          startDate = new Date(Math.min(...validDates.map((date) => date.getTime())));

          endDate = new Date(Math.max(...validDates.map((date) => date.getTime())));

        } else {

          // Jika tidak ada data tahun\_pembentukan yang valid, gunakan periode\_awal dan periode\_akhir dari kabupaten

          startDate = new Date(kabupatenData.periode\_awal);

          endDate = new Date(kabupatenData.periode\_akhir);

        }

        const labels = generateTimelineLabels(startDate, endDate);

        setTimelineLabels(labels);

        // Generate timeline labels

        const categories = ["maju", "berkembang", "tumbuh"];

        const countData = {};

        for (const category of categories) {

          try {

            const mappedNamaKabupatenKategori = mapKabupatenKategori(kabupatenData.nama\_kabupaten);

            console.log("Mapped Nama Kabupaten:", mappedNamaKabupatenKategori);

            const response = await axios.get(`http://localhost:5000/api/desa/count/${mappedNamaKabupatenKategori}/${category}`);

            countData[category] = response.data.count || 0;

          } catch (err) {

            console.error(`Gagal memuat data untuk kategori ${category}`, err);

            countData[category] = 0;

          }

        }

        setKabupaten({

          ...kabupatenData,

          jumlah\_maju: countData.maju,

          jumlah\_berkembang: countData.berkembang,

          jumlah\_tumbuh: countData.tumbuh,

        });

        setLoading(false);

      } catch (err) {

        console.error("Gagal memuat detail kabupaten", err);

        setLoading(false);

      }

    };

    fetchKabupatenDetail();

  }, [id]);

  if (loading) {

    return (

      <div className="flex items-center justify-center h-screen">

        <Audio type="Bars" color="#3FA2F6" height={80} width={80} />

      </div>

    );

  }

  if (error) {

    return <div className="text-center text-xl text-red-500">{error}</div>;

  }

  const desaMaju = kabupaten.jumlah\_maju;

  const desaBerkembang = kabupaten.jumlah\_berkembang;

  const desaTumbuh = kabupaten.jumlah\_tumbuh;

  const totalDesa = kabupaten.jumlah\_desa;

  const totalJumlahDesa = kabupaten.jumlah\_maju + kabupaten.jumlah\_berkembang + kabupaten.jumlah\_tumbuh;

  const doughnutChartData = {

    desaMaju,

    desaBerkembang,

    desaTumbuh,

    totalDesa,

    totalJumlahDesa,

  };

  const lineChartData = {

    labels: timelineLabels,

    datasets: [

      {

        label: "maju",

        data: countDesaPerTimeline(timelineLabels, desaList, "Maju"),

        borderColor: "#4CAF50",

        backgroundColor: "rgba(76, 175, 80, 0.2)",

        fill: false,

      },

      {

        label: "berkembang",

        data: countDesaPerTimeline(timelineLabels, desaList, "Berkembang"),

        borderColor: "#FFC107",

        backgroundColor: "rgba(255, 193, 7, 0.2)",

        fill: false,

      },

      {

        label: "tumbuh",

        data: countDesaPerTimeline(timelineLabels, desaList, "Tumbuh"),

        borderColor: "#FF5722",

        backgroundColor: "rgba(255, 87, 34, 0.2)",

        fill: false,

      },

      {

        label: "total kelompok",

        data: timelineLabels.map(

          (date) =>

            countDesaPerTimeline(timelineLabels, desaList, "Maju")[timelineLabels.indexOf(date)] +

            countDesaPerTimeline(timelineLabels, desaList, "Berkembang")[timelineLabels.indexOf(date)] +

            countDesaPerTimeline(timelineLabels, desaList, "Tumbuh")[timelineLabels.indexOf(date)]

        ),

        borderColor: "#3F51B5",

        backgroundColor: "rgba(63, 81, 181, 0.2)",

        fill: false,

      },

      {

        label: "total desa",

        data: timelineLabels.map(() => totalDesa),

        borderColor: "#999999",

        backgroundColor: "rgba(153, 153, 153, 0.2)",

        fill: false,

      },

    ],

  };

  return (

    <>

      <div className="p-5">

        {/\* <Breadcrumb items={breadcrumbPaths} /> \*/}

        <div className="bg-white p-4 rounded-md">

          <div className="px-2 flex justify-between items-center mb-4">

            <h1 className="text-lg lg:text-2xl font-medium">{kabupaten.nama\_kabupaten === "Kota Yogyakarta" ? kabupaten.nama\_kabupaten : `Kabupaten ${kabupaten.nama\_kabupaten}`}</h1>

            <Link to={`/kelompok-desa?kabupaten=${kabupaten.nama\_kabupaten === "Kota Yogyakarta" ? kabupaten.nama\_kabupaten : `KAB. ${kabupaten.nama\_kabupaten.toUpperCase()}`}`}>

              <button className="bg-blue-700 text-white text-xs lg:text-lg py-2 px-4 rounded-md shadow-md">Daftar Kelompok</button>

            </Link>

          </div>

          <div className="grid grid-cols-1 lg:grid-cols-3 gap-6">

            <div className="lg:col-span-2 grid grid-cols-1 sm:grid-cols-2 gap-4">

              <div className="bg-blue-500 text-white p-4 rounded-md shadow-md">

                <h2 className="text-sm lg:text-lg font-bold mb-2">Ketua Forum</h2>

                <p className="text-sm lg:text-lg">{kabupaten.ketua\_forum}</p>

              </div>

              <div className="bg-blue-500 text-white p-4 rounded-md shadow-md">

                <h2 className="text-sm lg:text-lg font-bold mb-2">Jumlah Desa</h2>

                <p className="text-sm lg:text-lg">{kabupaten.jumlah\_desa}</p>

              </div>

              <div className="bg-blue-500 text-white p-4 rounded-md shadow-md">

                <h2 className="text-sm lg:text-lg font-bold mb-2">Periode Pembentukan</h2>

                <p className="text-sm lg:text-lg">

                  {formatDate(kabupaten.periode\_awal)} — {formatDate(kabupaten.periode\_akhir)}

                </p>

              </div>

              <div className="bg-blue-500 text-white p-4 rounded-md shadow-md">

                <h2 className="text-sm lg:text-lg font-bold mb-2">Kelompok Desa Prima</h2>

                <p className="text-sm lg:text-lg">{kabupaten.jumlah\_maju + kabupaten.jumlah\_berkembang + kabupaten.jumlah\_tumbuh}</p>

              </div>

            </div>

            <div className="bg-white p-6 rounded-md shadow-md">

              <h2 className="text-sm lg:text-lg font-bold mb-4 text-center">Jumlah Kelompok Berdasarkan Kategori</h2>

              <div className="flex flex-col items-center space-y-4 px-6">

                <div className="flex justify-between w-full sm:w-40">

                  <span className="text-sm lg:text-lg text-gray-600">Maju</span>

                  <span className="text-sm lg:text-lg bg-green-200 text-green-800 px-4 py-1 rounded-md">{kabupaten.jumlah\_maju || 0}</span>

                </div>

                <div className="flex justify-between w-full sm:w-40">

                  <span className="text-sm lg:text-lg text-gray-600">Berkembang</span>

                  <span className="text-sm lg:text-lg bg-blue-200 text-blue-800 px-4 py-1 rounded-md">{kabupaten.jumlah\_berkembang || 0}</span>

                </div>

                <div className="flex justify-between w-full sm:w-40">

                  <span className="text-sm lg:text-lg text-gray-600">Tumbuh</span>

                  <span className="text-sm lg:text-lg bg-orange-200 text-orange-800 px-4 py-1 rounded-md">{kabupaten.jumlah\_tumbuh || 0}</span>

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

      <div className="px-5 pb-5 grid grid-cols-1 lg:grid-cols-3 gap-5">

        {/\* Line Chart \*/}

        <div ref={lineChartRef} className="lg:col-span-2 bg-white shadow-md p-6 rounded-md">

          <LineChart data={lineChartData} ref={lineChartRef} />

        </div>

        {/\* Doughnut Chart \*/}

        <div ref={doughnutChartRef} className="bg-white shadow-md p-6 rounded-md">

          <DoughnutChart data={doughnutChartData} ref={doughnutChartRef} />

        </div>

      </div>

    </>

  );

};

export default KabupatenDetail;