

# LAMPIRAN



PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI**  
**FAKULTAS TEKNOLOGI INDUSTRI**

**Nama : Nayla Dwi Salsabila**  
**Nim : 2118062**  
**Jurusan : Teknik Informatika**  
**Judul : Penerapan Algoritma K-Means Untuk Klasterisasi  
Produktivitas Tanaman Jahe Berdasarkan Luas Panen Dan  
Jumlah Produksi Di Kabupaten Sumenep**

Dipertahankan Dihadapan Majelis Penguji Skripsi Jenjang Strata Satu(S-1)  
Pada

**Hari : Rabu**  
**Tanggal : 22 Januari 2025**  
**Nilai : 70**

Panitia Ujian Skripsi :  
Ketua Majelis Penguji

Yosep Agus Pranoto ST., MT  
NIP.P 1031000432

Anggota Penguji :

Dosen/Penguji I

Dr. Agung Panji Sasmito, S.Pd, M.Pd  
NIP.P 1031500499

Dosen Penguji II

Deddy Rudhistiar S.Kom., M.Cs  
NIP.P 1032000578



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**FORMULIR PERBAIKAN SKRIPSI**

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : Nayla Dwi Salsabila  
Nim : 2118062  
Jurusan : Teknik Informatika  
Judul : Penerapan Algoritma K-Means Untuk Klusterisasi Produktivitas Tanaman Jahe Berdasarkan Luas Panen Dan Jumlah Produksi Di Kabupaten Sumenep

No.	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	22 Januari 2025	1. Abstrak 2. Latar Belakang 3. Kebutuhan Fungsional. 4. Use Case Diagram. 5. DFD. 6. Cluster Apa Cuma 2. 7. Pengujian User.	
2.	Penguji II	22 Januari 2025	1. Setiap Tabel Diberikan Penjelasan. 2. Perhitungan 3 Cluster. 3. Tabel Hasil Cluster Tiap Tahun. 4. Perhitungan Evaluasi Masuk Disistem	

Anggota Penguji :

~~Dosen Penguji I~~

~~Dr. Agung Panji Sasmito, S.Pd, M.Pd  
NIP.P 1031500499~~

Dosen Penguji II

Deddy Rudhristiar S.Kom., M.Cs  
NIP.P 1032000578

Mengetahui :

Dosen Pembimbing I

Karina Auliasari ST., M.Eng  
NIP.P 1031000426

Dosen Pembimbing II

Hani Zulfia Zahro S.Kom., M.Kom  
NIP.P 1031500480



PT BNI (PERSERO) MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**  
FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Panting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 Oktober 2024

Nomor : ITN-883/III.INF/TA/2024

Lampiran : ---

Perihal : Pembimbing Utama Skripsi

Kepada : Yth. Bpk/Ibu Karina Auliasari ST., M.Eng.  
Dosen Program Studi Teknik Informatika S-1  
Institut Teknologi Nasional  
Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : Nayla Dwi Salsabila  
Nim : 2118062  
Prodi : Teknik Informatika S-1  
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

**21 Agustus 2024 s/d 21 Pebruari 2025**

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui  
Program Studi Teknik Informatika S-1  
Ketua,

Yosep Agus Pranoto, ST., MT.  
NIPP. 1031000432

Form 5-4a



PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 Oktober 2024

Nomor : ITN-883/III.INF/TA/2024  
Lampiran : ---  
Perihal : Pembimbing Pendamping Skripsi

Kepada : **Yth. Bpk/Ibu Hani Zulfia Zahro' S.Kom., M.Kom.**  
Dosen Pembina Program Studi Teknik Informatika S-1  
Institut Teknologi Nasional  
Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : Nayla Dwi Salsabila  
Nim : 2118062  
Prodi : Teknik Informatika S-1  
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

**21 Agustus 2024 s/d 21 Februari 2025**

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui  
Program Studi Teknik Informatika S-1  
Ketua,

Yosep Agus Pranto, ST., MT.  
NIP. 1031000432

Form S-4a



INSTITUT TEKNOLOGI NASIONAL MALANG  
Fakultas Teknologi Industri  
Program Studi Teknik Informatika S1

## FORMULIR BIMBINGAN SKRIPSI

Nama : Mayla Du Salsabila  
Nim : 2118062  
Masa Bimbingan : 21 Agustus 2024 - 21 Februari 2025  
Judul Skripsi : Penerapan Metode k-means untuk Klasifikasi Profitabilitas Tanaman Jabe berdasarkan Kac Pagan dan Jumlah Produksi di Kabupaten Gunungrejo.

No.	Tanggal	Uraian	Paraf Pembimbing
1	9-10-2024	Revisi Judul	li
2	11-10-2024	Literature Review & Pengumpulan Data	li
3	16-10-2024	Simulasi Metode	li
4	24-10-2024	Database & Web Dashboard	li
5	29-10-2024	CRUD pada sistem	li
6	4-11-2024	Revisi Laporan Bab III & IV	li
7	16-11-2024	Membuat paper jurnal INFOTEK	li
8	21-11-2024	Submit jurnal INFOTEK	li
9	28-11-2024	Revisi hasil pengujian blackbox	li
10	17-1-2025	Acc Laporan & Acc Maju Kompre	li

Malang, 9-10-2024  
Dosen Pembimbing

  
(Karina Auliasari, ST.,) M.Eng.  
NIP. P 1031000426

## FORMULIR BIMBINGAN SKRIPSI

Nama : Nayla Aul Salsabila  
 Nim : 2118062  
 Masa Bimbingan : 21 Agustus 2024 - 21 Februari 2025  
 Judul Skripsi : Penerapan Metode k-means untuk Klasterisasi produktivitas tanaman Jaito berdasarkan luas panen dan jumlah produksi di Kabupaten Kuning.

No.	Tanggal	Uraian	Paraf Pembimbing
1.	23/10/24	Proposal clustering & bll. Acc	H.
2.	31/10/24	Flouchart, Uraian, Daftar pustaka.	H.
3.	1/11.2024.	Desain hiker. tampilan luas lahan & produksi ga. & tampilan. Analisa Analisa	H.
4.	29/11.2024	Demo program. Pengguna aplikasi belum jelas	H.
5.	3/12.2024	Revisi Laporan kemitra - program belum selesai	H.
6.	7/12.2024	Revisi Program	H.
7.	17/12.2024	Demo Program. Hasil kemitra	H.
8.	18/1.2025.	Laporan Revisi	H.
9.	16/1.2025	Laporan Revisi program blm Acc.	H.
10.	21/1-2025	Acc kompie	H.

Malang, 16.1.2025.

Dosen Pembimbing

  
 Hani Zulfia Zahro, S.Kom, M.Kom  
 NIP. P. 1051500480



BADAN PUSAT STATISTIK KABUPATEN SUMENEP

SURAT PERNYATAAN

Yang bertandatangan di bawah ini:

Nama : DIMAS DWI JUHANSTYAH  
Jabatan : STATISTISI AHLI PERTAMA  
Instansi/Lembaga : BPS Kabupaten SUMENEP  
No. Telepon/HP : 081571 882431  
Alamat : SUMENEP

Dengan ini menyatakan bersedia menjadi mitra dalam pembuatan skripsi dari

Nama : Nayla Dwi Salsabila  
NIM : 2118062  
Judul Skripsi : Penerapan Algoritma K-means untuk Klusterisasi Produktivitas Tanaman Jahe Berdasarkan Luas Panen dan Jumlah Produksi di Kabupaten Sumenep

Dan saya menyatakan bahwa data yang digunakan dalam skripsi tersebut adalah benar dan boleh dipublikasikan. Demikian surat pernyataan ini saya buat tanpa ada paksaan dan tekanan dari pihak manapun serta dapat dipergunakan sebagaimana mestinya.

Catatan :

DATA TANAMAN JAHE 2020-2023

Sumenep, 14 Desember 2024

Yang membuat pernyataan

  
METERAI TEMPEL  
Rp 10.000  
Dwi Juhanstyah





INSTITUT TEKNOLOGI NASIONAL MALANG  
PERPUSTAKAAN PUSAT  
Jln. Bendungan Sigura-gura No.2 Malang 65145  
Telp. (0341) 551431 Pcs. 863-145-147 Fax. (0341) 553015 Website: [www.itn.ac.id](http://www.itn.ac.id)

### FORM UJI PLAGIASI UNTUK MAHASISWA

Yang bertandatangan di bawah ini, Mahasiswa Institut Teknologi Nasional Malang:

Nama : Nayla Dwi Salsabila  
NIM : 218062  
Fakultas / Jurusan : Teknologi Industri / Teknik Informatika S-1  
Email : dwisalsabila101@gmail.com  
No. Tlp : 08971623160  
Judul/ Jml artikel : Penerapan Algoritma k-means untuk Efisiensi Produktivitas  
Tanaman Jate Kedondong pada Produktivitas dan Leas panen  
Di kabupaten Cugeneg

Karya ilmiah yang bersangkutan di atas melalui proses cek plagiatsi menggunakan aplikasi trumtin dengan hasil kemiripan ( Similarity) sebesar 14..... %

Demikian surat keterangan ini dibuat agar dapat dipergunakan sebagaimana mestinya.

Mahasiswa

  
Nayla Dwi Salsabila

Malang, 11, Februari, 2025

Pelaksana,

  
Pelaksana

# PENERAPAN ALGORITMA K-MEANS UNTUK KLASTERISASI PRODUKTIVITAS TANAMAN JAHE BERDASARKAN LUAS PANEN DAN JUMLAH PRODUKSI DI KABUPATEN SUMENEP

## ORIGINALITY REPORT

14%

SIMILARITY INDEX

2%

INTERNET SOURCES

14%

PUBLICATIONS

2%

STUDENT PAPERS

## PRIMARY SOURCES

1

Nayla Salsabila, Karina Aulisari, Hani Zulfia Zahro. "Penerapan Algoritma K-Means Untuk Klasterisasi Produktivitas Tanaman Jahe", Infotek: Jurnal Informatika dan Teknologi, 2025

Publication

13%

2

Submitted to Xavier University

Student Paper

2%

Exclude quotes  Off

Exclude bibliography  On

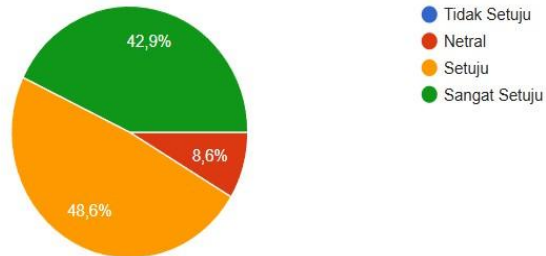
Exclude matches  < 2%

## Penilaian Website Klasterisasi Produktivitas Tanaman Jahe Di Kabupaten Sumenep

Apakah tampilan home untuk user yang ditampilkan sudah informatif?

[Salin diagram](#)

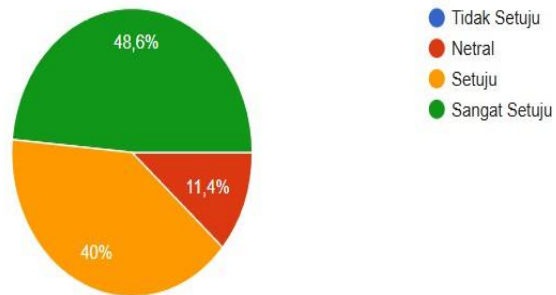
35 jawaban



Apakah data kecamatan yang ditampilkan sudah informatif?

[Salin diagram](#)

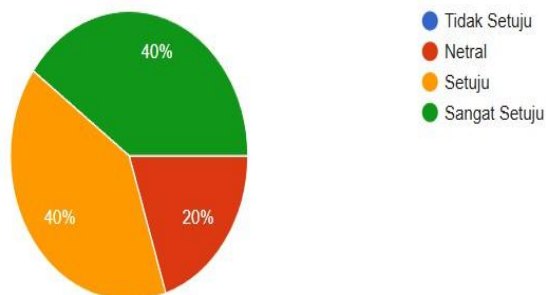
35 jawaban



Apakah tampilan UI (User Interface) aplikasi menarik untuk digunakan?

[Salin diagram](#)

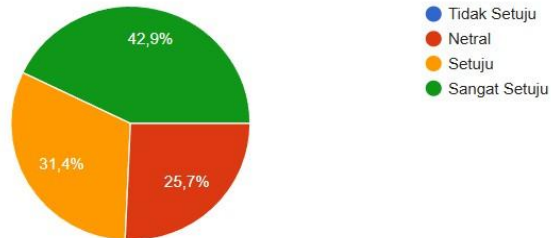
35 jawaban



Apakah menu halaman kecamatan, halaman produksi, halaman luas panen mudah untuk dipahami dan dioperasikan?

[Salin diagram](#)

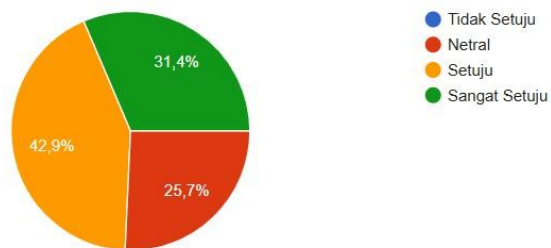
35 jawaban



Apakah tampilan data hasil klasterisasi yang ditampilkan sudah informatif?

[Salin diagram](#)

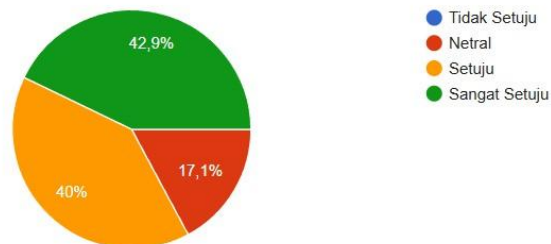
35 jawaban



Apakah hasil dari klasterisasi di aplikasi sudah sesuai dengan yang diharapkan?

[Salin diagram](#)

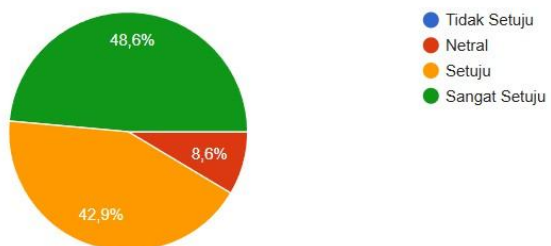
35 jawaban



Apakah aplikasi ini memberikan kemudahan dalam melakukan pengelompokan data produktivitas tanaman jahe di kabupaten sumenep?

[Salin diagram](#)

35 jawaban



### LAMPIRAN DATA

No	Tahun	Kecamatan	Produksi	Luas Lahan
1	2020	Pragaan	715	1175
2	2020	Saronggi	751	955
3	2020	Giligenteng	20	14
4	2020	Talango	28859	12155
5	2020	Kalianget	183	200
6	2020	Kota Sumenep	58	13
7	2020	Batuan	4261	4100
8	2020	Lenteng	6197	10000
9	2020	Ganding	0	0
10	2020	Guluk Guluk	1438	1017
11	2020	Pasongsongan	6099	4250
12	2020	Ambunten	12474	8500
13	2020	Rubaru	573	1030
14	2020	Dasuk	881	1000
15	2020	Manding	2513	3500
16	2020	Batuputih	21	15
17	2020	Gapura	4981	1500
18	2020	Batang Batang	42	11
19	2020	Dungkek	3163	3750
20	2020	Nonggunong	13613	7500
21	2020	Gayam	0	0
22	2020	Raas	28	24
23	2020	Sapeken	658	400
24	2020	Arjasa	10706	14700
25	2020	Kangayan	5128	4530
26	2020	Masalembu	119	106
27	2020	Sumenep	103461	80431
28	2021	Pragaan	519	1150
29	2021	Saronggi	773	975
30	2021	Giligenteng	0	0
31	2021	Talango	32960	14165
32	2021	Kalianget	172	200
33	2021	Kota Sumenep	67	13
34	2021	Batuan	4273	4100
35	2021	Lenteng	6201	10000
36	2021	Ganding	0	0
37	2021	Guluk Guluk	1892	1305
38	2021	Pasongsongan	6342	4340
39	2021	Ambunten	12854	8900
40	2021	Rubaru	581	1030
41	2021	Dasuk	1411	1000
42	2021	Manding	2637	3500

<b>No</b>	<b>Tahun</b>	<b>Kecamatan</b>	<b>Produksi</b>	<b>Luas Lahan</b>
43	2021	Batuputih	22	15
44	2021	Gapura	5104	1600
45	2021	Batang Batang	45	15
46	2021	Dungkek	3871	4000
47	2021	Nonggunong	14093	7500
48	2021	Gayam	1321	1000
49	2021	Raas	14	12
50	2021	Sapeken	711	400
51	2021	Arjasa	10845	14700
52	2021	Kangayan	5203	4540
53	2021	Masalembu	135	108
54	2021	Sumenep	112046	84568
55	2022	Pragaan	530	1175
56	2022	Saronggi	866	975
57	2022	Giligenteng	22	0
58	2022	Talango	38006	16355
59	2022	Kalianget	181	200
60	2022	Kota Sumenep	98	20
61	2022	Batuan	4301	4100
62	2022	Lenteng	6301	10000
63	2022	Ganding	0	0
64	2022	Guluk Guluk	2564	1300
65	2022	Pasongsongan	10452	4420
66	2022	Ambunten	14033	9000
67	2022	Rubaru	597	1030
68	2022	Dasuk	1424	1000
69	2022	Manding	2790	3500
70	2022	Batuputih	31	15
71	2022	Gapura	5682	1500
72	2022	Batang Batang	46	15
73	2022	Dungkek	3921	4000
74	2022	Nonggunong	14408	7500
75	2022	Gayam	1371	1000
76	2022	Raas	18	15
77	2022	Sapeken	773	400
78	2022	Arjasa	10998	14700
79	2022	Kangayan	5512	4600
80	2022	Masalembu	148	110
81	2022	Sumenep	125051	86930
82	2023	Pragaan	535	1175
83	2023	Saronggi	929	1000
84	2023	Giligenteng	20	0
85	2023	Talango	39360	16700
86	2023	Kalianget	171	200
87	2023	Kota Sumenep	98	20

<b>No</b>	<b>Tahun</b>	<b>Kecamatan</b>	<b>Produksi</b>	<b>Luas Lahan</b>
88	2023	Batuan	4404	4200
89	2023	Lenteng	7709	10000
90	2023	Ganding	0	0
91	2023	Guluk Guluk	2799	1500
92	2023	Pasongsongan	11635	5100
93	2023	Ambunten	16184	9000
94	2023	Rubaru	632	1030
95	2023	Dasuk	1459	1000
96	2023	Manding	2888	3500
97	2023	Batuputih	29	15
98	2023	Gapura	5774	1500
99	2023	Batang Batang	46	15
100	2023	Dungkek	3876	4000
101	2023	Nonggunong	14572	7600
102	2023	Gayam	1457	1100
103	2023	Raas	18	15
104	2023	Sapeken	928	550
105	2023	Arjasa	11497	14500
106	2023	Kangayan	5573	4700
107	2023	Masalembu	156	115
108	2023	Sumenep	132729	88535

## LAMPIRAN SOURCE CODE

### 1. Source Code KecamatanController.php

```
<?php
namespace App\Http\Controllers;
use App\Models\TbKecamatan;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Validator;
use RealRashid\SweetAlert\Facades\Alert;
class KecamatanController extends Controller
{
    public function index()
    {
        $data['kecamatan'] = TbKecamatan::all();
        $title = 'Hapus Kecamatan';
        $text = "Apakah anda yakin untuk hapus?";
        confirmDelete($title, $text);
        return view('kecamatan.index', $data);
    }
    public function create() // untuk menampilkan halaman
    {
        return view('kecamatan.create');
    }
    public function store(Request $request)
    {
        $messages = [
            'required' => 'Field :attribute wajib diisi.'
        ];
        $validator = Validator::make($request->all(), [
            'nama_kecamatan' => 'required|string|max:255'
        ], $messages);
        if ($validator->fails()) {
            return redirect()->back()->withInput()->withErrors($validator);
        }
        DB::beginTransaction();
        try {
            $kecamatanData = $request->all();
            TbKecamatan::create($kecamatanData);
            Alert::success("Success", "Data berhasil disimpan");
            DB::commit();

            return redirect("kecamatan");
        } catch (\Exception $e) {
            DB::rollBack();
            return redirect()->back()->withInput()->with('error', 'Terjadi kesalahan saat menyimpan data. ' . $e->getMessage());
        }
    }
    public function edit(TbKecamatan $kecamatan)
    {
        return view('kecamatan.edit', compact('kecamatan'));
    }
    public function update(Request $request, TbKecamatan $kecamatan)
```



```

    {
        $messages = [
            'required' => 'Field :attribute wajib diisi.'
        ];
        $validator = Validator::make($request->all(), [
            'nama_kecamatan' => 'required|string|max:255'
        ], $messages);
        if ($validator->fails()) {
            return redirect()->back()->withInput()-
>withErrors($validator);
        }

        DB::beginTransaction();
        try {
            $kecamatanData = $request->all();
            $kecamatan->update($kecamatanData
                Alert::success("Success", "Data berhasil
disimpan"));
            DB::commit();
            return redirect("kecamatan");
        } catch (\Exception $e) {
            DB::rollBack();
            return redirect()->back()->withInput()-
>with('error', 'Terjadi kesalahan saat menyimpan data. ' .
$e->getMessage());
        }
    }
    public function destroy(TbKecamatan $kecamatan)
    {
        $kecamatan->delete();
        Alert::success('Success', 'Kecamatan berhasil
dihapus');
        return redirect()->route('kecamatan.index');
    }
}

```

## 2. Source Code ProduksiController.php

```

<?php
namespace App\Http\Controllers;
use App\Imports\ProduksiImport;
use App\Models\TbProduksi;
use App\Models\TbKecamatan;
use Exception;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Validator;
use Maatwebsite\Excel\Facades\Excel;
use RealRashid\SweetAlert\Facades\Alert;
class ProduksiController extends Controller
{
    public function index()
    {
        $data['produksi'] = TbProduksi::all();
        $title = 'Hapus Produksi';
        $text = "Apakah anda yakin untuk hapus?";
        confirmDelete($title, $text);
        return view('produksi.index', $data);
    }
}

```

```

}
public function import(Request $request)
{
    $request->validate([
        'file' => 'required|mimes:xls,xlsx',
    ]);
    try {
        Excel::import(new ProduksiImport, $request-
>file('file'));
        Alert::success("Success", "Data berhasil
diimpor!");
    } catch (Exception $e) {
        Alert::error('Error', 'Terjadi kesalahan: ' . $e-
>getMessage());
        return redirect()->back();
    }
    return redirect()->back();
}
public function create()
{
    $data['kecamatan'] = TbKecamatan::all();
    return view('produksi.create', $data);
}
public function store(Request $request)
{
    $messages = [
        'required' => 'Field :attribute wajib diisi.'
    ];
    $validator = Validator::make($request->all(), [
        'jumlah_produksi' => 'required|string|max:255'
    ], $messages);
    if ($validator->fails()) {
        return redirect()->back()->withInput()-
>withErrors($validator);
    }
    DB::beginTransaction(); // fungsi database
    try {
        $produksiData = $request->all();
        $existingData = TbProduksi::where('tahun',
$produksiData['tahun'])
->where('id_kecamatan',
$produksiData['id_kecamatan'])
->first();
        if ($existingData) {
            Alert::error("Error", "Data sudah ada");
            return redirect()->back();
        }
        TbProduksi::create($produksiData);
        Alert::success("Success", "Data berhasil
disimpan");
        DB::commit();
        return redirect("produksi");
    } catch (\Exception $e) {
        DB::rollBack();
        return redirect()->back()->withInput()-
>with('error', 'Terjadi kesalahan saat menyimpan data. ' . $e-
>getMessage());
    }
}
}

```

```

public function edit(TbProduksi $produksi)
{
    $kecamatan = TbKecamatan::all();
    return view('produksi.edit', compact('produksi',
'kecamatan'));
}
public function update(Request $request, TbProduksi
$produksi)
{
    $messages = [
        'required' => 'Field :attribute wajib diisi.',
    ];
    $validator = Validator::make($request->all(), [
        'jumlah_produk' => 'required|string|max:255'
    ], $messages);
    if ($validator->fails()) {
        return redirect()->back()->withInput()-
>withErrors($validator);
    }
    DB::beginTransaction();
    try {
        $produksiData = $request->all();
        $produksi->update($produksiData);
        Alert::success("Success", "Data berhasil
disimpan");
        DB::commit();
        return redirect("produksi");
    } catch (\Exception $e) {
        DB::rollBack();
        return redirect()->back()->withInput()-
>with('error', 'Terjadi kesalahan saat menyimpan data. ' . $e-
>getMessage());
    }
}
public function destroy(TbProduksi $produksi)
{
    $produksi->delete();
    Alert::success('Success', 'Produksi berhasil
dihapus');
    return redirect()->route('produksi.index');
}
}

```

### 3. Source Code LuasDaerahPanenController.php

```

<?php
namespace App\Http\Controllers;
use App\Imports\LuasDaerahPanenImport;
use App\Models\TbLuasDaerahPanen;
use App\Models\TbProduksi;
use App\Models\TbKecamatan;
use Exception;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Validator;
use Maatwebsite\Excel\Facades\Excel;
use RealRashid\SweetAlert\Facades\Alert;
class LuasDaerahPanenController extends Controller

```

```

{
    public function index()
    {
        $data['luas_daerah_panen'] =
TbLuasDaerahPanen::all();
        $title = 'Hapus Luas Panen';
        $text = "Apakah anda yakin untuk hapus?";
        confirmDelete($title, $text);
        return view('luas_daerah_panen.index', $data);
    }
    public function import(Request $request)
    {
        $request->validate([
            'file' => 'required|mimes:xls,xlsx',
        ]);
        try {
            Excel::import(new LuasDaerahPanenImport,
$request->file('file'));
            Alert::success("Success", "Data berhasil
diimpor!");
        } catch (Exception $e) {
            Alert::error('Error', 'Terjadi kesalahan: ' . $e-
>getMessage());
            return redirect()->back();
        }
        return redirect()->back();
    }
    public function create()
    {
        $data['kecamatan'] = TbKecamatan::all();
        return view('luas_daerah_panen.create', $data);
    }
    public function store(Request $request)
    {
        $messages = [
            'required' => 'Field :attribute wajib diisi.'
        ];
        $validator = Validator::make($request->all(), [
            'luas_daerah' => 'required|string|max:255'
        ], $messages);
        if ($validator->fails()) {
            return redirect()->back()->withInput()-
>withErrors($validator);
        }
        DB::beginTransaction();
        try {
            $luasdaerahpanenData = $request->all();
            $existingData = TbProduksi::where('tahun',
$luasdaerahpanenData['tahun'])
                ->where('id_kecamatan',
$luasdaerahpanenData['id_kecamatan'])
                ->first();
            if ($existingData) {
                Alert::error("Error", "Data sudah ada");
                return redirect()->back();
            }
            TbLuasDaerahPanen::create($luasdaerahpanenData);
            Alert::success("Success", "Data berhasil
disimpan");
        }
    }
}

```

```

        DB::commit();
        return redirect("luasdaerahpanen");
    } catch (\Exception $e) {
        DB::rollBack();
        return redirect()->back()->withInput()-
>with('error', 'Terjadi kesalahan saat menyimpan data. ' .
    $e->getMessage());
    }
}
public function edit(TbLuasDaerahPanen $luasdaerahpanen)
{
    $kecamatan = TbKecamatan::all();
    return view('luas_daerah_panen.edit',
compact('luasdaerahpanen', 'kecamatan'));
}
public function update(Request $request,
TbLuasDaerahPanen $luasdaerahpanen)
{
    $messages = [
        'required' => 'Field :attribute wajib diisi.'
    ];
    $validator = Validator::make($request->all(), [
        'luas_daerah' => 'required|string|max:255'
    ], $messages);
    if ($validator->fails()) {
        return redirect()->back()->withInput()-
>withErrors($validator);
    }
    DB::beginTransaction();
    try {
        $luasdaerahpanenData = $request->all();
        $luasdaerahpanen->update($luasdaerahpanenData);
        Alert::success("Success", "Data berhasil
disimpan");
        DB::commit();
        return redirect("luasdaerahpanen");
    } catch (\Exception $e) {
        DB::rollBack();
        return redirect()->back()->withInput()-
>with('error', 'Terjadi kesalahan saat menyimpan data. ' .
    $e->getMessage());
    }
}
public function destroy(TbLuasDaerahPanen
$luasdaerahpanen)
{
    $luasdaerahpanen->delete();
    Alert::success('Success', 'Luas Panen berhasil
dihapus');
    return redirect()->route('luasdaerahpanen.index');
}
}
}

```

#### 4. Source Code KlasterisasiController.php

```

<?php
namespace App\Http\Controllers;
use App\Models\TbClustering;

```

```

use App\Models\TbKecamatan;
use App\Models\TbLuasDaerahPanen;
use App\Models\TbSilhouetteScore;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Facades\Validator;
use RealRashid\SweetAlert\Facades\Alert;
use function PHPUnit\Framework\isNull;
class KlasterisasiController extends Controller
{
    public function index(Request $request)
    {
        $tahunList = TbLuasDaerahPanen::select('tahun')->distinct()->pluck('tahun');
        $selectedYear = $request->get('tahun');
        if ($selectedYear) {
            $silhouetteScore = TbSilhouetteScore::where('tahun', '=', $selectedYear)->first();
            $data = TbClustering::where('tahun', $selectedYear)->get();
            if ($data->isEmpty()) {
                $this->prosesKlasterisasi($selectedYear);
                $data = TbClustering::where('tahun', $selectedYear)->get();
                Alert::success('Success', 'Proses klasterisasi untuk tahun ' . $selectedYear . ' selesai');
            } else {
                Alert::success('Success', 'Menampilkan hasil klasterisasi untuk tahun ' . $selectedYear);
            }
        } else {
            $data = collect();
        }
        return view('clustering.index', compact('tahunList', 'data', 'selectedYear', 'silhouetteScore'));
    }
    public function prosesKlasterisasi($tahun)
    {
        $dataJahe = TbKecamatan::selectRaw('
            tb_kecamatan.id AS id_kecamatan,
            tb_kecamatan.nama_kecamatan,
            tb_luas_daerah_panen.tahun,
            SUM(tb_luas_daerah_panen.luas_daerah) AS
total_luas_daerah,
            SUM(tb_produk.jumlah_produk) AS
total_jumlah_produk
        ')
        ->join('tb_luas_daerah_panen', function ($join) {
            $join->on('tb_luas_daerah_panen.id_kecamatan', '=', 'tb_kecamatan.id');
        })
        ->join('tb_produk', function ($join) {
            $join->on('tb_produk.id_kecamatan', '=', 'tb_kecamatan.id')
                ->on('tb_produk.tahun', '=', 'tb_luas_daerah_panen.tahun');
        });
    }
}

```

```

    })
    ->where('tb_luas_daerah_panen.tahun', $tahun)
    ->groupBy('tb_kecamatan.id',
'tb_kecamatan.nama_kecamatan', 'tb_luas_daerah_panen.tahun')
    ->get();
    $data = [];
    foreach ($dataJahe as $item) {
        $data[] = [
            'id_kecamatan' => $item->id_kecamatan,
            'nama_kecamatan' => $item->nama_kecamatan,
            'tahun' => $item->tahun,
            'total_luas_daerah' => $item-
>total_luas_daerah,
            'total_jumlah_produksi' => $item-
>total_jumlah_produksi
        ];
    }
    $currentHash = md5(json_encode($data));
    $lastHash = TbClustering::where('tahun', $tahun)-
>orderBy('created_at', 'desc')->value('data_hash');

    if ($currentHash === $lastHash) {
        Alert::error('Peringatan', 'Tidak ada perubahan
data, proses klasterisasi tidak dilakukan.');
```

```

        return redirect('clustering');
    }
    $jumlahCluster = 2;
    $clusters = $this->kMeans($data, $jumlahCluster,
$tahun);
    $centroids = $this->initCentroids($data,
$jumlahCluster);
    TbClustering::where('tahun', $tahun)->delete();
    $this->insertHasilKlasterisasi($clusters,
$currentHash);
    Alert::success('Success', 'Klasterisasi untuk tahun '
. $tahun . ' Berhasil');
```

```

    return redirect('clustering');
}
private function kMeans($data, $k, $tahun)
{
    $centroids = $this->initCentroids($data, $k);
    $iterations = 100;
    $iterationData = [];
    for ($i = 0; $i < $iterations; $i++) {
        $clusters = $this->assignClusters($data,
$centroids);
        $newCentroids = $this->updateCentroids($data,
$clusters, $k);
        $iterationData[] = [
            'iteration' => $i + 1,
            'centroids' => $this-
>labelCentroids($centroids),
            'clusters' => $clusters
        ];
        $this->insertIterationData($i + 1, $this-
>labelCentroids($centroids), $clusters);
        if ($centroids == $newCentroids) {
            break;
        }
    }
}

```

```

        $centroids = $newCentroids;
    }
    ksort($clusters);
    $averageSilhouetteScore = $this->
>calculateSilhouetteScore($data, $clusters, $centroids);
    $silhouetteScore = TbSilhouetteScore::where('tahun',
'=', $tahun)->first();
    if (!$silhouetteScore) {
        DB::table('tb_silhouette_score')->insert([
            'avg_silhouette_score' =>
$averageSilhouetteScore,
            'tahun' => $tahun,
        ]);
    }
    return $clusters;
}
}
private function calculateSilhouetteScore($data, $clusters,
$centroids)
{
    $calculateEuclideanDistance = function ($point1, $point2)
    {
        return sqrt(pow($point1['total_luas_daerah'] -
$point2['total_luas_daerah'], 2) +
            pow($point1['total_jumlah_produksi'] -
$point2['total_jumlah_produksi'], 2));
    };
    $distanceMatrix = [];
    foreach ($data as $i => $point1) {
        foreach ($data as $j => $point2) {
            if (!isset($distanceMatrix[$i])) {
                $distanceMatrix[$i] = [];
            }
            $distanceMatrix[$i][$j] =
$calculateEuclideanDistance($point1, $point2);
        }
    }
    $silhouetteScores = [];
    foreach ($data as $i => $point) {
        $clusterId = $this->getClosestCluster($point,
$centroids);
        $sameClusterPoints = $clusters[$clusterId];
        $a = 0;
        $numSameClusterPoints = count($sameClusterPoints);
        if ($numSameClusterPoints > 1) {
            for ($j = 0; $j < $numSameClusterPoints; $j++) {
                if ($sameClusterPoints[$j] !== $point) {
                    $samePointIndex =
array_search($sameClusterPoints[$j], $data);
                    $a +=
$distanceMatrix[$i][$samePointIndex];
                }
            }
            $a /= ($numSameClusterPoints - 1);
        } else {
            $a = 0;
        }
        $b = PHP_INT_MAX;
        foreach ($centroids as $index => $centroid) {

```



```

        if ($index !== $clusterId) {
            $otherClusterPoints = $clusters[$index];
            $bClusterDistance = 0;
            $numOtherClusterPoints =
count($otherClusterPoints);
            foreach ($otherClusterPoints as $otherPoint) {
                $otherPointIndex =
array_search($otherPoint, $data);
                $bClusterDistance +=
$distanceMatrix[$i][$otherPointIndex];
            }
            $bClusterDistance /= $numOtherClusterPoints;
            if ($bClusterDistance < $b) {
                $b = $bClusterDistance;
            }
        }
    }
    if ($a === 0 && $b === PHP_INT_MAX) {
        $silhouetteScore = 0;
    } else {
        $silhouetteScore = ($b - $a) / max($a, $b);
    }
    $silhouetteScores[] = $silhouetteScore;
}
$averageSilhouetteScore = array_sum($silhouetteScores) /
count($silhouetteScores);
return $averageSilhouetteScore;
}
private function getClosestCluster($point, $centroids)
{
    $minDistance = PHP_INT_MAX;
    $closestClusterId = null;
    foreach ($centroids as $index => $centroid) {
        $distance = sqrt(pow($point['total_luas_daerah'] -
$centroid['total_luas_daerah'], 2) +
        pow($point['total_jumlah_produksi'] -
$centroid['total_jumlah_produksi'], 2));
        if ($distance < $minDistance) {
            $minDistance = $distance;
            $closestClusterId = $index;
        }
    }
    return $closestClusterId;
}
private function labelCentroids($centroids)
{
    $labeledCentroids = [];
    foreach ($centroids as $index => $centroid) {
        $label = '';

        switch ($index) {
            case 0:
                $label = 'C1';
                break;
            case 1:
                $label = 'C2';
                break;
        }
    }
    $labeledCentroids[] = [

```

```

        'label' => $label,
        'centroid' => $centroid
    ];
}
return $labeledCentroids;
}
private function insertHasilKlasterisasi($clusters,
$currentHash)
{
    foreach ($clusters as $index => $clusterData) {
        $clusterLabel = '';

        switch ($index) {
            case 0:
                $clusterLabel = 'C1';
                break;
            case 1:
                $clusterLabel = 'C2';
                break;
        }
        foreach ($clusterData as $data) {
            DB::table('tb_clustering')->insert([
                'id_kecamatan' => $data['id_kecamatan'],
                'tahun' => $data['tahun'],
                'luas_daerah' => $data['total_luas_daerah'],
                'jumlah_produksi' =>
                $data['total_jumlah_produksi'],
                'cluster' => $clusterLabel,
                'created_at' => now(),
                'data_hash' => $currentHash,
            ]);
        }
    }
}
private function initCentroids($data, $k)
{
    {
        usort($data, function ($a, $b) {
            $sumA = $a['total_luas_daerah'] +
            $a['total_jumlah_produksi'];
            $sumB = $b['total_luas_daerah'] +
            $b['total_jumlah_produksi'];
            return $sumA <=> $sumB;
        });
        $centroids = [];
        $centroids[] = [
            'nama_kecamatan' => $data[0]['nama_kecamatan'],
            'total_luas_daerah' => $data[0]['total_luas_daerah'],
            'total_jumlah_produksi' =>
            $data[0]['total_jumlah_produksi']
        ];
        $lastIndex = count($data) - 1;
        $centroids[] = [
            'nama_kecamatan' =>
            $data[$lastIndex]['nama_kecamatan'],
            'total_luas_daerah' =>
            $data[$lastIndex]['total_luas_daerah'],
            'total_jumlah_produksi' =>
            $data[$lastIndex]['total_jumlah_produksi']
        ];
    }
}

```

```

    return $centroids;
}
private function assignClusters($data, $centroids)
{
    $clusters = [];
    foreach ($data as $point) {
        $minDistance = PHP_INT_MAX;
        $closestCentroid = null;
        foreach ($centroids as $index => $centroid) {
            $distance = sqrt(
                pow($point['total_luas_daerah'] -
$centroid['total_luas_daerah'], 2) +
                pow($point['total_jumlah_produksi'] -
$centroid['total_jumlah_produksi'], 2)
            );
            if ($distance < $minDistance) {
                $minDistance = $distance;
                $closestCentroid = $index;
            }
        }
        $clusters[$closestCentroid][] = $point;
    }
    return $clusters;
}
private function updateCentroids($data, $clusters, $k)
{
    $newCentroids = [];
    for ($i = 0; $i < $k; $i++) {
        if (isset($clusters[$i]) && count($clusters[$i]) > 0) {
            $luasTotal =
array_sum(array_column($clusters[$i], 'total_luas_daerah'));
            $produksiTotal =
array_sum(array_column($clusters[$i],
'total_jumlah_produksi'));
            $clusterSize = count($clusters[$i]);
            $newCentroids[] = [
                'total_luas_daerah' => $luasTotal /
$clusterSize,
                'total_jumlah_produksi' => $produksiTotal /
$clusterSize
            ];
        } else {
            $newCentroids[] = $this->initCentroids($data,
1)[0];
        }
    }
    return $newCentroids;
}
private function insertIterationData($iteration, $centroids,
$clusters)
{
    $labeledCentroids = [];
    foreach ($centroids as $index => $centroid) {
        $label = '';
        switch ($index) {
            case 0:
                $label = 'C1';
                break;
            case 1:

```

```
        $label = 'C2';
        break;
    }
    $labeledCentroids[] = [
        'label' => $label,
        'centroid' => $centroid,
    ];
}
$centroidData = json_encode($labeledCentroids);
$clusterData = json_encode($clusters);
DB::table('tb_log_iterasi')->insert([
    'iterasi' => $iteration,
    'centroid_data' => $centroidData,
    'cluster_data' => $clusterData,
    'created_at' => now(),
]);
}
public function getTahun()
{
    $tahun = DB::table('tb_clustering')->orderBy('tahun',
'asc')->distinct()->pluck('tahun');
    return response()->json(['tahun' => $tahun]);
}
}
```