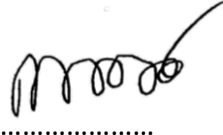
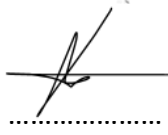




LAPORAN TUGAS AKHIR

PROTOTYPE APLIKASI EARLY WARNING SYSTEM GUNUNG MELETUS DAERAH MERAPI DAN SEKITARNYA



Matakuliah	TI0263 – Kecerdasan Buatan (Grup A) - Genap 2021/2022
Dosen Pengampu	Matahari Bhakti Nendya, S.Kom., M.T
Nama Kelompok	Kelompok Ho'o (6)
Anggota Kelompok	<div>1. Maximiliano Bagas (71200535) </div> <div>2. Ignatius Barry (71200577) </div> <div>3. Christina Andrea (71200620) </div> <div>4. Joiner Tenny Ariel (71200651) </div>
Deklarasi	Dengan ini kami menyatakan bahwa tugas ini merupakan hasil karya kelompok kami, tidak ada manipulasi data serta bukan merupakan plagiasi dari karya orang lain.



UNIVERSITAS KRISTEN DUTA WACANA
Fakultas Teknologi Informasi
Program Studi Informatika



Pendahuluan

Gunung Merapi merupakan gunung berapi aktif di perbatasan Jawa Tengah dan Yogyakarta. Melihat kondisi gunung Merapi saat ini yang masih terus berada di posisi siaga, satu level dibawah level tertinggi dan dapat Meletus kapan saja, karena kita tidak dapat memperhitungkan secara pasti faktor alam yang mungkin memicu erupsi tiba-tiba. Melihat permasalahan tadi kami mencoba membuat prototype aplikasi yang dapat memberikan early warning kalau kondisi gunung Merapi mengalami perubahan signifikan yang tiba-tiba dan berkemungkinan besar untuk terjadi erupsi, **prototype** ini kami beri judul **Merapi Early Warning System**. Sasaran dari aplikasi Merapi Early Warning System adalah wilayah sekitar 10-15 kilo meter dari gunung Merapi, pertimbangannya karena itu merupakan daerah paling krusial yang mungkin terdampak erupsi, dan memerlukan warning paling cepat dan sesegera mungkin, aplikasi yang kami buat merupakan prototype yang nantinya dapat dijadikan dasar ide untuk aplikasi yang lebih canggih dan serius, bisa dengan memasang alarm di sekitar wilayah sasaran tadi, kemudian aplikasi akan menggunakan data yang didapat dari sensor yang diletakkan di sekitar kawah gunung Merapi dan memperhitungkan kemungkinan terjadinya erupsi, berdasarkan aturan-aturan yang ditetapkan di production system aplikasi yang telah dibuat, jika data yang realtime yang diperoleh telah memenuhi kondisi tertentu, alarm yang telah dipasang tadi akan memberikan semacam sinyal berupa announcer yang akan memberikan arahan pengungsian tergantung dengan posisi alarm tadi dipasang, bisa juga dijadikan dasar aplikasi mobile dengan fungsi yang sama, namun bisa menyesuaikan dengan posisi user berdasarkan lokasi GPS di perangkat enduser. Aplikasi **Merapi Early Warning System** yang sudah kami buat saat ini masih terbatas pada aplikasi console namun memiliki beberapa fungsi utama untuk membentuk aplikasi early warning system, aplikasi kami akan dapat memberikan data terkait kondisi gunung Merapi, dan memberikan saran lokasi pengungsian terbaik berdasarkan pada posisi user saat ini walaupun implementasinya masih serba manual, lalu karena tujuan aplikasi ini hanya untuk skala kecil, kami baru mendaftarkan 3 titik pengungsian yang mungkin dituju oleh user jika erupsi terjadi.

Representasi Pengetahuan

Frame

Frame merupakan representasi yang terdiri dari kumpulan atribut dan nilainya untuk menggambarkan suatu objek. Pada aplikasi ini kami menggunakan representasi pengetahuan frame untuk menyimpan data set kami, contohnya pada penyimpanan koordinat lokasi pengungsian.

```
data_location = {"Name":["Sawangan","Dukun","Mranggen"],  
                 "Longitude":["110.39235831626016","110.37533734339334","110.3400719"],  
                 "Latitude":["-7.493767950000005","-7.54226065","-7.5859438"]}
```

Metode Pengambilan Keputusan

Forward Chaining

Pada aplikasi ini kami menggunakan percabangan dengan pendekatan forward chaining (data driven) sebagai dasar algoritma aplikasi ini, kami memilih menggunakan forward chaining karena dalam kasus ini kami perlu untuk mencari solusi yang dapat diberikan sesuai dengan kondisi user masing-masing, dan kami sudah menyiapkan rule-rule production system yang kami sesuaikan dengan tree yang telah kami buat pada tugas laporan sebelumnya, jadi kami rasa forward chaining merupakan pilihan yang tepat.

Modul

Geopy

Aplikasi ini memanfaatkan geopy, yang digunakan sebagai metode untuk mencari jarak antara lokasi dan user, serta memberikan user rekomendasi tempat evakuasi berdasarkan perbandingan jarak terdekat user dan masing-masing tempat evakuasi. Cara untuk mendapatkan jarak antar lokasi tersebut adalah menggunakan koordinat yang diambil dari modul geopy untuk memberikan data lokasi perangkat enduser, dan memperhitungkan jaraknya dengan lokasi gunung Merapi dengan untuk selanjutnya digunakan dalam pengambilan keputusan

Random

Pada aplikasi ini kami menggunakan modul random untuk mendapatkan data mengenai keadaan gunung Merapi, karena kami tidak menemukan data-set yang cocok untuk kasus aplikasi kami, dan karena keterbatasan biaya, waktu dan SDM untuk merealisasikan pemasangan sensor di lokasi sesungguhnya.

Cara Kerja

Alur aplikasi :

1. User input lokasinya (address, city, country)
2. Sistem akan menghitung jarak antara lokasi user dan Merapi, jika berada di luar jangkauan maka akan mengeluarkan pesan aman
3. Sistem akan melakukan random di setiap parameter data yang ada dan output yang akan keluar akan sesuai dengan parameter-parameter yang ada
4. Jika status yang keluar adalah Waspada atau Siaga, maka sistem akan mengeluarkan tempat evakuasi terdekat dari jarak user.

Source Code

```
from geopy.geocoders import Nominatim
from geopy.distance import geodesic
import random

def evacuation(d1,d2,d3):
    if d1 < d2 or d1 < d3:
        print("Best evacuation is in Sawangan")
    elif d2 < d3 or d2 < d1:
        print("Best evacuation is in Dukun")
    elif d3 < d1 or d3 < d2:
        print("Best evacuation is in Mranggen")

def kondisi():
    print ("Kondisi Gunung Berapi")
    print("-----")
    print ("Getaran : ", getaran)
    print ("Deformasi Tanah : ", deformasi_tanah)
    print ("Aktivitas Kawah : ", kawah)
```

```

    if kawah == "Berkabut":
        print ("Warna Asap : ", warna)
        print ("Ketebalan Asap : ", tebal)

def randomise (data, key):
    x = random.choice(data[key])
    return x

def tuple_gen(var1,var2):
    emp1 = []
    emp1.append(float(var2))
    emp1.append(float(var1))
    tup = tuple(emp1)
    return tup

def cek_aktivitas(dist_mount, deformasi_tanah, getaran, kawah, warna, tebal):
    if dist_mount<=10.0:
        kondisi()
        if getaran == "< 1.5 SR":
            if kawah == "Berkabut":
                if warna == "Terang":
                    if tebal == True:
                        print("Status: Normal")
                    else:
                        print("Status: Normal")
                else:
                    if tebal == True:
                        print("Status: Normal")
                    else:
                        print("Status: Normal")
            else:
                print("Status: Normal")
        elif getaran == "1.5 SR - 3.5 SR":
            if deformasi_tanah == True:
                if kawah == "Berkabut":
                    if warna == "Terang":
                        if tebal == True:
                            print("Status: Waspada")
                            print("Disarankan Evakuasi")
                            evacuation(dist1, dist2, dist3)
                        else:
                            print("Status: Waspada")
                            print("Disarankan Evakuasi")
                            evacuation(dist1, dist2, dist3)
                    else:
                        if tebal == True:
                            print("Status: Siaga")

```

```

        print("Silahkan Mencari Tempat Evakuasi
Terdekat")

        evacuation(dist1, dist2, dist3)
    else:
        print("Status: Waspada")
        print("Disarankan Evakuasi")
        evacuation(dist1, dist2, dist3)
    else:
        print("Status: Normal")
else:
    if kawah == "Berkabut":
        if warna == "Terang":
            if tebal == True:
                print("Status: Normal")
            else:
                print("Status: Normal")
        else:
            if tebal == True:
                print("Status: Waspada")
                print("Disarankan Evakuasi")
                evacuation(dist1, dist2, dist3)
            else:
                print("Status: Waspada")
                print("Disarankan Evakuasi")
                evacuation(dist1, dist2, dist3)
        else:
            print("Status: Normal")
elif getaran == "> 3.5 SR":
    if deformasi_tanah == True:
        if kawah == "Berkabut":
            if warna == "Gelap":
                if tebal == True:
                    print("Status: Siaga")
                    print("Silahkan Mencari Tempat Evakuasi
Terdekat")

                    evacuation(dist1, dist2, dist3)
            elif tebal == False:
                print("Status: Waspada")
                print("Disarankan Evakuasi")
                evacuation(dist1, dist2, dist3)
            elif warna == "Terang":
                if tebal == True:
                    print("Status: Waspada")
                    print("Disarankan Evakuasi")
                    evacuation(dist1, dist2, dist3)
            elif tebal == False:
                print("Status: Waspada")
                print("Disarankan Evakuasi")
                evacuation(dist1, dist2, dist3)

```

```

        elif kawah == "Tidak Berkabut":
            print("Status: Normal")

    elif deformasi_tanah == False:
        if kawah == "Berkabut":
            if warna == "Gelap":
                if tebal == True:
                    print("Status: Waspada")
                    print("Disarankan Evakuasi")
                    evacuation(dist1, dist2, dist3)
                elif tebal == False:
                    print("Status: Waspada")
                    print("Disarankan Evakuasi")
                    evacuation(dist1, dist2, dist3)
            elif warna == "Terang":
                if tebal == True:
                    print("Status: Normal")
                elif tebal == False:
                    print("Status: Waspada")
                    evacuation(dist1, dist2, dist3)
            else:
                print("Status: Normal")
    else:
        kondisi()
        print("Anda berada diluar jangkauan letusan")

try:
    geolocator = Nominatim(user_agent="GetLoc")
    addr = input("Address : ")
    city = input("City : ")
    country = input("Country : ")
    location = geolocator.geocode(addr+","+city+","+country)
    longUser = float(location.latitude)
    latUser = float(location.longitude)
    mount = "Gunung Merapi,Yogyakarta,Indonesia"
    mount_loc = geolocator.geocode(mount)
    mount_lat = float(mount_loc.latitude)
    mount_long = float(mount_loc.longitude)

    data_location = {"Name":["Sawangan","Dukun","Mranggen"],
                    "Longitude":["110.39235831626016","110.37533734339334","110.3400719"],
                    "Latitude":["-7.4937679500000005","-7.54226065","-7.5859438"]}

    emp = {}

    for i in range(len(data_location)):
        loc = data_location["Name"][i]
        long = data_location["Longitude"][i]

```

```

        lat = data_location["Latitude"][i]
        emp[loc] = tuple_gen(long,lat)

    locUser = tuple_gen(latUser,longUser)
    loc1 = emp['Sawangan']
    loc2 = emp['Dukun']
    loc3 = emp['Mranggen']
    locMountain = tuple_gen(mount_long,mount_lat)
    dist1 = geodesic(locUser,loc1).km
    dist2 = geodesic(locUser,loc2).km
    dist3 = geodesic(locUser,loc3).km
    dist_mount = geodesic(locUser,locMountain).km

    data = {"Category" : ["High","Intermediate","Low"],
            "Deformasi Tanah":[True, False],
            "Getaran":["< 1.5 SR", "1.5 SR - 3.5 SR", "> 3.5 SR"],
            "Aktivitas Kawah":["Berkabut","Tidak Berkabut"],
            "Warna Asap":["Terang","Gelap"],
            "Ketebalan Asap":[True,False]}

    deformasi_tanah = randomise(data, "Deformasi Tanah")
    getaran = randomise(data, "Getaran")
    kawah = randomise(data, "Aktivitas Kawah")
    warna = randomise(data, "Warna Asap")
    tebal = randomise(data,"Ketebalan Asap")

    cek_aktivitas(dist_mount,deformasi_tanah, getaran, kawah, warna, tebal)
except:
    print("Cek kembali data anda")

```