

Judul Paten : **APLIKASI CERDAS UNTUK PENCACAH KENDARAAN BERMOTOR**

Penulis : **Puguh Budi Prakoso dan Yuslena Sari**

No. Paten : **EC00202148415**

Alamat website : -

Deskripsi : Pada aplikasi cerdas ini dapat melakukan perhitungan jumlah kendaraan bergerak berdasarkan jenisnya berbasis computer vision dan kecerdasan buatan.

REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202148415, 20 September 2021

Pencipta

Nama : **Puguh Budi Prakoso dan Yuslena Sari**
Alamat : Fakultas Teknik, Universitas Lambung Mangkurat Jl. Bridgen Hasan Basry Kayutangi, Banjarmasin, KALIMANTAN SELATAN, 70123
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Puguh Budi Prakoso dan Yuslena Sari**
Alamat : Fakultas Teknik, Universitas Lambung Mangkurat Jl. Bridgen Hasan Basry Kayutangi, Banjarmasin, KALIMANTAN SELATAN, 70123
Kewarganegaraan : Indonesia

Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **APLIKASI CERDAS UNTUK PENCACAH KENDARAAN BERMOTOR**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 18 September 2021, di Banjarmasin

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000275818

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

Disclaimer

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.



APLIKASI CERDAS UNTUK PENCACAH KENDARAAN BERMOTOR

Oleh:

PUGUH B. PRAKOSO

YUSLENA SARI

FAKULTAS TEKNIK

UNIVERSITAS LAMBUNG MANGKURAT

BANJARMASIN

2021

DAFTAR ISI

COVER.....	1
DAFTAR ISI.....	2
BAB 1 PROGRAM.....	3
1.1 IMPLEMENTASI FUZZY CLUSTERING MEANS	3
BAB 2 PROGRAM MANUAL	10
2.1 HALAMAN UTAMA	10
2.2 HALAMAN DETAIL.....	10
2.3 HALAMAN INPUT VIDEO.....	11

BAB 1

PROGRAM

1.1 IMPLEMENTASI FUZZY CLUSTERING MEANS

Implementasi ini merupakan tahapan akhir dan inti dari program. Objek-objek yang sudah melalui proses pengolahan data dan analisis *blob* selanjutnya akan diidentifikasi menggunakan metode *Fuzzy Clustering Means* untuk diklasifikasikan dan dihitung jumlah kendaraan yang terdeteksi berdasarkan jenisnya. Kode program *Fuzzy Clustering Means* adalah sebagai berikut.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace WindowsFormsApplication1
{
    class FCM
    {
        private int cluster;
        private int maxIterasi;
        private int bobot;
        private double epsilon;
        private List<double[]> data;
        //private double[,] data;
        private double[,] anggota_cluster;
        private double[,] miu_kuadrat;
        private double[,] miu_x1;
        private double[,] sum_kuadrat;
        private double[,] sum_acak_kuadrat;
        private List<double[,]> list_miu_x;
        private double[,] centroid;
        private double[,] x_v;
        private double[,] hasil_kali_centroid;
        private double[,] sum_kali_centroid;
        private double objektif = 0;
        private double prev_objektif = 0;
        private double selisih_objektif = 0;
        private double[,] kali_centroid_kuadrat;
        private double[,] sum_centroid_kuadrat;
        //private double[,] anggota_baru;
        private string[,] hasil_clustering;

        public FCM()
        {
        }
        public FCM(List<double[]> data, int cluster, int maxIterasi, int
        bobot, double epsilon)
        {
            this.cluster = cluster;
            this.maxIterasi = maxIterasi;
            this.bobot = bobot;
            this.epsilon = epsilon;
            this.data = data;
        }
    }
}
```

```

        list_miu_x = new List<double[,]>();
    }

```

Langkah 1 Acak Anggota *Cluster*

```

public void AcakAnggotaCluster()
{
    int jml_data = data.Count;

    anggota_cluster = new double[jml_data, cluster];

    Random acak = new Random(4);

    for (int i = 0; i < jml_data; i++)
    {
        double total = 0;

        for (int j = 0; j < cluster; j++)
        {
            double test;

            if (j == cluster - 1)
            {
                test = 1 - total;
                total += test;
            }
            else
            {
                do
                {
                    test = acak.NextDouble();
                    test = test / 5;
                    test = Math.Round(test, 2);
                    if (test != 1 && test != 0 && (total + test) < 1)
                    {
                        break;
                    }
                } while (true);

                total += test;
            }

            anggota_cluster[i, j] = test;
        }
    }
}

public static void cetakArray(double[,] data)
{
    int baris = data.GetLength(0);
    int kolom = data.GetLength(1);

    for (int i = 0; i < baris; i++)
    {
        for (int j = 0; j < kolom; j++)
        {
            Console.Write(data[i, j] + " \t ");
            //Console.WriteLine(Environment.NewLine);
        }
        Console.WriteLine();
    }
}

public static void cetakArray(string[,] data)

```

```

    {
        int baris = data.GetLength(0);
        int kolom = data.GetLength(1);

        for (int i = 0; i < baris; i++)
        {
            for (int j = 0; j < kolom; j++)
            {
                Console.Write(data[i, j] + " \t ");
                //Console.WriteLine(Environment.NewLine);
            }
            Console.WriteLine();
        }
    }
}

public static void cetakList(List<double[]> data)
{
    int baris = data.Count;
    int kolom = data.ElementAt(0).Length;

    for (int i = 0; i < baris; i++)
    {
        for (int j = 0; j < kolom; j++)
        {
            Console.Write(data[i][j] + " \t ");
            //Console.WriteLine(Environment.NewLine);
        }
        Console.WriteLine();
    }
}

```

Langkah 2 Menentukan Kuadrat

```

public void Kuadrat()
{
    bobot = 2;
    int jml_data = data.Count;
    sum_acak_kuadrat = new double[1, cluster];
    miu_kuadrat = new double[jml_data, cluster];

    for (int i = 0; i < jml_data; i++)
    {
        for (int j = 0; j < cluster; j++)
        {
            miu_kuadrat[i, j] = Math.Pow(anggota_cluster[i, j],
            bobot);
            sum_acak_kuadrat[0, j] += miu_kuadrat[i, j];
        }
    }
    Console.WriteLine("\nKuadrat Acak");
    cetakArray(miu_kuadrat);

    Console.WriteLine("\nSum Kuadrat Acak");
    cetakArray(sum_acak_kuadrat);
}

```

Langkah 3, Total semua hasil kuadrat

```

public void sum_kuadrat_x()
{
    int pjpg_data = data.Count;
    int lbr_data = data[0].Length;

```

```

sum_kuadrat = new double[cluster, lbr_data];

for (int k = 0; k < cluster; k++)
{
    Console.WriteLine("\n Kuadrat Miu X" + (k+1) + "\n");
    miu_x1 = new double[pjg_data, lbr_data];
    for (int i = 0; i < pjg_data; i++)
    {
        for (int j = 0; j < lbr_data; j++)
        {
            miu_x1[i, j] = data[i][j] * miu_kuadrat[i, k];
            sum_kuadrat[k, j] += miu_x1[i, j];
        }
    }
    cetakArray(miu_x1);
    Console.WriteLine();
    list_miu_x.Add(miu_x1);
}

Console.WriteLine("Sum Kuadrat Tiap Miu X");
cetakArray(sum_kuadrat);
}

public void Sum_Array()
{
    int lbr_data = data[0].Length;
    double[] sum_arr = new double[lbr_data];
}

```

Langkah 4 Menghitung *Centroid*

```

public void hitung_centroid()
{
    int lbr_data = data[0].Length;
    centroid = new double[cluster, lbr_data];

    for (int i = 0; i < cluster; i++)
    {
        for (int j = 0; j < lbr_data; j++)
        {
            centroid[i, j] = sum_kuadrat[i, j] / sum_acak_kuadrat[0, i];
        }
    }
    Console.WriteLine("Centroid / Pusat Cluster");
    cetakArray(centroid);
}

```

Langkah 5, Menjumlahkan kuadrat dari data – *centroid*

```

public void data_centroid()
{
    int jml_data = data.Count;
    int lbr_data = data[0].Length;
    x_v = new double[jml_data, cluster];
    for (int i = 0; i < jml_data; i++)
    {
        for (int j = 0; j < cluster; j++)
        {
            for (int k = 0; k < lbr_data; k++)
            {

```



```

        x_v[i, j] += Math.Pow((data[i][k] - centroid[j, k]),2);
    }
}
Console.WriteLine("Jumlah data centroid");
cetakArray(x_v);
}

```

Langkah 6, Kalikan jumlah data *centroid* dengan miu kuadrat

```

public void Kali_centroid()
{
    bobot = 2;
    int jml_data = data.Count;
    hasil_kali_centroid = new double[jml_data, cluster];
    sum_kali_centroid = new double[jml_data, 1];
    kali_centroid_kuadrat = new double[jml_data, cluster];
    sum_centroid_kuadrat = new double[jml_data, 1];

    for (int i = 0; i < jml_data; i++)
    {
        for (int j = 0; j < cluster; j++)
        {
            hasil_kali_centroid[i, j] = x_v[i, j] * miu_kuadrat[i, j];

            sum_kali_centroid[i, 0] += hasil_kali_centroid[i, j];

            kali_centroid_kuadrat[i, j] = Math.Pow(x_v[i, j], (-1 / (bobot - 1)));

            sum_centroid_kuadrat[i, 0] += kali_centroid_kuadrat[i, j];
        }
        for (int j = 0; j < cluster; j++)
        {
            anggota_cluster[i, j] = kali_centroid_kuadrat[i, j] /
            sum_centroid_kuadrat[i, 0];
        }

        objektif += sum_kali_centroid[i, 0];
    }
    selisih_objektif = Math.Abs(objektif - prev_objektif);
    prev_objektif = objektif;

    Console.WriteLine("Hasil Kali Data Centroid");
    cetakArray(hasil_kali_centroid);

    Console.WriteLine("\nTotal Kali Data Centroid");
    cetakArray(sum_kali_centroid);

    Console.WriteLine("\n Fungsi Objektif: " + objektif);
    Console.WriteLine("\n Selisih Objektif: " + selisih_objektif);

    Console.WriteLine("\n Data Perbaikan Anggota Cluster");
    cetakArray(kali_centroid_kuadrat);

    Console.WriteLine("\nTotal Data Perbaikan");
    cetakArray(sum_centroid_kuadrat);

    Console.WriteLine("\nAnggota Cluster Baru");
    cetakArray(anggota_cluster);
}

```

Selanjutnya setiap objek yang telah diidentifikasi akan diklasifikasikan berdasarkan labelnya dan dihitung jumlahnya berdasarkan jenisnya. Berikut ini kode program pengklasifikasian jenis kendaraan.

Langkah 7 Proses *Clustering*

```
public void Clustering()
{
    int jml_data = data.Count;
    hasil_clustering = new string[jml_data, cluster];

    for (int i = 0; i < jml_data; i++)
    {
        double cekNilai = 0;
        int indexI = 0, indexJ = 0;
        for (int j = 0; j < cluster; j++)
        {
            if (anggota_cluster[i, j] > cekNilai)
            {
                cekNilai = anggota_cluster[i, j];
                indexI = i;
                indexJ = j;
            }
        }
        hasil_clustering[indexI, indexJ] = "C" + (indexJ+1);
    }
    Console.WriteLine("\n Anggota Cluster Terakhir yang
    Digunakan");
    cetakArray(anggota_cluster);

    Console.WriteLine("\n List Data");
    cetakList(data);

    Console.WriteLine("\n Hasil Clustering");
    cetakArray(hasil_clustering);
}

public void Run()
{
    this.AcakAnggotaCluster();
    Console.WriteLine();

    int i = 0;
    while (i < maxIterasi)
    {
        Console.WriteLine("\nIterasi Ke-" + (i + 1));
        Console.WriteLine("Anggota Cluster");
        cetakArray(anggota_cluster);
        this.Kuadrat();
        Console.WriteLine();
        this.sum_kuadrat_x();
        Console.WriteLine();
        this.hitung_centroid();
        Console.WriteLine();
        this.data_centroid();
        Console.WriteLine();
        this.Kali_centroid();
    }
}
```

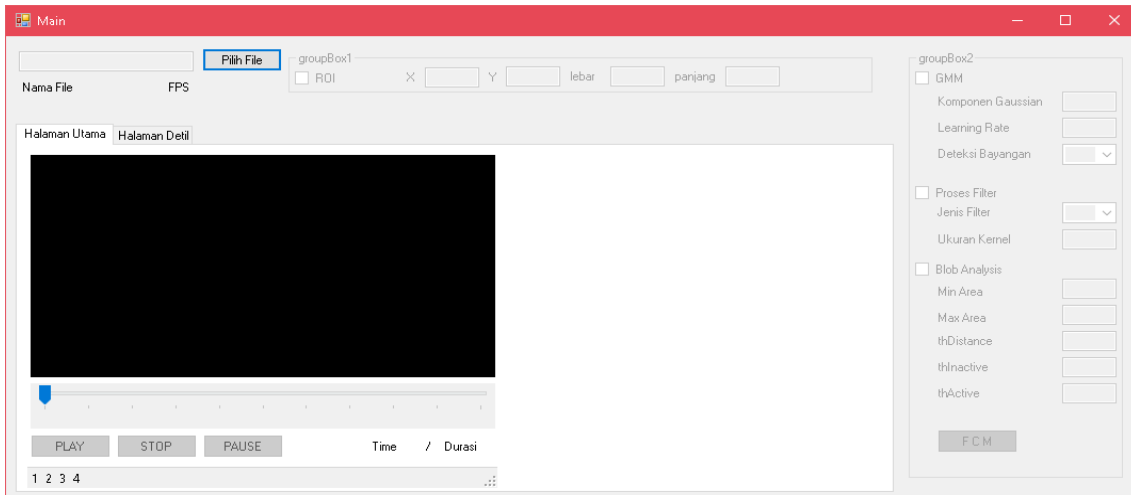
```
        if (selisih_objektif < epsilon)
        {
            break;
        }
        i++;
    }
    Console.WriteLine();
    this.Clustering();
}
}
```

BAB 2

PROGRAM MANUAL

2.1 HALAMAN UTAMA

Halaman Utama pada Aplikasi Perhitungan Jumlah Kendaraan Bergerak Berdasarkan Jenisnya dapat dilihat pada Gambar 1 berikut.

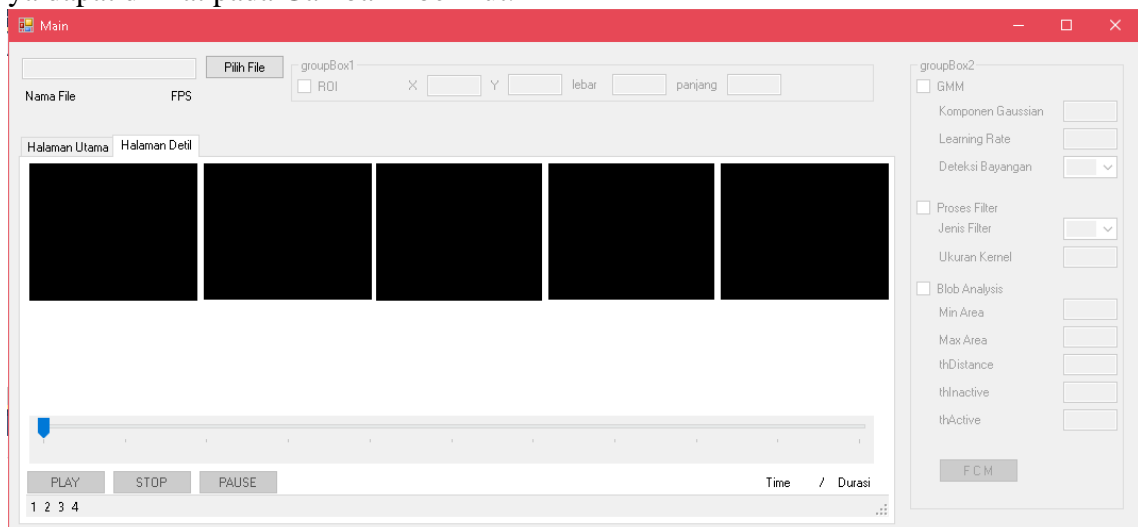


Gambar 1. Halaman Utama Aplikasi Perhitungan Jumlah Kendaraan Bergerak Berdasarkan Jenisnya

Pada halaman ini dilakukan proses pemilihan file video yang akan diproses, pemilihan ROI dan mengisi parameter proses. Pada halaman ini akan menampilkan video hasil klasifikasi jenis kendaraan dan hasil perhitungan setiap jenis kendaraan.

2.2 HALAMAN DETAIL

Halaman Detail pada Aplikasi Perhitungan Jumlah Kendaraan Bergerak Berdasarkan Jenisnya dapat dilihat pada Gambar 2 berikut.

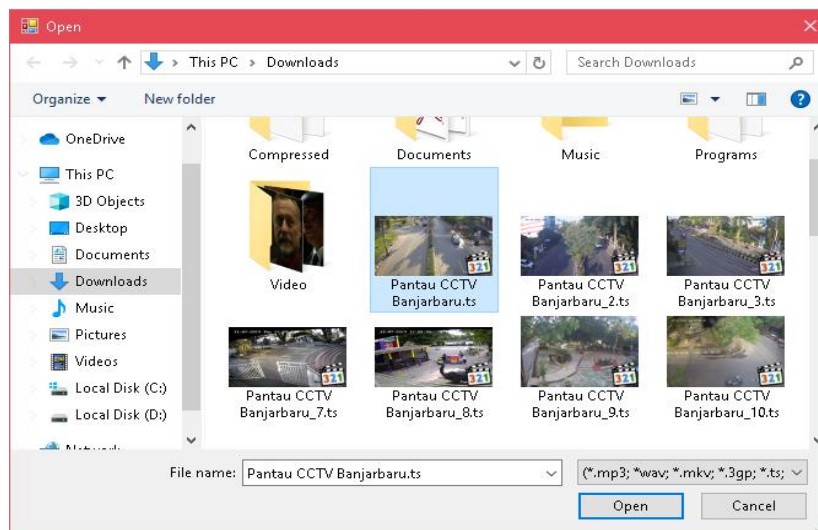


Gambar 2. Halaman Detail Aplikasi Perhitungan Jumlah Kendaraan Bergerak Berdasarkan Jenisnya

Pada halaman detail ini ditampilkan video *input*, video citra *foreground* hasil GMM, video citra *foreground* yang telah difilter, video citra *foreground* hasil analisis *blob* dan *shadow removal*, dan video *output* hasil klasifikasi.

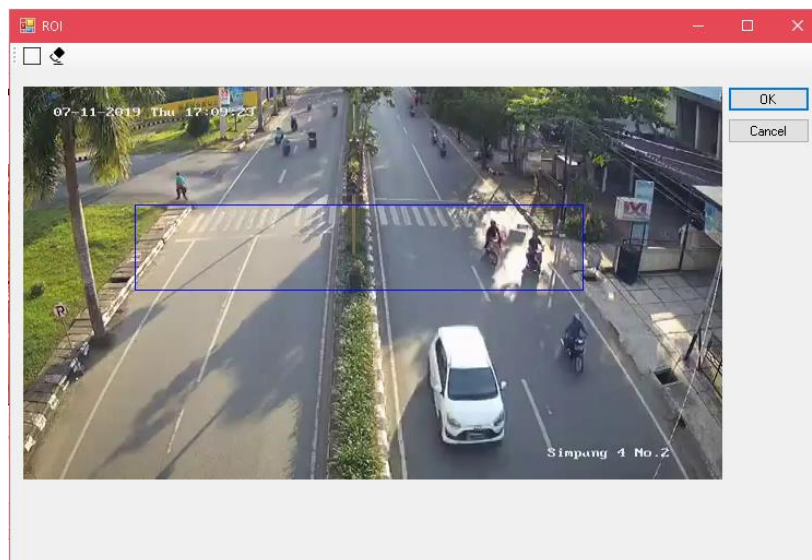
2.3 HALAMAN INPUT VIDEO

Data yang digunakan sebagai data masukan berupa video rekaman CCTV dari beberapa jalan dua arah di wilayah Banjarbaru, maka dari itu untuk menjalankan aplikasi ini kita perlu menginputkan salah satu *file* video yang sudah tersedia di *directory* terlebih dahulu dan kemudian akan diproses pada tahap selanjutnya.



Gambar 3. Antarmuka input video

Penentuan ukuran ROI pada sistem ini dianjurkan memuat setengah dari ukuran *frame* inputan. Karena program hanya mengolah informasi dari bagian piksel yang dibutuhkan saja.



Gambar 4. Kotak Biru merupakan area ROI