

# BAB V. IMPLEMENTASI DAN PENGUJIAN

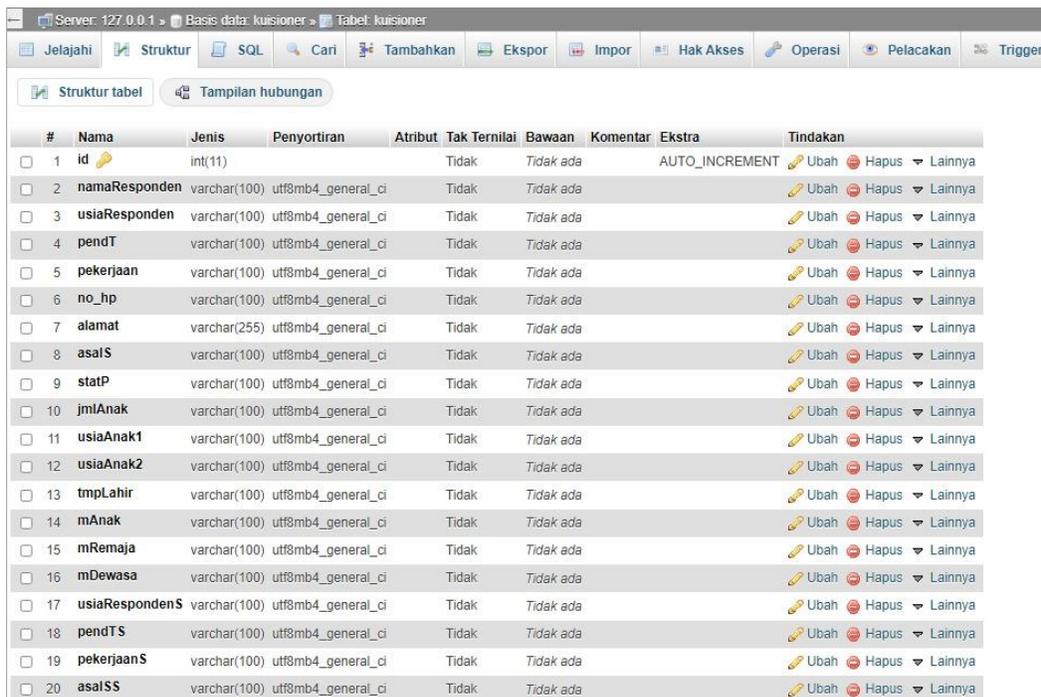
## 5.1 Implementasi Basis Data

Pada sub ini merupakan implementasi dari perancangan basis data sesuai dengan rancangan yang telah dilakukan pada bab sebelumnya. Basis data yang digunakan dalam sistem *clustering* respon emosi marah wanita jawa adalah *MySQL*. Basis data yang dibuat diberi nama “skripsi\_wahid” yang memiliki 5 tabel yaitu tabel *clustering*, *cost*, kedekatan, *k-medoid* dan normalisasi. Berikut ini database yang telah dibuat. (5.1)



Gambar 5.1 Tampilan Database “skripsi\_wahid”

Tabel-tabel tersebut digunakan untuk menyimpan data yang didapat dari hasil proses *clustering* pada aplikasi. Disini juga menggunakan basis data yang dibuat dan diberi nama “kuesioner” yang mana memiliki 1 tabel yaitu tabel kuesioner. Berikut ini database yang telah dibuat. (5.2)



Gambar 5.2 Tampilan Database Kuesioner (1)

Tabel kuesioner ini digunakan untuk menyimpan data yang didapat dari hasil upload pada aplikasi. Pada Gambar 5.2 Tabel Kuesioner tersebut merupakan data identitas dari responden yang akan digunakan sebagai data dukung proses *clustering*.

Gambar 5.3 Tampilan Database Kuesioner (2)

42	2_p5	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
43	2_p6	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
44	2_p7	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
45	2_p8	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
46	2_p9	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
47	2_p10	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
48	3_p1	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
49	3_p2	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
50	3_p3	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
51	3_p4	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
52	3_p5	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
53	3_p6	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
54	3_p7	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
55	3_p8	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
56	3_p9	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
57	3_p10	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
58	4_p1	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
59	4_p2	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
60	4_p3	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
61	4_p4	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
62	4_p5	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
63	4_p6	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya
64	4_p7	varchar(1)	utf8mb4_general_ci	Tidak	Tidak ada	Ubah Hapus Lainnya

Pada Gambar 5.3 Merupakan lanjutan dari tabel kuesioner tersebut, data yang ada pada *field* inilah yang akan diolah pada proses *clustering*. *Field* ini terdiri dari 1\_p1 hingga 5\_p10.

Selain database untuk kuesioner, disini juga menggunakan database yang mana untuk menyimpan proses dan juga hasil dari *clustering*. Berikut untuk hasil implementasi database yang telah dibuat.

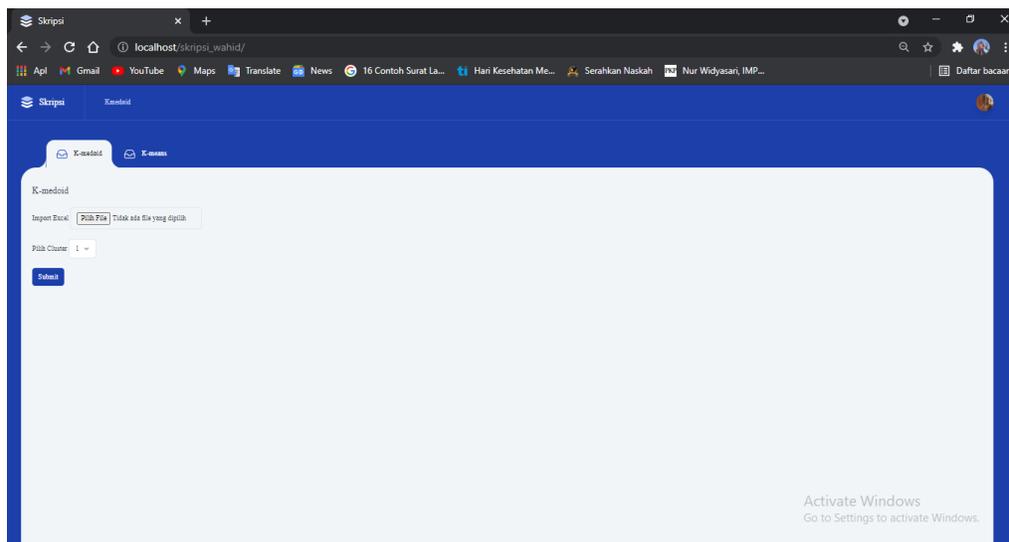
<b>skripsi_wahid normalisasi</b> nama : varchar(100) kota : varchar(25) p1_1 : varchar(10) p2_1 : varchar(10) p3_1 : varchar(10) p4_1 : varchar(10) p5_1 : varchar(10) p6_1 : varchar(10) p7_1 : varchar(10) p8_1 : varchar(10) p1_2 : varchar(10) p2_2 : varchar(10) p3_2 : varchar(10) p4_2 : varchar(10) p5_2 : varchar(10) p6_2 : varchar(10) p7_2 : varchar(10) p8_2 : varchar(10) p9_2 : varchar(10) p10_2 : varchar(10)	<b>skripsi_wahid kedekatan</b> kedekatan : varchar(20)	<b>skripsi_wahid kmedoid</b> nama : varchar(100) kota : varchar(25) p1_1 : varchar(10) p2_1 : varchar(10) p3_1 : varchar(10) p4_1 : varchar(10) p5_1 : varchar(10) p6_1 : varchar(10) p7_1 : varchar(10) p8_1 : varchar(10) p1_2 : varchar(10) p2_2 : varchar(10) p3_2 : varchar(10) p4_2 : varchar(10) p5_2 : varchar(10) p6_2 : varchar(10) p7_2 : varchar(10) p8_2 : varchar(10) p9_2 : varchar(10)	<b>skripsi_wahid clustering</b> nama_clustering : varchar(100) daerah_clustering : varchar(100) cost_clustering : varchar(100)
--	---	---	---

Gambar 5.4 Tampilan Database *K-Medoid Clustering*

Pada gambar 5.4 terdapat 6 buah tabel yaitu tabel normalisasi, dimana tabel ini guna untuk menyimpan hasil dari proses normalisasi pada tahap *clustering* menggunakan metode *k-medoid*. Setelah disimpan hasil normalisasi, dilakukan perhitungan *cost* dan lalu kedekatan. Dimana hasilnya nanti akan disimpan pada tabel *cost* dan juga tabel *kedekatan*. Lalu, selanjutnya terdapat tabel *k-medoid* berdasarkan hasil *cost* dan hasil kedekatan diperoleh data-data pengelompokan berdasarkan kedekatan, hasil pengelompokan disimpan sementara untuk dapat ditampilkan menjadi hasil *clustering*. Hasil dari *clustering* tersebut disimpan kedalam tabel database *clustering*. Yang mana setelah itu akan dilakukan pelabelan sesuai dengan kelompok-kelompok yang sesuai dengan jawabannya.

## 5.2 Implementasi Sistem

Implementasi sistem merupakan implementasi dari perancangan antarmuka sistem yang dibuat bentuk *layout*. Rancangan yang dibuat tersebut kemudian diimplementasikan untuk membangun aplikasi menggunakan PHP sebagai bahasa pemrograman. Berikut merupakan *interface* yang terdapat pada sistem *Clustering Respon Emosi Marah Wanita Jawa*.



Gambar 5.5 Tampilan Awal Sistem

Pada tampilan awal, terdapat *button click* dimana *button* ini akan digunakan untuk upload *file* data yang akan di *upload* dan dilakukan proses perhitungan. Pada Gambar 5.5. Juga terdapat tab menu pilihan yaitu *K-Medoid* dan juga *K-Means*. Pada halaman awal juga terdapat tombol *drop down* yang mana jika dilakukan *click* maka akan muncul angka-angka *cluster* yang akan di bentuk. Selain itu, juga terdapat *button click* proses. Jadi, jika *file* data excel sudah di *upload* selanjutnya pilih jumlah *cluster* yang akan dibentuk lalu klik *button* proses maka system akan memproses sesuai dengan tab menu mana yang dipilih.

Apabila proses *clustering* sudah dijalankan, maka hasil yang akan tampil pada sistem adalah seperti Gambar 5.6 Tampilan Hasil Proses *Clustering*. Untuk warna merah muda, menunjukkan bahwa data tersebut merupakan pusat *cluster* nya dan untuk yang warna biru laut menunjukkan data anggotanya.

Gambar 5.6 Tampilan Hasil Clustering K-Medoid

Setelah muncul hasil *clustering* lalu ditekan tombol analisa, untuk melihat persentase setiap aspeknya. Berikut dapat dilihat hasil analisa tiap-tiap aspek pada Gambar 5.7 dan 5.8.

Kategori	Emosi	Cluster 1	Cluster 2	Cluster 3
Suami	Tidak Marah	19.00%	10.47%	9.26%
	Sedikit Marah	23.50%	19.26%	8.80%
	Agak Marah	17.00%	25.34%	12.50%
	Marah	27.00%	34.46%	29.17%
	Sangat Marah	13.50%	10.47%	40.28%
Anak	Tidak Marah	17.20%	7.57%	4.07%
	Sedikit Marah	19.60%	17.57%	10.37%
	Agak Marah	18.80%	25.95%	16.30%

Gambar 5.7 Tampilan Hasil Analisa Tiap Aspek (1)

Saudara	Tidak Marah	16.40%	5.68%	5.93%
	Sedikit Marah	25.20%	12.43%	3.33%
	Agak Marah	22.40%	25.95%	11.48%
	Marah	20.40%	31.35%	34.44%
	Sangat Marah	15.60%	24.32%	44.81%
Pekerjaan	Tidak Marah	22.00%	6.22%	3.33%
	Sedikit Marah	32.40%	29.19%	9.26%
	Agak Marah	18.40%	27.84%	14.07%
	Marah	23.60%	22.70%	35.56%
	Sangat Marah	3.60%	14.05%	37.78%

Gambar 5.8 Tampilan Hasil Analisa Tiap Aspek (2)

Sebelum dilakukan langkah pengambilan pusat *cluster* langkah awal metode ini yaitu mencari minimum dan maximum data serta normalisasi. Berikut kode program minimum dan maximum.

```

$max_p1_1 = max(array_column($data, 'p1_1'));
$min_p1_1 = min(array_column($data,
'p1_1'));
$max_p2_1 = max(array_column($data,
'p2_1'));
$min_p2_1 = min(array_column($data,
'p2_1'));
$max_p3_1 = max(array_column($data,
'p3_1'));
$min_p3_1 = min(array_column($data,
'p3_1'));

```

Setelah dilakukan min max, selanjutnya normalisasi. Berikut kode program proses normalisasi.

```
foreach ($data as $a) {
    $normalisasi[] = array(
        'nama'      => $a['nama'],
        'kota'      => $a['kota'],
        'p1_1'      =>
number_format(($a['p1_1']-$min_p1_1)/($max_p1_1-
$min_p1_1),3),
        'p2_1'      =>
number_format(($a['p2_1']-$min_p2_1)/($max_p2_1-
$min_p2_1),3),
        'p3_1'      =>
number_format(($a['p3_1']-$min_p3_1)/($max_p3_1-
$min_p3_1),3),
        'p4_1'      =>
number_format(($a['p4_1']-$min_p4_1)/($max_p4_1-
$min_p4_1),3),
        'p5_1'      =>
number_format(($a['p5_1']-$min_p5_1)/($max_p5_1-
$min_p5_1),3),
        'p6_1'      =>
```

Untuk langkah pengambilan pusat *cluster* pada metode *K-Medoids*, berikut untuk kode programnya.

```
foreach          ((array)array_rand($normalisasi,
$_POST['cluster']) as $index) {
    $row = $normalisasi[$index];
    $kmedoid1[] = array(
        'nama'      => $row['nama'],
        'kota'     => $row['kota'],
        'p1_1'     => $row['p1_1'],
        'p2_1'     => $row['p2_1'],
        'p3_1'     => $row['p3_1'],
        'p4_1'     => $row['p4_1'],
        'p5_1'     => $row['p5_1'],
        'p6_1'     => $row['p6_1'],
        'p7_1'     => $row['p7_1'],
        'p8_1'     => $row['p8_1'],
        'p1_2'     => $row['p1_2'],
        'p2_2'     => $row['p2_2'],
        'p3_2'     => $row['p3_2'],
        'p4_2'     => $row['p4_2'],
        'p5_2'     => $row['p5_2'],
        'p6_2'     => $row['p6_2'],
```



Langkah selanjutnya, yaitu mencari kedekatan berdasarkan perhitungan jarak. Berikut kode program untuk mencari kedekatan jarak.

```
$kedekatan1 = array();
    foreach ($cost1 as $keys => $a){
        foreach($a as $key => $val){
            $tmp[$key][] = array('nilai_cost' =>
$val['nilai_cost'], 'daerah' => $val['daerah'],
'posisi' => $keys+1);
        }
    }

    foreach($tmp as $key => $val){
        sort($tmp[$key]);
        $kedekatan1[$key] = $tmp[$key][0];
    }
```

Setelah mendapatkan hasil kedekatan seluruh data, langkah selanjutnya yaitu menghitung total seluruh kedekatannya, berikut kode program proses perhitungan total seluruh kedekatan.

```
$jumlah_kedekatan1 =
array_sum(array_column($kedekatan1, 'nilai_cost'));
```

Dilakukan langkah ulang untuk mendapatkan total kedekatan baru. Dimulai dari langkah pengambilan pusat *cluster* hingga menghitung seluruh total kedekatan. Setelah menghitung total

kedekatan, dilakukan proses mencari selisih kedekatan, berikut kode program proses perhitungan selisih kedekatan.

```
selisih = $jumlah_kedekatan2 -
$jumlah_kedekatan1;
}while($selisih < '0');
$kedekatan = array();
$tmp = array();
foreach ($cost1 as $keys => $a){
    foreach($a as $key => $val){
        $tmp[$key][] = array('nilai_cost' =>
$val['nilai_cost'], 'daerah' => $val['daerah'],
'posisi' => $keys+1);
    }
}
```

Jadi kode program tersebut menjelaskan, bahwa jika hasil selisih kurang dari 0, maka dia akan mengulang perhitungan dari awal. Tetapi, jika sudah lebih dari 0, maka sistem akan mengambil hasil perhitungan pertama kali.

### 5.3 Pengujian

Pengujian disini, penulis menggunakan pengujian dengan menggunakan *Silhouette Coefficient* dimana disini metode *K-Medoid* dan *K-Means* di uji keakuratannya berdasarkan data yang digunakan dengan sistem yang sudah dibuat. Berikut untuk tampilan uji keakuratan sistem.

Uji akurasi *K-Means* dapat dilihat pada Gambar 5.9 dan untuk uji akurasi *K-Medoid* dapat dilihat pada Gambar 5.10

Hasil Cluster	Nilai SKJ
1	0.90547226266183
2	0.99751759565326
3	0.96164517946355
Global	0.73865900944466
SC	2.1824485289351

Gambar 5.9 Tampilan Hasil Akurasi K-Means

SC K-medoid

Show 10 entries Search:

Hasil Cluster	Nilai SJJ
1	0.99831269510904
2	0.98357349499405
3	0.99059930132177
Global	0.74312137285621
SC	2.2089174991828

Showing 1 to 5 of 5 entries Previous 1 Next

Activate Windows  
Go to Settings to activate Windows.

Gambar 5.10 Tampilan Hasil Akurasi K-Medoid