**Visualisasi Data Calon Pemilih dalam Peta Kota Malang**

***Disusun untuk memenuhi tugas akhir mata kuliah Pemrosesan dan Infrastruktur Data***



**Oleh:**

**1.** **John Nicholas F Lumbanbatu (225150307111054)**

**2.** **Hafidz Ghaza Al Ghozali (225150307111053)**

**3.**

**PROGRAM STUDI S1 TEKNIK KOMPUTER**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2024**

**1. Pendahuluan**

**1.1 Latar Belakang**

Pemilihan umum membutuhkan analisis data pemilih untuk memahami distribusi dukungan politik. Proyek ini bertujuan untuk memvisualisasikan data pemilih di Kota Malang dalam bentuk peta interaktif, memanfaatkan Hadoop untuk pengolahan data besar.

**1.2 Lingkup Pekerjaan**

**Cakupan pekerjaan:**

* Mengolah data pemilih menggunakan Hadoop MapReduce.
* Membuat peta interaktif dengan Folium di Python.

Output berupa file HTML.**2. Perancangan Sistem**

**2.1 Arsitektur Sistem**

**Sistem terdiri dari:**

* Hadoop Cluster: Untuk pengolahan data besar.
* Python: Untuk scripting pengolahan hasil dan visualisasi data.
* Folium: Untuk membuat peta interaktif.

**2.2 Pemilihan Teknologi**

**Teknologi yang digunakan:**

* Hadoop MapReduce: Mengolah data pemilih.
* Python Pandas: Mengonversi hasil MapReduce ke format yang dapat diolah.
* Folium: Membuat peta berbasis data geografis

**3. Implementasi**

**3.1 Langkah-Langkah Implementasi**

1. **Setup Hadoop Cluster:**
   * Instal Hadoop di server atau gunakan cluster lokal.
   * Siapkan file input data pemilih dalam format CSV.
2. **Implementasi MapReduce:**
   * Map: Membaca data pemilih berdasarkan kelurahan.
   * Reduce: Menghitung jumlah pendukung di setiap kelurahan.
3. **Pengolahan Hasil MapReduce:**
   * Ekspor hasil ke file CSV menggunakan Python.
   * Gunakan Pandas untuk membaca data.
4. **Visualisasi Data dengan Folium:**
   * Gunakan koordinat kelurahan untuk membuat marker.
   * Tampilkan data jumlah pendukung sebagai popup.

**3.2 Studi Kasus: Pengolahan Data Pemilih**

* Input: File CSV berisi data pemilih dengan kolom Kelurahan dan Jumlah Dukungan.
* Output: File HTML peta interaktif.

**3.3 Kode yang Digunakan**

**1. MapReduce Job:**

| 1  2  3  4  5  6  7  8  9  10  11  12  13 | from mrjob.job import MRJob  class DukunganPemilih(MRJob):  def mapper(self, \_, line):  data = line.split(',')  kelurahan, jumlah = data[0], int(data[1])  yield kelurahan, jumlah  def reducer(self, kelurahan, jumlah):  yield kelurahan, sum(jumlah)  if \_\_name\_\_ == '\_\_main\_\_':  DukunganPemilih.run() |
| --- | --- |

**2. Visualisasi dengan Folium:**

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | import pandas as pd  import folium  # Load data hasil MapReduce  data = pd.read\_csv('hasil\_mapreduce.csv')  # Membuat peta  peta = folium.Map(location=[-7.978, 112.63], zoom\_start=12)  # Menambahkan marker untuk setiap kelurahan  for index, row in data.iterrows():  folium.CircleMarker(  location=[row['latitude'], row['longitude']],  radius=row['jumlah\_pendukung'] \* 0.1,  color='blue',  fill=True,  fill\_opacity=0.6,  popup=f"{row['kelurahan']}: {row['jumlah\_pendukung']} pendukung"  ).add\_to(peta)  # Simpan peta ke file HTML  peta.save('peta\_calon\_pemilih.html') |
| --- | --- |

**4. Pengujian dan Hasil**

**4.1 Pengujian**

* Input: File CSV data pemilih.
* Proses: Jalankan MapReduce, olah data dengan Python, buat peta dengan Folium.
* Output: File HTML peta interaktif.

**4.2 Hasil**

* Peta menunjukkan jumlah pendukung di setiap kelurahan.
* Marker peta interaktif menampilkan detail jumlah pendukung saat di-klik.

**5. Analisis dan Evaluasi**

**5.1 Evaluasi Hasil**

Proses berjalan lancar dan hasil sesuai ekspektasi.

**5.2 Kendala dan Solusi**

* Kendala: Koordinat kelurahan tidak tersedia.
* Solusi: Gunakan layanan geokode untuk mendapatkan koordinat.

**6. Kesimpulan dan Rekomendasi**

**6.1 Kesimpulan**

Proyek berhasil memvisualisasikan data pemilih menggunakan Hadoop dan Python.

**6.2 Rekomendasi**

**Untuk pengembangan lebih lanjut:**

* Tambahkan analisis tren pendukung berdasarkan waktu.
* Implementasikan API untuk integrasi data real-time.

# 