

# LAMPIRAN



PT BNI (PERSERO) MALANG  
BANK MIGA 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 : J. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : J. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

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

**Nama : Madelbertha Fridolin Puka**  
**Nim : 2118132**  
**Jurusan : Teknik Informatika S-1**  
**Judul : Penerapan Data Mining Untuk Clustering Wilayah Produksi**  
**Kakao Menggunakan Metode K-Means**  
**(Studi Kasus: Kabupaten Sikka, Nusa Tenggara Timur)**

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

**Hari : Rabu**  
**Tanggal : 22 Januari 2025**  
**Nilai : 87**

**Panitia Ujian Skripsi :**  
**Ketua Majelis Penguji**

**Yosep Agus Pranoto ST., MT**  
**NIP.P 1031000432**

**Dosen Penguji I**

**Ahmad Faisol, ST., MT**  
**NIP. P. 1031000431**

**Anggota Penguji :**

**Dosen Penguji II**

**Eko Heri Susanto, S.Kom., M.Kom**  
**NIP.P 1032400605**



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

**FORMULIR PERBAIKAN SKRIPSI**

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

NAMA : Madelbertha Fridolin Puka  
NIM : 2118132  
JURUSAN : Teknik Informatika S-1  
JUDUL : Penerapan Data Mining Untuk Clustering Wilayah Produksi Kakao  
Menggunakan Metode K-Means  
(Studi Kasus: Kabupaten Sikka, Nusa Tenggara Timur)

No	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	22 Januari 2025	1. Perbaiki resolusi gambar yang kurang jelas 2. Perbaiki kesimpulan 3. Lakukan pengujian optimasi cluster	
2.	Penguji II	22 Januari 2025	1. Pada latar belakang seharusnya dijelaskan kelebihan metode K-Means dibandingkan dengan metode clustering lainnya 2. Data harus divalidasi lagi	

**Anggota Penguji:**

**Dosen Penguji I**

Ahmad Faisol, ST., MT  
NIP. P. 1031000431

**Dosen Penguji II**

Eko Heri Susanto, S.Kom., M.Kom  
NIP.P 1032400605

**Mengetahui:**

**Dosen Pembimbing I**

(Joseph Dedy Irawan, ST., MT)  
NIP. 197404162005 011002

**Dosen Pembimbing II**

(Nurlaily Vendvansyah, ST., MT)  
NIP:1031200557



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 : J. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Huntri), Fax. (0341) 553015 Malang 65145  
Kampus II : J. 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 Joseph Dedy Irawan ST., MT.**  
Dosen Program Studi Teknik Informatika S-1  
Institut Teknologi Nasional  
Malang

Dengan Hormat,  
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : Madelbertha Fridolin Puka  
Nim : 2118132  
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.**  
NIP.P. 1031000432

Form S-4a



PT. BNI (PERSERO) MALANG  
BANK NAGA 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 : J. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : J. 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 Nurlaily Vendyansyah ST., MT.**  
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 : Madelbertha Fridolin Puka  
Nim : 2118132  
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.


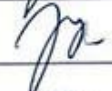



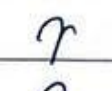

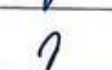
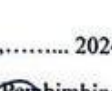
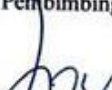
Mengetahui  
Program Studi Teknik Informatika S-1  
Ketua,

**Yosep Agus Pranoto, ST., MT.**  
NIP.P. 1031000432



## FORMULIR BIMBINGAN SKRIPSI

Nama : Masduki Firdaus Puka  
 Nim : 2118132  
 Masa Bimbingan : 21 Agustus 2024 - 21 Februari 2025  
 Judul Skripsi : Penerapan Data Mining Untuk Clustering Website Produk Kakao

No.	Tanggal	Uraian	Paraf Pembimbing
1.	19-10-2024	MEMAHAMI KEBERINTUNGAN MANUAL EXCEL & PYTHON	
2.	28-10-2024	Pengisian CRUD Produk dan Lokasi. Melanjutkan menu Clustering	
3.	01-11-2024	Melengkap metode k-Means	
4.	26-11-2024	Memperbaiki Perhitungan dan website	
5.	29-11-2024	Konsultasi Jurnal	
6.	29-11-2024	Konsultasi Jurnal + Aplikasi	
7.	29-11-2024	NU KEMAH	
8.	09-01-2025	Revisi perhitungan manual per tahun	
9.	14-01-2025	Revisi website, perbaikan Revisi laporan	
10.	15-01-2025	ACC Kompra	

Malang,..... 2024


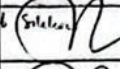





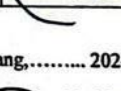


Dosen Pembimbing



Joseph Dedy Wawan, ST.MT

## FORMULIR BIMBINGAN SKRIPSI

**Nama** : Madelbertha Erickson Purno  
**Nim** : 2118032  
**Masa Bimbingan** : 21 Agustus 2024 - 21 Februari 2025  
**Judul Skripsi** : Penerapan Data Mining Untuk Clustering dengan Produksi Kacang

No.	Tanggal	Uraian	Paraf Pembimbing
1.	16/10/24	Uraian Pembimbing : Struktur, Data, Perhitungan, dan sifat-sifat kuantitatif; Data; Perhitungan; dan -keperluan; Sistematisasi Penulisan;	
2.	20/10/24	Tesis : 10 tahun terakhir; Profil Kesmi; dan Floriatur KMeans & K-Means;	
3.	3/11/24	Rencana-rencana metode belum ada; Perencanaan Analisis kebutuhan; Floriatur KMeans.	
4.	26/11/2024	Konsultasi Jurnal	
5.	29/11/2024	Flowchart sistem	
6.	29/11/2024	Perbandingan hasil sistem dengan hasil pakar	
7.	23/11/2024	ACC Laporan Samas	
8.	09/01/2025	Konsultasi revisi Samas	
9.	14/01/2025	Perbaikan revisi laporan	
10.	15/01/2025	ACC Laporan	

Malang, ..... 2024

Dosen Pembimbing

(Nurhidayah Vandyansyah, ST., MT)



**BADAN PUSAT STATISTIK KABUPATEN SIKKA**

**SURAT PERNYATAAN**

Yang bertandatangan di bawah ini :

Nama : Maryo Yoseph Ambarto Dwi Sili Osan  
Jabatan : Fungsional Statistika Pelaksana  
Instansi/Lembaga : BPS Kabupaten Sikka  
No. Telepon/HP : 081239263146  
Alamat : Jalan Waiklau, Kelurahan Madawat, Kecamatan Alok

Dengan ini menyatakan bersedia menjadi mitra dalam pembuatan skripsi dari

Nama : Madelbertha Fridolin Puka  
NIM : 2118132  
Judul Skripsi : Penerapan Data Mining Untuk Clustering Wilayah Produksi Kakao Menggunakan Metode K-Means (Studi Kasus: Kabupaten Sikka, Nusa Tenggara Timur)

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 kakao 2011-2022

Maumere, 13 Desember 2024  
Yang membuat pernyataan



(Maryo Yoseph Ambarto Dwi Sili Osan)





**INSTITUT TEKNOLOGI NASIONAL MALANG**  
**PERPUSTAKAAN PUSAT**  
Jln. Bendungan Sigura-gura No 2 Malang 65145  
Telp. (0341) 551431 Pns. 163-146-147 Fax. (0341) 553015 Website: library.itn.ac.id

### FORM UJI PLAGIASI UNTUK MAHASISWA

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

Nama : Modelbartha F. Puka  
NIM : 218132  
Fakultas / Jurusan : Teknologi Industri / Teknik Informatika S-1  
Email : modelbartha.f.puka@gmail.com  
No. Tlp : 08121780916  
Judul/ Jml artikel : Penerapan Data Mining Untuk Clustering Wilayah  
Produksi Kakao Menggunakan metode K-means  
(Studi kasus : Kabupaten Sukka, Nusa Tenggara Timur)

Karya ilmiah yang bersangkutan di atas melalui proses cek plagiatsi menggunakan aplikasi trunitin dengan hasil kemiripan ( Similarity) Sebesar....15..... %  
Demikian surat keterangan ini dibuat agar dapat dipergunakan sebagaimana mestinya.

Mahasiswa

  
Modelbartha F. Puka

Malang, 11 Februari 2025  
Pelaksana,

  
Pelaksana

SKRIPSI PENERAPAN DATA MINING UNTUK CLUSTERING  
WILAYAH PRODUKSI KAKAO MENGGUNAKAN METODE K-  
MEANS (STUDI KASUS: KABUPATEN SIKKA, NUSA TENGGARA  
TIMUR)

ORIGINALITY REPORT



PRIMARY SOURCES



Exclude quotes Off

Exclude matches < 2%

Exclude bibliography On

## LAMPIRAN DATA

No	Kecamatan	Tahun	Produksi	Luas Lahan
1	Paga	2011	173	882
2	Mego	2011	1354	259
3	Tanawawo	2011	183	1032
4	Lela	2011	474	1216
5	Bola	2011	412	1651
6	Doreng	2011	400	1649
7	Mapitara	2011	312	930
8	Talibura	2011	562	1263
9	Waigete	2011	248	1089
10	Waiblama	2011	230	536
11	Kewapante	2011	323	849
12	Hewokloang	2011	875	2664
13	Kangae	2011	325	683
14	Palue	2011	230	422
15	Koting	2011	225	815
16	Nelle	2011	350	936
17	Nita	2011	730	2478
18	Magepanda	2011	91	619
19	Alok	2011	2	177
20	Alok Barat	2011	5	386
21	Alok Timur	2011	0	0
22	Paga	2012	26	918
23	Mego	2012	77	4963,2
24	Tanawawo	2012	18,3	103,2
25	Lela	2012	25	394,25
26	Bola	2012	199	1171,95
27	Doreng	2012	65	568,75
28	Mapitara	2012	109	35,44
29	Talibura	2012	55	96,2
30	Waigete	2012	636	422,9
31	Waiblama	2012	236	36,8
32	Kewapante	2012	0,4	31,85
33	Hewokloang	2012	158	577,5
34	Kangae	2012	18	5502,79
35	Palue	2012	1,1	4,47
36	Koting	2012	5	113
37	Nelle	2012	0,35	0,93
38	Nita	2012	20	1188,93
39	Magepanda	2012	0,2	6,45
40	Alok	2012	0	0
41	Alok Barat	2012	3,9	0,113
42	Alok Timur	2012	750	2,7
43	Paga	2013	422	981
44	Mego	2013	266	1462

45	Tanawawo	2013	211	1032
46	Lela	2013	474	1214
47	Bola	2013	479	1642
48	Doreng	2013	405	1649
49	Mapitara	2013	313	930
50	Talibura	2013	685	1260
51	Waigete	2013	308	1081
52	Waiblama	2013	232	536
53	Kewapante	2013	326	852
54	Hewokloang	2013	999	2664
55	Kangae	2013	325	1183
56	Palue	2013	126	396
57	Koting	2013	225	815
58	Nelle	2013	353	963
59	Nita	2013	855	2478
60	Magepanda	2013	102	619
61	Alok	2013	3	177
62	Alok Barat	2013	8	388
63	Alok Timur	2013	1	5
64	Paga	2014	318	923
65	Mego	2014	427	1462
66	Tanawawo	2014	350	1052
67	Lela	2014	371	1232
68	Bola	2014	513	1636
69	Doreng	2014	683	1699
70	Mapitara	2014	338	930
71	Talibura	2014	354	1260
72	Waigete	2014	341	1087
73	Waiblama	2014	159	536
74	Kewapante	2014	365	847
75	Hewokloang	2014	806	2666
76	Kangae	2014	254	1183
77	Palue	2014	95	396
78	Koting	2014	208	855
79	Nelle	2014	438	936
80	Nita	2014	1413	2528
81	Magepanda	2014	233	616
82	Alok	2014	10	188
83	Alok Barat	2014	15	386
84	Alok Timur	2014	4	28
85	Paga	2015	471	898
86	Mego	2015	630	1434
87	Tanawawo	2015	380	1115
88	Lela	2015	459	1217
89	Bola	2015	571	1606
90	Doreng	2015	758	1799



91	Mapitara	2015	384	918
92	Talibura	2015	350	1232
93	Waigete	2015	406	1086
94	Waiblama	2015	139	532
95	Kewapante	2015	361	836
96	Hewokloang	2015	1147	2636
97	Kangae	2015	271	1172
98	Palue	2015	90	386
99	Koting	2015	236	796
100	Nelle	2015	429	925
101	Nita	2015	1327	2599
102	Magepanda	2015	218	606
103	Alok	2015	8	188
104	Alok Barat	2015	13	388
105	Alok Timur	2015	4	28
106	Paga	2016	417	898
107	Mego	2016	380	1115
108	Tanawawo	2016	640	1434
109	Lela	2016	466	1217
110	Bola	2016	596	1606
111	Doreng	2016	766	1819
112	Mapitara	2016	384	918
113	Talibura	2016	360	1252
114	Waigete	2016	139	542
115	Waiblama	2016	416	1086
116	Kewapante	2016	318	846
117	Hewokloang	2016	1149	2646
118	Kangae	2016	271	1172
119	Palue	2016	433	925
120	Koting	2016	250	796
121	Nelle	2016	90	386
122	Nita	2016	1367	2599
123	Magepanda	2016	218	606
124	Alok	2016	9	188
125	Alok Barat	2016	13	388
126	Alok Timur	2016	7	28
127	Paga	2017	475	960
128	Mego	2017	643	1462
129	Tanawawo	2017	380	1115
130	Lela	2017	469	1237
131	Bola	2017	608	1623
132	Doreng	2017	766	1623
133	Mapitara	2017	384	968
134	Talibura	2017	364	1356
135	Waigete	2017	421	1105
136	Waiblama	2017	141	633

137	Kewapante	2017	387	835
138	Hewokloang	2017	1149	2688
139	Kangae	2017	271	1178
140	Palue	2017	91	386
141	Koting	2017	253	748
142	Nelle	2017	435,8	925
143	Nita	2017	1379	2628
144	Magepanda	2017	218	606
145	Alok	2017	9,5	188
146	Alok Barat	2017	13	388
147	Alok Timur	2017	7	24
148	Paga	2018	2,07	1187
149	Mego	2018	4,2	1731
150	Tanawawo	2018	3,8	1235
151	Lela	2018	1,22	675
152	Bola	2018	2,3	1665
153	Doreng	2018	2,7	1445
154	Mapitara	2018	0,5	968
155	Talibura	2018	5	1031
156	Waigete	2018	0,3	1110
157	Waiblama	2018	4	463
158	Kewapante	2018	5,12	860
159	Hewokloang	2018	3,6	2688
160	Kangae	2018	1,66	1172
161	Palue	2018	0,34	461
162	Koting	2018	4,78	768
163	Nelle	2018	3,68	925
164	Nita	2018	2,1	2638
165	Magepanda	2018	1,12	606
166	Alok	2018	0,13	186
167	Alok Barat	2018	0,4	386
168	Alok Timur	2018	0,2	32
169	Paga	2019	5	79
170	Mego	2019	4,6	108
171	Tanawawo	2019	13	107
172	Lela	2019	1	126
173	Bola	2019	10	47
174	Doreng	2019	23	65
175	Mapitara	2019	27	53
176	Talibura	2019	0,4	135
177	Waigete	2019	5	136
178	Waiblama	2019	38	80
179	Kewapante	2019	5	13
180	Hewokloang	2019	4	120
181	Kangae	2019	5,7	22
182	Palue	2019	2,1	1

183	Koting	2019	6,7	76
184	Nelle	2019	8,98	30
185	Nita	2019	5,7	289
186	Magepanda	2019	6,7	34
187	Alok	2019	0,12	186
188	Alok Barat	2019	0,4	461
189	Alok Timur	2019	0,5	22
190	Paga	2020	1,25	79
191	Mego	2020	0,95	108
192	Tanawawo	2020	0,13	107
193	Lela	2020	0,28	126
194	Bola	2020	1,03	47
195	Doreng	2020	0,73	65
196	Mapitara	2020	0,21	53
197	Talibura	2020	0,74	135
198	Waigete	2020	0,21	136
199	Waiblama	2020	1,12	80
200	Kewapante	2020	2,5	13
201	Hewokloang	2020	3,2	120
202	Kangae	2020	3,3	22
203	Palue	2020	0,2	1
204	Koting	2020	1,32	76
205	Nelle	2020	3,5	30
206	Nita	2020	4,4	289
207	Magepanda	2020	2,13	34
208	Alok	2020	0,4	104
209	Alok Barat	2020	0,4	235
210	Alok Timur	2020	0,5	102
211	Paga	2021	472,2	1191
212	Mego	2021	540,7	1712
213	Tanawawo	2021	380,3	1463
214	Lela	2021	122	675
215	Bola	2021	610	1659
216	Doreng	2021	370,5	1439
217	Mapitara	2021	385	963
218	Talibura	2021	74	1096
219	Waigete	2021	201	1103
220	Waiblama	2021	26	463
221	Kewapante	2021	289	864
222	Hewokloang	2021	1160,8	2688
223	Kangae	2021	217	1204
224	Palue	2021	120,5	451
225	Koting	2021	214,7	761
226	Nelle	2021	437	925
227	Nita	2021	2136	2630
228	Magepanda	2021	219	606

229	Alok	2021	9,2	180
230	Alok Barat	2021	13	386
231	Alok Timur	2021	6,8	32
232	Paga	2022	470	1191
233	Mego	2022	544,3	1712
234	Tanawawo	2022	380,3	1463
235	Lela	2022	122	675
236	Bola	2022	610	1659
237	Doreng	2022	370,5	1439
238	Mapitara	2022	385	963
239	Talibura	2022	74	1096
240	Waigete	2022	201,7	1098
241	Waiblama	2022	26	463
242	Kewapante	2022	288	864
243	Hewokloang	2022	1160,8	2689
244	Kangae	2022	271	1204
245	Palue	2022	120,5	451
246	Koting	2022	214,7	799
247	Nelle	2022	437	926
248	Nita	2022	2138	2630
249	Magepanda	2022	219	606
250	Alok	2022	3,8	180
251	Alok Barat	2022	13	386
252	Alok Timur	2022	16,8	32



## LAMPIRAN SOURCE CODE

### 1. Source Code Kecamatan Controller

```
<?php

namespace App\Http\Controllers;

use App\Models\Kecamatan;
use App\Models\warehouse;
use Illuminate\Http\Request;
use Dompdf\Dompdf;
use RealRashid\SweetAlert\Facades\Alert;

class KecamatanController extends Controller
{
    public function index()
    {
        $dtlokasi = Kecamatan::all();
        $title = 'Hapus Kecamatan';
        $text = "Apakah anda yakin untuk hapus?";
        confirmDelete($title, $text);
        return view('kecamatan.index', compact('dtlokasi'));
    }

    public function create()
    {
        return view('kecamatan.create');
    }

    public function store(Request $request)
    {
        Kecamatan::create([
            'kecamatan' => $request->input('kecamatan'),
        ]);
        Alert::success("Success", "Data berhasil disimpan");

        return redirect('kecamatan');
    }

    public function show(string $id)
    {
        //
    }

    public function edit(Kecamatan $kecamatan)
    {
        return view('kecamatan.edit', compact('kecamatan'));
    }

    public function update(Request $request, Kecamatan $kecamatan)
    {
        $dtlokasi = $request->all();
        $kecamatan->update($dtlokasi);
        Alert::success("Success", "Data berhasil diupdate");
        return redirect("kecamatan");
    }
}
```

```
public function destroy(Kecamatan $kecamatan)
{
    $kecamatan->delete();
    Alert::success("Success", "Data berhasil dihapus");
    return redirect('kecamatan');
}
}
```

## 2. Source Code Luas Lahan Controller

```
<?php

namespace App\Http\Controllers;

use App\Models\Kecamatan;
use App\Models\Luaslahan;
use App\Models\warehouse;
use Illuminate\Http\Request;
use Dompdf\Dompdf;
use RealRashid\SweetAlert\Facades\Alert;

class LuasLahanController extends Controller
{
    public function index()
    {
        $dtlokasi = Luaslahan::orderBy('tahun', 'asc')->get();
        $title = 'Hapus Lokasi';
        $text = "Apakah anda yakin untuk hapus?";
        confirmDelete($title, $text);
        return view('luaslahan.index', compact('dtlokasi'));
    }

    public function create()
    {
        $kecamatan = Kecamatan::all();
        return view('luaslahan.create', compact('kecamatan'));
    }

    public function store(Request $request)
    {
        Luaslahan::create([
            'id_kecamatan' => $request->input('id_kecamatan'),
            'luas_lahan' => $request->input('luas_lahan'),
            'tahun' => $request->input('tahun'),
        ]);
        Alert::success("Success", "Data berhasil disimpan");
        return redirect('luaslahan');
    }

    public function show(string $id)
    {
        //
    }

    public function edit(Luaslahan $luaslahan)
    {

```

```

        $kecamatan = Kecamatan::all();
        return view('luaslahan.edit', compact('luaslahan',
'kecamatan'));
    }

    public function update(Request $request, Luaslahan
$luaslahan)
    {
        $dtlokasi = $request->all();
        $luaslahan->update($dtlokasi);
        Alert::success("Success", "Data berhasil diupdate");
        return redirect("luaslahan");
    }

    public function destroy(Luaslahan $luaslahan)
    {
        $luaslahan->delete();
        Alert::success("Success", "Data berhasil dihapus");
        return redirect('luaslahan');
    }
}

```

### 3. Source Code Produksi Controller

```

<?php

namespace App\Http\Controllers;

use App\Models\Kecamatan;
use App\Models\Lokasi;
use App\Models\Produksi;
use Illuminate\Http\Request;
use RealRashid\SweetAlert\Facades\Alert;

class ProduksiController extends Controller
{
    public function index()
    {
        $produksi = Produksi::orderBy('tahun', 'asc')->get();
        $title = 'Hapus Produksi';
        $text = "Apakah anda yakin untuk hapus?";
        confirmDelete($title, $text);
        return
view('Produksi.produksi', compact('produksi'));
    }

    public function create()
    {
        $kecamatan = Kecamatan::all();
        return view('Produksi.create', compact('kecamatan'));
    }

    public function store(Request $request)
    {
        Produksi::create([
            'id_kecamatan' => $request->input('id_kecamatan'),
            'tahun' => $request->input('tahun'),
            'produksi' => $request->input('produksi'),

```

```

    });

    Alert::success("Success", "Data berhasil disimpan");
    return redirect('produksi');
}

public function show(string $id)
{
    //
}

public function edit(Produksi $produksi)
{
    $kecamatan = Lokasi::all();
    return view('Produksi.editproduksi',
compact('produksi', 'kecamatan'));
}

public function update(Request $request, Produksi
$produksi)
{
    $data = $request->all();
    $produksi->update($data);
    Alert::success("Success", "Data berhasil diupdate");
    return redirect("produksi");
}

public function destroy(Produksi $produksi)
{
    $produksi->delete();
    Alert::success("Success", "Data berhasil dihapus");
    return redirect('produksi');
}
}

```

#### 4. Source Code Klasterisasi Controller

```

<?php

namespace App\Http\Controllers;

use App\Models\Kecamatan;
use App\Models\LogElbowModel;
use App\Models\Lokasi;
use App\Models\Produksi;
use App\Models\TbClustering;
use App\Models\TbKecamatan;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Storage;
use Illuminate\Support\Facades\Validator;
use RealRashid\SweetAlert\Facades\Alert;

class KlasterisasiController extends Controller
{
    public function index()
    {
        $tahunList = Produksi::selectRaw('tahun')->distinct()->orderBy('tahun', 'asc')->pluck('tahun');
    }
}

```



```

        return view('klasterisasi.index',
compact('tahunList'));
    }

    public function fetchData(Request $request)
    {
        $tahunDipilih = $request->input('tahun');
        if (!$tahunDipilih) {
            return response()->json(['status' => 'error',
'message' => 'Tahun belum dipilih.'], 400);
        }
        $data = TbClustering::with('kecamatan')->where('tahun', $tahunDipilih)->get();
        if ($data->isEmpty()) {
            return response()->json(['status' => 'empty',
'message' => 'Data klasterisasi tidak ditemukan untuk tahun ini.']);
        }
        return response()->json(['status' => 'success',
'data' => $data]);
    }

    public function cetak($tahun)
    {
        $data = TbClustering::where('tahun', $tahun)->get();
        return view('klasterisasi.cetak', compact('data'));
    }

    private function elbowMethod($data, $maxClusterCount = 10)
    {
        $sseValues = [];
        $existingLogs = LogElbowModel::all();

        if ($existingLogs->isNotEmpty()) {
            foreach ($existingLogs as $log) {
                $sseValues[$log->cluster_count] = $log->sse_value;
            }
            return $sseValues;
        }

        for ($k = 1; $k <= $maxClusterCount; $k++) {
            $clusters = $this->kMeans($data, $k);
            $sse = 0;
            foreach ($clusters as $clusterData) {
                foreach ($clusterData as $point) {
                    $centroid = $this->getCentroidOfCluster($point['id_kecamatan'], $clusters);
                    $sse += pow($this->euclideanDistance($point, $centroid), 2);
                }
            }
            $sseValues[$k] = $sse;
            LogElbowModel::create(['cluster_count' => $k,
'sse_value' => $sse]);
        }

        return $sseValues;
    }

```

```

    }

    private function euclideanDistance($point, $centroid)
    {
        return sqrt(pow($point['total_luas_lahan'] -
$centroid['total_luas_lahan'], 2) +
        pow($point['total_produksi'] -
$centroid['total_produksi'], 2));
    }

    private function getCentroidOfCluster($kecamatanId,
$clusters)
    {
        foreach ($clusters as $cluster) {
            foreach ($cluster as $point) {
                if ($point['id_kecamatan'] == $kecamatanId) {
                    return [
                        'total_luas_lahan' =>
array_sum(array_column($cluster, 'total_luas_lahan')) /
count($cluster),
                        'total_produksi' =>
array_sum(array_column($cluster, 'total_produksi')) /
count($cluster),
                    ];
                }
            }
        }
        return null;
    }
}

public function findOptimalCluster($sseValues)
{
    if (empty($sseValues) || count($sseValues) < 2) {
        throw new \Exception('Jumlah nilai SSE tidak cukup
untuk menentukan jumlah kluster optimal.');
```

```

        $tahun = $request->input('tahun');
        $dataKakao = Kecamatan::selectRaw('
            kecamatan.id AS id_kecamatan,
            kecamatan.kecamatan,
            luaslahan.tahun,
            SUM(luaslahan.luas_lahan) AS total_luas_lahan,
            SUM(produksi.produksi) AS total_produksi
        ')
        ->join('luaslahan', function ($join) {
            $join->on('luaslahan.id_kecamatan', '=',
'kecamatan.id');
        })
        ->join('produksi', function ($join) {
            $join->on('produksi.id_kecamatan', '=',
'kecamatan.id')
            ->on('produksi.tahun', '=',
'luaslahan.tahun');
        })

        ->where('luaslahan.tahun', $tahun)
        ->groupBy('kecamatan.id', 'kecamatan.kecamatan',
'luaslahan.tahun')
        ->get();

        $data = [];
        foreach ($dataKakao as $item) {
            $data[] = [
                'id_kecamatan' => $item->id_kecamatan,
                'kecamatan' => $item->kecamatan,
                'tahun' => $item->tahun,
                'total_luas_lahan' => $item->total_luas_lahan,
                'total_produksi' => $item->total_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');
        }

        $sseValues = $this->elbowMethod($data, 6);
        $jumlahCluster = $this->findOptimalCluster($sseValues);
        $clusters = $this->kMeans($data, $jumlahCluster,
$tahun);

        TbClustering::where('tahun', $tahun)->delete();
        $this->insertHasilKlasterisasi($clusters,
$currentHash);

        Alert::success('Success', 'Klasterisasi Berhasil');
        return redirect('clustering');
    }
}

```

```

private function kMeans($data, $k, $tahun = null)
{
    $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, $tahun);
        if ($centroids == $newCentroids) {
            break;
        }
        $centroids = $newCentroids;
    }
    ksort($clusters);
    return $clusters;
}

private function labelCentroids($centroids)
{
    $labeledCentroids = [];
    foreach ($centroids as $index => $centroid) {
        $label = 'C' . ($index + 1);
        $labeledCentroids[] = ['label' => $label,
'centroid' => $centroid];
    }
    return $labeledCentroids;
}

private function insertHasilKlasterisasi($clusters,
$currentHash)
{
    foreach ($clusters as $index => $clusterData) {
        $clusterLabel = 'C' . ($index + 1);
        foreach ($clusterData as $data) {
            DB::table('tb_clustering')->insert([
                'id_kecamatan' => $data['id_kecamatan'],
                'tahun' => $data['tahun'],
                'luas_lahan' =>
$data['total_luas_lahan'],
                'produksi' => $data['total_produksi'],
                'cluster' => $clusterLabel,
                'created_at' => now(),
                'data_hash' => $currentHash,
            ]);
        }
    }
}

private function initCentroids($data, $k)

```



```

    {
        usort($data, function ($a, $b) {
            return
                ($a['total_luas_lahan']          +
                $a['total_produksi'])             <=>
                ($b['total_luas_lahan']          +
                $b['total_produksi']);
        });
        return [
            $data[0],
            $data[floor(count($data) / 2)],
            $data[count($data) - 1]
        ];
    }

    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_lahan'] -
                    $centroid['total_luas_lahan'], 2) +
                    pow($point['total_produksi'] - $centroid['total_produksi'],
                    2));
                if ($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_lahan'));
                $produksiTotal =
                    array_sum(array_column($clusters[$i], 'total_produksi'));
                $clusterSize = count($clusters[$i]);
                $newCentroids[] = [
                    'total_luas_lahan' => $luasTotal /
                    $clusterSize,
                    'total_produksi' => $produksiTotal /
                    $clusterSize
                ];
            } else {
                $newCentroids[] = $this->initCentroids($data,
                1)[0];
            }
        }
    }

    return $newCentroids;
}

```

```

    }

    private function insertIterationData($iteration,
    $centroids, $clusters, $tahun)
    {
        $labeledCentroids = [];
        foreach ($centroids as $index => $centroid) {
            $labeledCentroids[] = ['label' => 'C' . ($index +
1), 'centroid' => $centroid];
        }
        DB::table('tb_log_iterasi')->insert([
            'tahun' => $tahun,
            'iterasi' => $iteration,
            'centroid_data' =>
json_encode($labeledCentroids),
            'cluster_data' => json_encode($clusters),
            'created_at' => now(),
        ]);
    }
}

```

## 5. Source Code Elbow Controller

```

<?php

namespace App\Http\Controllers;

use App\Models\LogElbowModel;
use Illuminate\Http\Request;

class ElbowController extends Controller
{
    public function getElbowData()
    {
        $logElbowData =
LogElbowModel::orderBy('cluster_count')-
>get(['cluster count', 'sse value']);

        if ($logElbowData->isEmpty()) {
            return response()->json(['status' => 'error',
'message' => 'No data found']);
        }

        return response()->json(['status' => 'success',
'data' => $logElbowData]);
    }
}

```