

BAB IV

IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi berdasarkan metodologi penelitian dan perancangan yang telah dibahas pada bab sebelumnya.

4.1 Lingkungan implementasi

Dalam menerapkan metode penelitian ke dalam sistem serta dalam pengembangan sistem, perlu adanya beberapa faktor yang perlu diperhatikan dalam menerapkan metode dan pengembangan sistem. Tujuannya untuk memenuhi kebutuhan dari sistem yang nantinya akan dikembangkan dan metode yang diimplementasikan. Lingkungan implementasi yang akan dijelaskan pada sub bab ini adalah lingkungan perangkat keras (*hardware*) dan lingkungan perangkat lunak (*software*) yang digunakan untuk membangun sistem klasifikasi kanker payudara menggunakan algoritma K-Means dan Naïve Bayes.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan sistem klasifikasi kanker payudara menggunakan algoritma K-Means dan Naïve Bayes adalah sebagai berikut :

1. Processor Intel® Core™ i5 CPU M 460 @ 2.53GHz.
2. RAM 2.00 GB.
3. Harddisk 500 GB.
4. Monitor 14.1”.
5. Keyboard.
6. Mouse.

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan sistem klasifikasi kanker payudara menggunakan algoritma K-Means dan Naïve Bayes adalah sebagai berikut :

1. Sistem operasi Windows 7 Ultimate.
2. CodeLobster PHP Edition Free Version 4.2.1 sebagai *software development* dalam implementasi perancangan sistem.
3. Microsoft Excel 2007 sebagai pengolahan data.

4.2 Implementasi Program

4.2.1 Proses Input Data Latih

Pada proses ini digunakan untuk mendapatkan masukan berupa data latih yang memiliki format file yaitu (.)csv. Data masukan terdiri dari 9 atribut yang akan digunakan untuk perhitungan K-Means dan 10 atribut untuk perhitungan Naïve Bayes dengan indikator yang bervariasi antara lain adalah 1-10 yang akan digunakan untuk pemrosesan data. Fungsi proses input data latih ditunjukkan pada *source code* 4.1.

```
<form method="POST" action="m_pelatihan.php" enctype="multipart/form-data">
    <h3>Pilih Data Pelatihan</h3>
    <table align="left" style="margin-left:5px;">
        <tr>
            <td><input class="styled-tombol-6" type="file"
name="data_latih"/><b>&nbsp; : &nbsp;   </b></td>
            <td><input class="styled-tombol-10" type="submit"
name="Latih" value="Latih_KMNB" style="border:1px solid; margin-
top:15px;"/></td> </tr> </table></form><br /><br /><br /><br /><br />
<?php
    if(isset($_POST['Latih'])) {
        $nama_file = $_FILES['data_latih']['name'];
        $data_latih = "data/$nama_file";
        $k=3;
        set_time_limit(3600);
        pelatihan($data_latih, $k);}
?><br />
```

Source code 4.1 Proses input data latih

4.2.2 Proses membuat *centroid* baru secara acak

Proses membuat *centroid* baru berfungsi untuk menentukan *centroid* awal. Penentuan *centroid* awal dilakukan secara random/acak dari data/obyek yang tersedia sebanyak jumlah kluster k. Parameter yang dimiliki dalam proses ini

yaitu, \$dimensi, \$mini, dan \$maxi. Fungsi membuat *centroid* baru dapat dilihat pada *source code* 4.2.

```
function _buatCentroid($dimensi, $mini, $maxi){
    $total = 0;
    for($i = 0; $i < $dimensi; $i++) {
        $centroid[$i] = (mt_rand($mini[$i] * 10, $maxi[$i] * 10));
        $total += $centroid[$i] * $centroid[$i]; }
    return self::_normalisasi($centroid, sqrt($total));
}
```

Source code 4.2 Proses membuat *centroid* baru secara acak

4.2.3 Proses menghitung jarak data ke *centroid*

Jarak euclidian sering digunakan dalam perhitungan jarak, hal ini dikarenakan hasil yang diperoleh merupakan jarak terpendek dari dua titik yang diperhitungkan. Dalam menghitung jarak antara dua titik pada K-Means digunakan jarak euclidian. Pada proses ini akan menghasilkan jarak minimum yang akan ditandai ke dalam sebuah *cluster*. Parameter yang dimiliki antara lain, \$data_latih dan \$centroid. Fungsi untuk perhitungan jarak data ke *centroid* dapat dilihat pada *source code* 4.3.

```
function _tandaiCentroid($data_latih, $centroid){
    foreach($data_latih as $key => $val) {
        $jarakMinimum = 100;
        $centroidMinimum = null;
        foreach($centroid as $kunci => $nilai) {
            $jarak = 0;
            foreach($nilai as $vlib => $kids) {
                $jarak += abs($kids - $val[$vlib]);}
            if($jarak < $jarakMinimum) {
                $jarakMinimum = $jarak;
                $centroidMinimum = $kunci; }
        }$arrayjarak[$key] = $centroidMinimum;}
    return $arrayjarak;
}
```

Source code 4.3 Proses menghitung jarak data ke *centroid*

4.2.4 Proses meng-update *centroid*

Setelah didapatkan anggota tiap *cluster* dari perhitungan jarak euclidian, maka perhitungan untuk *centroid* baru akan dilakukan. Untuk mendapatkan *centroid* baru dilakukan dengan cara menghitung rata-rata data anggota *cluster* dari setiap *cluster*. Parameter yang dimiliki adalah, \$arrayjarak, \$data_latih, dan \$k. Fungsi untuk mengupdate *centroid* dapat dilihat pada *source code* 4.4.

```
function _updateCentroid($arrayjarak, $data_latih, $k){
    $jumlah = array_count_values($arrayjarak);
    foreach($arrayjarak as $key => $val) {
        foreach($data_latih[$key] as $kunci => $nilai){
            $jarak[$val][$kunci] += ($nilai/$jumlah[$val]);}}
    if(count($jarak) < $k) {
        $jarak = array_merge($jarak, self::_inisialisasi($data_latih, $k
        - count($jarak)));}
    return $jarak;
}
```

Source code 4.4 Proses mengupdate *centroid*

4.2.5 Proses mendapatkan data latih untuk Naïve Bayes

Proses untuk mendapatkan data latih untuk Naïve Bayes mengambil informasi dari data latih yang dipilih, selanjutnya akan ditujukan ke file data latih yang memiliki kelas. Dalam kasus ini file untuk pemrosesan K-Means berbeda dengan file data untuk pemrosesan Naïve Bayes. Hal ini dilakukan karena data latih awal untuk masukan K-Means tidak menyertakan atribut kelas, sehingga atribut yang digunakan menjadi berjumlah 9 atribut, sedangkan untuk proses Naïve Bayes dibutuhkan 10 atribut (dengan kelas). Fungsi yang dimiliki untuk pemrosesan data latih Naïve Bayes memiliki atribut \$data_latih dan \$jumlah_data. Setelah mendapatkan data latih untuk Naïve Bayes dilakukan pemecahan data untuk mendapatkan array data berupa \$x1, \$x2, \$x3, \$x4, \$x5, \$x6, \$x7, \$x8, \$x9, dan \$x10. Fungsi pemrosesan data latih untuk Naïve Bayes dapat dilihat pada *source code* 4.5.

```
Function      getDataLatih($data_uji,          $jumlah_data_latih,
    $jumlah_data_uji){
    echo "JUMLAH DATA LATIH = ".$jumlah_data_latih."<br>";
    echo "JUMLAH DATA UJI = ".$jumlah_data_uji."<br>";
    $fp = fopen('data/temp_nama_data_latih.csv', 'r');
```

```

$nama_data_latih =
get1DArray('data/temp_nama_data_latih.csv', ',');
$nama_file = $nama_data_latih[0];
if($nama_file == "d_latih_40%.csv"){
$xi=get2DArray_Input('data/data_seimbang/data_latih/dengan_kelas/d
_latih_40%.csv', ',');}
else if($nama_file == "d_latih_50%.csv"){
$xi=get2DArray_Input('data/data_seimbang/data_latih/dengan_kelas/d
_latih_50%.csv', ','); }
else if($nama_file == "d_latih_60%.csv"){
$xi=get2DArray_Input('data/data_seimbang/data_latih/dengan_kelas/d
_latih_60%.csv', ',');}
else if($nama_file == "d_latih_70%.csv"){
$xi=get2DArray_Input('data/data_seimbang/data_latih/dengan_kelas/d
_latih_70%.csv', ',');}
else if($nama_file == "d_latih_80%.csv"){
$xi=get2DArray_Input('data/data_seimbang/data_latih/dengan_kelas/d
_latih_80%.csv', ',');}
else if($nama_file == "d_latih_30%.csv") {
$xi=get2DArray_Input('data/data_seimbang/data_latih/dengan_kelas/d
_latih_30%.csv', ','); }
if($jumlah_data_uji == 88){
$xi_uji=get2DArray_Input('data/data_seimbang/data_uji/dengan_kelas
/d_uji_20%.csv', ','); }
else if($jumlah_data_uji == 132){
$xi_uji=get2DArray_Input('data/data_seimbang/data_uji/dengan_kelas
/d_uji_30%.csv', ','); }
else if($jumlah_data_uji == 176){
$xi_uji=get2DArray_Input('data/data_seimbang/data_uji/dengan_kelas
/d_uji_40%.csv', ','); }
else if($jumlah_data_uji == 220){
$xi_uji=get2DArray_Input('data/data_seimbang/data_uji/dengan_kelas
/d_uji_50%.csv', ','); }
else if($jumlah_data_uji == 264){
$xi_uji=get2DArray_Input('data/data_seimbang/data_uji/dengan_kelas
/d_uji_60%.csv', ','); }
else if($jumlah_data_uji == 308){
$xi_uji=get2DArray_Input('data/data_seimbang/data_uji/dengan_kelas
/d_uji_70%.csv', ','); }
$i = 0;
while($i < $jumlah_data_latih){
for($j = 0; $j<10; $j++){
$ddata[$i][$j] = $xi[$i][$j]; }
$x1[$i] = $ddata[$i][0];
$x2[$i] = $ddata[$i][1];
$x3[$i] = $ddata[$i][2];
$x4[$i] = $ddata[$i][3];
$x5[$i] = $ddata[$i][4];
$x6[$i] = $ddata[$i][5];
$x7[$i] = $ddata[$i][6];

```

```

        $x8[$i] = $data[$i][7];
        $x9[$i] = $data[$i][8];
        $x10[$i] = $data[$i][9];
        $i++; }
    $k = 0;
    while($k < $jumlah_data_uji){
        for($l = 0; $l<10; $l++){
            $data_aktual[$k][$l] = $xi_uji[$k][$l];
        }
        $x9_uji[$k] = $data_aktual[$k][8];
        $x10_uji[$k] = $data_aktual[$k][9];
        $k++; }
    return array($x1, $x2, $x3, $x4, $x5, $x6, $x7, $x8,
        $x9, $x10, $x10_uji);
}

```

Source code 4.5 Proses mendapatkan data latihan untuk Naïve Bayes

4.2.6 Proses menghitung rata-rata dan varian Naïve Bayes

Nilai rata-rata dapat ditentukan dengan membagi jumlah data dengan banyaknya data. Fungsi rata-rata memiliki parameter \$jumlah_data, \$xi, \$x10, \$count_ganas, dan \$count_jinak. Dimana jika \$x10[\$i] = 4 maka data tersebut merupakan kategori ganas dan jika \$x10[\$i] = 2 maka data tersebut merupakan kategori jinak. Setelah didapatkan data berdasarkan kategorinya lalu dihitung masing-masing rata-rata per kategori. Selanjutnya adalah fungsi yang digunakan untuk mendapatkan varian. Fungsi varian memiliki parameter \$jumlah_data, \$xi, \$x10, \$mean_ganas, \$mean_jinak, \$count_ganas, dan \$count_jinak. Sama halnya dengan mencari rata-rata, dalam mencari varian juga didapatkan kategori data dan dihitung menggunakan rumus varian. Fungsi untuk menghitung rata-rata dan varian dapat dilihat pada *source code 4.6*.

```

function mean($jumlah_data, $xi, $x10, $count_ganas,
    $count_jinak){
    $i=0;
    while($i < $jumlah_data){
        if($x10[$i] == 4){
            $data_ganas[$i] = $xi[$i];
            $jumlah_ganas = array_sum($data_ganas);
            $mean_ganas = $jumlah_ganas/$count_ganas;
        }
        else if($x10[$i] == 2){
            $data_jinak[$i] = $xi[$i];
            $jumlah_jinak = array_sum($data_jinak);
            $mean_jinak = $jumlah_jinak/$count_jinak;
        }
        $i++;
    }
}

```

```

        $i++;}
        echo "MEAN GANAS : ".$mean_ganas."<br>";
        echo "MEAN JINAK : ".$mean_jinak."<br>";
        return array($mean_ganas, $mean_jinak);}

        function varian($jumlah_data, $xi, $x10, $mean_ganas,
        $mean_jinak, $count_ganas, $count_jinak){
            $i=0;
            while($i < $jumlah_data){
                if($x10[$i] == 4){
                    $data_ganas[$i] = $xi[$i];
                    $a_ganas[$i] = ($data_ganas[$i] - $mean_ganas);
                    $var_kuadrat_ganas = $var_kuadrat_ganas +
                    ($a_ganas[$i]*$a_ganas[$i]);
                    $i++; }
                else if($x10[$i] == 2){
                    $data_jinak[$i] = $xi[$i];
                    $a_jinak[$i] = ($data_jinak[$i] - $mean_jinak);
                    $var_kuadrat_jinak = $var_kuadrat_jinak +
                    ($a_jinak[$i]*$a_jinak[$i]);
                    $i++; } }
            $var_ganas = $var_kuadrat_ganas/($count_ganas-1);
            $var_jinak = $var_kuadrat_jinak/($count_jinak-1);
            return array ($var_ganas, $var_jinak);
        }

```

Source code 4.6 Proses mengitung rata-rata dan varian Naïve Bayes

4.2.7 Proses perhitungan menggunakan distribusi gauss

Distribusi Gaussian sering dipilih untuk merepresentasikan peluang kelas bersyaarat untuk atribut kontinyu. Distribusi dikarakterisasi dengan dua parameter yaitu rata-rata dan varian. Dalam proses perhitungan distribusi gauss digunakan parameter \$jumlah_data, \$xi, \$x10, \$mean, \$mean_jinak, \$mean_ganas, \$var_jinak, \$var_ganas, dan \$kolom_mungkin. Langkah awal didapatkan standar deviasi untuk ganas dan jinak. Lalu menghitung dengan rumus distribusi gauss untuk mendapatkan peluang kelas dari masing-masing atribut. Fungsi untuk menghitung distribusi gauss dapat dilihat pada *source code 4.7*.

```

function gaussian($jumlah_data, $xi, $x10, $mean_ganas,
        $mean_jinak, $var_ganas, $var_jinak, $kolom_mungkin){
            $stdev_ganas = sqrt($var_ganas);
            $stdev_jinak = sqrt($var_jinak);
            $col_mungkin = $kolom_mungkin;
            solid'>x". ($col_mungkin+1). "</td>";

```

```

for($i = 0; $i<$jumlah_data; $i++){
    for($j = 0; $j<$jumlah_data; $j++){
        $arr_mungkin[$i] = $xi[$i][$j];
        $string = $arr_mungkin[$i];
        $arr_hasil = explode(",", $string);
        if($arr_hasil[$i] != ""){
            $data_mungkin[$i][$j] = $arr_hasil[$col_mungkin];
            $p_ganas[$i][$j] =
(1/($stdev_ganas*(sqrt(2*3.14))))*(exp(-1*((( $data_mungkin[$i][$j]
- $mean_ganas)*($data_mungkin[$i][$j] -
$mean_ganas))/(2*($stdev_ganas*$stdev_ganas)))));
            $p_jinak[$i][$j] =
(1/($stdev_jinak*(sqrt(2*3.14))))*(exp(-1*((( $data_mungkin[$i][$j]
- $mean_jinak)*($data_mungkin[$i][$j] -
$mean_jinak))/(2*($stdev_jinak*$stdev_jinak))))); } } }
return array($p_ganas, $p_jinak);
}

```

Source code 4.7 Proses perhitungan menggunakan distribusi gauss

4.2.8 Proses Naïve Bayes

Proses Naïve Bayes memiliki parameter antara lain, \$jumlah_data, \$x1, \$x2, \$x3, \$x4, \$x5, \$x6, \$x7, \$x8, \$x9, \$x10, \$mungkin, dan \$hasil_all. Proses Naïve Bayes membutuhkan nilai rata-rata, varian dan distribusi gauss yang telah didapatkan pada fungsi sebelumnya. Dalam mengambil data, dibutuhkan perulangan untuk mendapatkan kelas daripada data. Disini fungsi proses Naïve Bayes menggunakan perulangan *while*. Setelah mendapatkan kelas dari masing-masing data, dilakukan perhitungan untuk *posterior probability*. Dalam menghitung *posterior probability* dilakukan perkalian untuk semua parameter atribut berdasarkan kelasnya. Selanjutnya diperoleh nilai maksimum untuk menentukan kelas dari setiap data “mungkin”. Fungsi proses Naïve Bayes dapat dilihat pada *source code 4.8*.

```

function prosesNaiveBayes($jumlah_data, $jumlah_data_uji, $x1,
$x2, $x3, $x4, $x5, $x6, $x7, $x8, $x9, $x10, $x10_aktual,
$mungkin, $hasil_all){
    $i = 0;
    $count_ganas=0;
    $count_jinak=0;
    while($i < $jumlah_data){
        if($x10[$i] == 4){
            $count_ganas++;}
    }
}

```

```

        if($x10[$i] == 2){
            $count_jinak++;}
        $i++;
    }
    list($mean_ganas_x1,          $mean_jinak_x1)          =
self::mean($jumlah_data, $x1, $x10, $count_ganas, $count_jinak);
    list($mean_ganas_x2,          $mean_jinak_x2)          =
self::mean($jumlah_data, $x2, $x10, $count_ganas, $count_jinak);
    list($mean_ganas_x3,          $mean_jinak_x3)          =
self::mean($jumlah_data, $x3, $x10, $count_ganas, $count_jinak);
    list($mean_ganas_x4,          $mean_jinak_x4)          =
self::mean($jumlah_data, $x4, $x10, $count_ganas, $count_jinak);
    list($mean_ganas_x5,          $mean_jinak_x5)          =
self::mean($jumlah_data, $x5, $x10, $count_ganas, $count_jinak);
    list($mean_ganas_x6,          $mean_jinak_x6)          =
self::mean($jumlah_data, $x6, $x10, $count_ganas, $count_jinak);
    list($mean_ganas_x7,          $mean_jinak_x7)          =
self::mean($jumlah_data, $x7, $x10, $count_ganas, $count_jinak);
    list($mean_ganas_x8,          $mean_jinak_x8)          =
self::mean($jumlah_data, $x8, $x10, $count_ganas, $count_jinak);
    list($mean_ganas_x9,          $mean_jinak_x9)          =
self::mean($jumlah_data, $x9, $x10, $count_ganas, $count_jinak);
    list($var_ganas_x1,           $var_jinak_x1)            =
self::varian($jumlah_data,      $x1,      $x10,      $mean_ganas_x1,
$mean_jinak_x1, $count_ganas, $count_jinak);
    list($var_ganas_x2,           $var_jinak_x2)            =
self::varian($jumlah_data,      $x2,      $x10,      $mean_ganas_x2,
$mean_jinak_x2, $count_ganas, $count_jinak);
    list($var_ganas_x3,           $var_jinak_x3)            =
self::varian($jumlah_data,      $x3,      $x10,      $mean_ganas_x3,
$mean_jinak_x3, $count_ganas, $count_jinak);
    list($var_ganas_x4,           $var_jinak_x4)            =
self::varian($jumlah_data,      $x4,      $x10,      $mean_ganas_x4,
$mean_jinak_x4, $count_ganas, $count_jinak);
    list($var_ganas_x5,           $var_jinak_x5)            =
self::varian($jumlah_data,      $x5,      $x10,      $mean_ganas_x5,
$mean_jinak_x5, $count_ganas, $count_jinak);
    list($var_ganas_x6,           $var_jinak_x6)            =
self::varian($jumlah_data,      $x6,      $x10,      $mean_ganas_x6,
$mean_jinak_x6, $count_ganas, $count_jinak);
    list($var_ganas_x7,           $var_jinak_x7)            =
self::varian($jumlah_data,      $x7,      $x10,      $mean_ganas_x7,
$mean_jinak_x7, $count_ganas, $count_jinak);
    list($var_ganas_x8,           $var_jinak_x8)            =
self::varian($jumlah_data,      $x8,      $x10,      $mean_ganas_x8,
$mean_jinak_x8, $count_ganas, $count_jinak);
    list($var_ganas_x9,           $var_jinak_x9)            =
self::varian($jumlah_data,      $x9,      $x10,      $mean_ganas_x9,
$mean_jinak_x9, $count_ganas, $count_jinak);
    echo "<table class='table-list' style='border:1px
solid'>";

```

```

        list($p_ganas_x1,                $p_jinak_x1)                =
self::gaussian($jumlah_data_uji, $mungkin, $x10, $mean_ganas_x1,
$mean_jinak_x1, $var_ganas_x1, $var_jinak_x1, 0);
        list($p_ganas_x2,                $p_jinak_x2)                =
self::gaussian($jumlah_data_uji, $mungkin, $x10, $mean_ganas_x2,
$mean_jinak_x2, $var_ganas_x2, $var_jinak_x2, 1);
        list($p_ganas_x3,                $p_jinak_x3)                =
self::gaussian($jumlah_data_uji, $mungkin, $x10, $mean_ganas_x3,
$mean_jinak_x3, $var_ganas_x3, $var_jinak_x3, 2);
        list($p_ganas_x4,                $p_jinak_x4)                =
self::gaussian($jumlah_data_uji, $mungkin, $x10, $mean_ganas_x4,
$mean_jinak_x4, $var_ganas_x4, $var_jinak_x4, 3);
        list($p_ganas_x5,                $p_jinak_x5)                =
self::gaussian($jumlah_data_uji, $mungkin, $x10, $mean_ganas_x5,
$mean_jinak_x5, $var_ganas_x5, $var_jinak_x5, 4);
        list($p_ganas_x6,                $p_jinak_x6)                =
self::gaussian($jumlah_data_uji, $mungkin, $x10, $mean_ganas_x6,
$mean_jinak_x6, $var_ganas_x6, $var_jinak_x6, 5);
        list($p_ganas_x7,                $p_jinak_x7)                =
self::gaussian($jumlah_data_uji, $mungkin, $x10, $mean_ganas_x7,
$mean_jinak_x7, $var_ganas_x7, $var_jinak_x7, 6);
        list($p_ganas_x8,                $p_jinak_x8)                =
self::gaussian($jumlah_data_uji, $mungkin, $x10, $mean_ganas_x8,
$mean_jinak_x8, $var_ganas_x8, $var_jinak_x8, 7);
        list($p_ganas_x9,                $p_jinak_x9)                =
self::gaussian($jumlah_data_uji, $mungkin, $x10, $mean_ganas_x9,
$mean_jinak_x9, $var_ganas_x9, $var_jinak_x9, 8);
        echo "</table>";echo "<br>";
        echo "<b>HASIL PREDIKSI DATA MUNGKIN<b/>";
        echo " <table class='table-list' style='border:1px
solid'>";echo "<tr>";
        echo "<td style='border:1px solid;'>Data ke-</td>";
        echo "<td style='border:1px solid;'>Array Data</td>";
        echo "<td style='border:1px solid;'>Peluang GANAS</td>";
        echo "<td style='border:1px solid;'>Peluang JINAK</td>";
        echo "<td style='border:1px solid;'>Hasil Bayesian</td>";
echo "<td style='border:1px solid;'>Kelas Hasil Prediksi</td>";
        echo "</tr>";
        for($i = 0; $i<$jumlah_data_uji; $i++){
            for($j = 0; $j<$jumlah_data_uji; $j++){
                if($p_ganas_x1[$i][$j] != ""){
                    $kelas_ganas[$i][$j]
($p_ganas_x1[$i][$j] * $p_ganas_x2[$i][$j] * $p_ganas_x3[$i][$j] *
$p_ganas_x4[$i][$j] * $p_ganas_x5[$i][$j] * $p_ganas_x6[$i][$j] *
$p_ganas_x7[$i][$j] * $p_ganas_x8[$i][$j] * $p_ganas_x9[$i][$j]) *
($count_ganas/$jumlah_data);
                    $kelas_jinak[$i][$j]
($p_jinak_x1[$i][$j] * $p_jinak_x2[$i][$j] * $p_jinak_x3[$i][$j] *
$p_jinak_x4[$i][$j] * $p_jinak_x5[$i][$j] * $p_jinak_x6[$i][$j] *
$p_jinak_x7[$i][$j] * $p_jinak_x8[$i][$j] * $p_jinak_x9[$i][$j]) *

```

```

($count_jinak/$jumlah_data);
$kelas_max[$i][$j]=max($kelas_ganas[$i][$j],$kelas_jinak[$i][$j]);
echo "<tr>";
echo "<td style='border:1px solid;'>".($j+1)."</td>";
echo "<td style='border:1px solid;'>".$mungkin[$i][$j]."</td>";
echo "        <td style='border:1px solid; background-color:#f20000;'>".$kelas_ganas[$i][$j]."</td>";
echo "        <td style='border:1px solid; background-color:#918f9a;'>".$kelas_jinak[$i][$j]."</td>";
echo "<td style='border:1px solid;'>".$kelas_max[$i][$j]."</td>";
if($kelas_max[$i][$j] == $kelas_ganas[$i][$j]){
$nama_kelas_predik[$i][$j] = "GANAS";
echo "        <td style='border:1px solid;'>".$nama_kelas_predik[$i][$j]."</td>"; }
        else{ $nama_kelas_predik[$i][$j] = "JINAK";
echo "        <td style='border:1px solid;'>".$nama_kelas_predik[$i][$j]."</td>"; }
        echo "</tr>";
        }}}
        echo "</table>";echo "<br>";echo "<br>";
}

```

Source code 4.8 Proses Naïve Bayes

4.2.9 Proses Pelatihan KMNB (K-Means dan Naïve Bayes)

Parameter untuk fungsi pelatihan adalah \$data_latih dan \$k, dimana \$data_latih adalah inputan yang telah diberikan user dan \$k adalah jumlah *cluster* yang akan dibentuk. Pada fungsi ini akan menyertakan file KMNB.php. Hal ini dilakukan untuk pemanggilan fungsi-fungsi yang terdapat pada KMNB.php. Fungsi pelatihan dapat dilihat pada *source code* 4.9.

```

function pelatihan($data_latih, $k){

    echo "<h3>Data Input</h3>";
    echo "<div style='border: 1px solid rgb(204, 204, 204); padding: 5px; overflow: auto; width: 900px; height: 380px; background-color: rgb(255, 255, 255)'>";

        $xi = get2DArray_Input($data_latih, ",");
        echo "</div><br>";
        $t = get1DArray_Target($data_latih, ",");
        $jumlah_data = jumlahData($data_latih, ",");

    echo "<h2>Hasil Prediksi menggunakan algoritma K-Means dan Naive Bayes</h2>";

    echo "<div style='border: 1px solid rgb(204, 204, 204); height: 380px; width: 900px; background-color: rgb(255, 255, 255)'>";

```

```

204); padding: 5px; overflow: auto; width: 900px; height: 800px;
background-color: rgb(255, 255, 255)'>";
    include "KMNB.php";
    echo "</div><br>";
}

```

Source code 4.9 Proses Pelatihan KMNB

4.2.10 Proses Pengujian KMNB (K-Means dan Naïve Bayes)

Parameter untuk pengujian adalah \$data_uji dan \$k. Fungsi pengujian menyertakan file KMNB_2.php. Hal ini dilakukan untuk pemanggilan fungsi-fungsi yang terdapat pada KMNB_2.php. Fungsi pengujian dapat dilihat pada *source code 4.10*.

```

function pengujian($data_uji, $k){
    echo "<h2>Data Uji</h2>";
    echo "<div style='border: 1px solid rgb(204, 204, 204);
padding: 5px; overflow: auto; width: 900px; height: 380px;
background-color: rgb(255, 255, 255)'>";
    $xi_uji = get2DArray_Input($data_uji, ",");
    echo "</div><br>";
    $t = get1DArray_Target($data_uji, ',');
    $jumlah_data_uji = jumlahData($data_uji, ",");
    $fip = fopen('data/temp_nama_data_latih.csv', 'r');
    $nama_data_latih =
get1DArray('data/temp_nama_data_latih.csv', ',');
    $nama_file = $nama_data_latih[0];
    if($nama_file == "d_latih_30%.csv"){
        $jumlah_data_latih =
jumlahData('data/data_seimbang/data_latih/dengan_kelas/d_latih_30%
.csv', ',');
    }
    else if($nama_file == "d_latih_40%.csv"){
        $jumlah_data_latih =
jumlahData('data/data_seimbang/data_latih/dengan_kelas/d_latih_40%
.csv', ',');
    }
    else if($nama_file == "d_latih_50%.csv"){
        $jumlah_data_latih =

```

```

jumlahData('data/data_seimbang/data_latih/dengan_kelas/d_latih_50%
.csv', ',');}
    else if($nama_file == "d_latih_60%.csv"){
        $jumlah_data_latih
jumlahData('data/data_seimbang/data_latih/dengan_kelas/d_latih_60%
.csv', ',');}
    else if($nama_file == "d_latih_70%.csv"){
        $jumlah_data_latih
jumlahData('data/data_seimbang/data_latih/dengan_kelas/d_latih_70%
.csv', ',');}
    else if($nama_file == "d_latih_80%.csv"){
        $jumlah_data_latih
jumlahData('data/data_seimbang/data_latih/dengan_kelas/d_latih_80%
.csv', ',');}
    else{ $jumlah_data_latih = 0;
        echo "Jumlah Data = 0 !!<br>";}
    echo "<h2>Hasil Prediksi menggunakan algoritma K-Means dan
Naive Bayes</h2>";
    echo "<div style='border: 1px solid rgb(204, 204, 204);
padding: 5px; overflow: auto; width: 900px; height: 800px;
background-color: rgb(255, 255, 255)'>";
        include "KMNB_2.php";
        echo "</div><br>";
    }

```

Source code 4.10 Proses Pengujian KMNB

4.3 Implementasi Antarmuka

4.3.1 Antarmuka Utama

Antarmuka utama terdiri dari form utama yang berisi menu Home, Deskripsi sistem, Tentang K-Means dan Naïve Bayes, Galeri Kanker Payudara, dan Aplikasi. Pada umumnya tampilan utama hanya berisi informasi singkat mengenai penelitian yang dituang ke dalam aplikasi. Halaman utama dapat dilihat pada gambar 4.1.



Gambar 4. 1 Halaman utama

4.3.2 Antarmuka Aplikasi

4.3.2.1 Antarmuka Pelatihan KMNB (K-Means dan Naïve Bayes)

Dalam antarmuka pelatihan KMNB terdapat tombol untuk memilih file yang akan digunakan sebagai data latih. Data latih yang digunakan dalam aplikasi ini adalah file yang memiliki format .csv. Setelah menentukan file yang akan dijadikan data latih, *user* memilih tombol latih KMNB guna memulai proses pelatihan dengan menggunakan algoritma K-Means dan Naïve Bayes. Antarmuka pelatihan KMNB dapat dilihat pada gambar 4.2.



Gambar 4.2. Antarmuka untuk memasukkan data latih (KMNB)

Tahap selanjutnya dalam proses pelatihan yaitu menampilkan isi data yang dimiliki oleh file yang telah dipilih sebelumnya. Setelah itu didapatkan nilai *centroid* dari perhitungan K-Means dan nilai rata-rata dari perhitungan Naïve Bayes. Pada antarmuka ini ditampilkan data inputan (data latih) beserta atributnya. Selain itu juga ditampilkan hasil dari pemrosesan menggunakan algoritma K-Means dan Naïve Bayes. Informasi yang dibutuhkan untuk pengujian akan otomatis disimpan ke dalam file yang telah ditentukan sebelumnya. Tampilan antarmuka untuk proses pelatihan menggunakan algoritma K-Means dan Naïve Bayes dapat dilihat pada gambar 4.3.



Klasifikasi Kanker Payudara

[Pelatihan KMNB](#)
[Pengujian KMNB](#)
[Pelatihan K-Means](#)
[Pengujian K-Means](#)
[Keluar Aplikasi](#)

Pelatihan K-Means - Naive Bayes

Pilih Data Pelatihan

Telusuri...
 Tidak ada berkas dipilih.
 :
 [Latih KMNB](#)

Data Input

Data	Ketebalan gumpalan tumor(1)	Keseragaman ukuran(2)	Keseragaman bentuksel(3)	Kelelantan pinggiran sel(4)	Ukuran sel tiap jaringan epitel(5)	Kelosongan pada inti sel(6)	Kromatin lunak(7)	Inti sel normal(8)	Pem belahan mitosis(9)
1	5	1	1	1	2	1	3	1	1
2	3	1	1	1	2	2	3	1	1
3	4	1	1	3	2	1	3	1	1
4	1	1	1	1	2	10	3	1	1
5	2	1	2	1	2	1	3	1	1
6	2	1	1	1	2	1	1	1	5
7	4	2	1	1	2	1	2	1	1
8	1	1	1	1	1	1	3	1	1
9	2	1	1	1	2	1	2	1	1
10	1	1	1	1	2	3	3	1	1
11	4	1	1	1	2	1	2	1	1
12	4	1	1	1	2	1	3	1	1
13	6	1	1	1	2	1	3	1	1
14	3	1	1	1	2	1	2	1	1

Hasil Prediksi menggunakan algoritma K-Means dan Naive Bayes

Centroid	Centroid Ganas	Centroid Mungkin	Centroid Jinak
1	7.65	7.2698412698413	2.8510638297872
2	8.65	4.3492063492063	1.3262411347518
3	8.5	4.9365079365079	1.4184397163121
4	6.3	4.5396825396825	1.2553191489362
5	7.1333333333333	4.3650793650794	2.1205673758865
6	7.6666666666667	8.5714285714286	1.4468085106383
7	6.3166666666667	4.8888888888889	2.6312056737589
8	7.5666666666667	5.0952380952381	1.3120567375887
9	3.8	2.1587301587302	1.177304964539

HASIL PREDIKSI DATA DENGAN K-MEANS

No.	Array Data	Anggota Kluster
1	5,1,1,1,2,1,3,1,1	JINAK
2	3,1,1,1,2,2,3,1,1	JINAK
3	4,1,1,3,2,1,3,1,1	JINAK
4	1,1,1,1,2,10,3,1,1	JINAK
5	2,1,2,1,2,1,3,1,1	JINAK
6	2,1,1,1,2,1,1,1,5	JINAK
7	4,2,1,1,2,1,2,1,1	JINAK
8	1,1,1,1,1,1,3,1,1	JINAK
9	2,1,1,1,2,1,2,1,1	JINAK
10	1,1,1,1,2,3,3,1,1	JINAK
11	4,1,1,1,2,1,2,1,1	JINAK
12	4,1,1,1,2,1,3,1,1	JINAK
13	6,1,1,1,2,1,3,1,1	JINAK
14	3,1,1,1,2,1,2,1,1	JINAK
15	1,1,1,1,1,2,1,3,1,1	JINAK
16	3,2,1,1,1,1,2,1,1	JINAK
17	5,1,1,1,2,1,2,1,1	JINAK
18	2,1,1,1,2,1,2,1,1	JINAK
19	1,1,3,1,2,1,1,1,1	JINAK
20	3,1,1,1,1,1,2,1,1	JINAK
21	2,1,1,1,2,1,3,1,1	JINAK

Jumlah data = 264
 Data pelatihan telah tersimpan !
 Klik menu Pengujian KMNB untuk melakukan pengujian.

Anggreni
 NIM. 105090606111003
 Teknik Informatika / Ilmu Komputer
 Program Teknologi Informasi dan Ilmu Komputer

[Home](#)
[Deskripsi Sistem](#)
[Tentang K-Means dan Naive Bayes](#)
[Apliasi](#)

Gambar 4.3 Antarmuka pelatihan K-Means dan Naïve Bayes

4.3.2.2 Antarmuka Pengujian KMNB (K-Means dan Naïve Bayes)

Pada antarmuka pengujian KMNB terdapat tombol untuk memilih file yang akan digunakan sebagai data uji. Data uji yang digunakan dalam aplikasi ini adalah file yang memiliki format .csv. Setelah menentukan file yang akan dijadikan data uji, *user* memilih tombol uji KMNB guna memulai proses pelatihan dengan menggunakan algoritma K-Means dan Naïve Bayes. Antarmuka pengujian KMNB dapat dilihat pada gambar 4.2.



Gambar 4.4 Antarmuka untuk memasukkan data latih (KMNB)

Tahap selanjutnya dalam proses pengujian yaitu menampilkan isi data yang dimiliki oleh file yang telah dipilih sebagai data uji sebelumnya. Setelah itu ditampilkan hasil prediksi yang dilakukan oleh K-Means. Hasil prediksi K-Means ditampilkan dalam sebuah tabel. Prediksi K-Means akan menghasilkan data “mungkin” yang akan diproses dengan menggunakan algoritma Naïve Bayes. Dari pemrosesan Naïve Bayes dihasilkan data “mungkin” yang terkelompok. Selain itu juga ditampilkan hasil akhir seluruh data yang telah dikelompokkan. Untuk memperoleh tingkat akurasi dari hasil perhitungan dengan menggunakan algoritma K-Means dan Naïve Bayes digunakan rumus akurasi. Tampilan antarmuka untuk proses pelatihan menggunakan algoritma K-Means dan Naïve Bayes dapat dilihat pada gambar 4.5.

Klasifikasi Kanker Payudara

[Pelatihan KMNb](#) [Pengujian KMNb](#) [Pelatihan K-Means](#) [Pengujian K-Means](#) [Keluar Aplikasi](#)

Pengujian KMNb

Pilih Data Uji

Telusuri...

:

Uji KMNb

Data Uji

Data No	Ketebalan gumpalan tumor(1)	Keseragaman ukuran(2)	Keseragaman bentuk sel(3)	Ketebalan pinggiran sel(4)	Ukuran sel tiap saringan epitel(5)	Kolesongan pada inti sel(6)	Kromatin lunak(7)	Inti sel normal(8)	Pembelahan mitosis(9)
1	5	1	4	3	2	1	3	2	1
2	1	1	1	1	2	1	3	1	1
3	5	1	1	1	2	1	3	1	1
4	3	1	1	1	2	1	3	2	1
5	3	1	1	1	2	1	2	1	1
6	1	1	1	1	2	1	2	1	1
7	1	1	1	1	2	1	3	1	1
8	3	1	1	1	2	1	3	1	1
9	2	1	1	2	2	1	3	1	1
10	3	1	1	1	3	1	2	1	1
11	1	1	1	1	2	1	1	1	1
12	1	1	1	1	2	1	3	1	1
13	8	2	1	1	5	1	1	1	1

d_lath_60%.csv

Hasil Prediksi menggunakan algoritma K-Means dan Naive Bayes

CENTROID

Atribut	Centroid Ganas	Centroid Mungkin	Centroid Jinak
1	7.65	7.2698	2.8511
2	8.65	4.3492	1.3262
3	8.5	4.9165	1.4184
4	6.3	4.5397	1.2553
5	7.1333	4.3651	2.1206
6	7.6667	8.5714	1.4468
7	6.3167	4.8889	2.6312
8	7.5667	5.0952	1.3121
9	3.8	2.1587	1.1773

HASIL PREDIKSI DATA DENGAN K-MEANS

No	Array Data	Anggota Kluster
1	5,1,4,3,2,1,3,2,1	MUNGKIN
2	1,1,1,1,2,1,3,1,1	JINAK
3	5,1,1,1,2,1,3,1,1	JINAK
4	3,1,1,1,2,1,3,2,1	JINAK
5	3,1,1,1,2,1,2,1,1	JINAK
6	1,1,1,1,2,1,2,1,1	JINAK
7	1,1,1,1,2,1,3,1,1	JINAK
8	3,1,1,1,2,1,3,1,1	JINAK
9	2,1,1,2,2,1,3,1,1	JINAK
10	3,1,1,1,3,1,2,1,1	JINAK
11	1,1,1,1,2,1,1,1,1	JINAK
12	1,1,1,1,2,1,3,1,1	JINAK
13	8,2,1,1,5,1,1,1,1	JINAK
14	1,1,1,1,2,1,3,1,1	JINAK
15	1,1,1,1,2,1,3,1,1	JINAK
16	1,1,1,1,2,1,3,1,1	JINAK
17	1,1,1,1,2,1,3,1,1	JINAK
18	3,1,1,1,2,5,5,1,1	JINAK
19	2,1,1,1,3,1,2,1,1	JINAK
20	1,1,1,1,2,1,1,1,1	JINAK
21	1,1,1,1,2,1,1,1,1	JINAK

HASIL

Setelah Proses Clustering dengan K-Means

Jumlah Cluster GANAS : 28

Jumlah Cluster JINAK : 83

Jumlah Cluster MUNGKIN : 65

Setelah Proses Klasifikasi dengan Naive Bayes

JUMLAH TRUE GANAS : 28

JUMLAH FALSE GANAS : 0

JUMLAH TRUE JINAK : 79

JUMLAH FALSE JINAK : 4

JUMLAH TRUE MUNGKIN : 62

JUMLAH FALSE MUNGKIN : 3

Hasil Akhir

JUMLAH BENAR PREDIKSI SELURUH DATA : 169

JUMLAH SALAH PREDIKSI SELURUH DATA : 7

AKURASI = 96.02272727272727%

Anggreini

NIM. 105090606111003

Teknik Informatika / Ilmu Komputer

Program Teknologi Informasi dan Ilmu Komputer

2015

[Home](#) [Deskripsi Sistem](#) [Tentang K-Means dan Naive Bayes](#) [Aplikasi](#)

Gambar 4.5 Antarmuka pengujian algoritma K-Means dan Naive Bayes