

Implementasi Metode Clustering K-Means untuk Mengelompokan Baut Pada Toko Ali Baut Berdasarkan Ukuran Diameter, Panjang, dan Pitch

Cep Adi Wiharja¹⁾, Taufik Kurahman²⁾, Wahyu Pebi Romansah³⁾, Sugeng Riyadi⁴⁾

¹⁾ Teknik Komputer, AMIK BSI Jakarta

²⁾ Sistem Informasi, STMIK Nusa Mandiri Jakarta

³⁾ Sistem Informasi, STMIK Nusa Mandiri Jakarta

⁴⁾ Sistem Informasi, STMIK Nusa Mandiri Jakarta

cep.caw@bsi.ac.id, taufikku2302@bsi.ac.id, wahyupeb0802@bsi.ac.id, sugengri2002@bsi.ac.id

Abstrak – Pengelompokan baut merupakan bagian proses dari penjualan baut agar mudah dalam menentukan ukuran baut sesuai dengan kebutuhan, tetapi pengelola toko ali baut terkendala dalam menentukan klasifikasi baut tersebut. Untuk mencapai tujuannya, perusahaan harus bisa mengembangkan proses bisnis secara terkomputerisasi sesuai dengan perkembangan zaman saat ini. Tujuan yang hendak dicapai dalam penelitian ini adalah untuk membuat terobosan baru dalam mengelompokan baut yang dilakukan oleh Toko Ali Baut secara komputerisasi. Dengan metode Clustering K-Means untuk mengelompokan baut berdasarkan ukuran diameter, panjang, dan pitch. Pemilihan *Dreamweaver CS6* dan *Apache*, serta bahasa pemrograman HTML, PHP, Javascript, SQL, dan CSS sebagai perangkat pemrograman dan database yang digunakan dalam pembuatan aplikasi pengelompokan baut berbasis *website*. Kesimpulan dari penelitian adalah perancangan sistem informasi pengelompokan baut ini sebagai solusi terbaik untuk membantu pengelola toko ali baut dalam pengklasterisasian setiap produk baut yang dijualnya.

Kata kunci : Data Mining, Clustering, K-Means, Baut, Ukuran.

1. Pendahuluan

1.a Latar Belakang

Pengelolaan data baut dalam jumlah yang banyak sulit dan memakan waktu yang lama untuk menentukan kelas berdasarkan ukurannya. Hal ini merupakan permasalahan yang ada pada Toko Ali Baut dalam mengelola baut untuk mengkategorikannya.

Baut adalah alat sambung dengan batang bulat dan berulir, salah satu ujungnya dibentuk kepala baut (umumnya bentuk kepala segi enam) dan ujung lainnya dipasang mur/pengunci. Dalam pemakaian di lapangan, baut dapat digunakan untuk membuat konstruksi sambungan tetap, sambungan bergerak, maupun sambungan sementara yang dapat dibongkar/dilepas kembali.

Dari tiap satu jenis baut memiliki spesifikasi ukuran dan bahan yang berbeda, sehingga mempunyai berbagai macam spesifikasi sesuai kebutuhan pengguna.

1.b Rumusan Masalah

Masalah yang ada terhadap pengelolaan baut pada Toko Ali Baut adalah :

1. Seberapa banyak jumlah baut dalam setiap kelasnya ?
2. Bagaimana mengaplikasikan algoritma *K-Means* untuk mengklasterisasi baut berdasarkan ukuran diameter, panjang, pitch ?

1.c Batasan Masalah

Adapun batasan masalahnya :

1. Penentuan ukuran baut yang paling banyak dalam tiap-tiap kelas.
2. Data yang digunakan adalah data ketersediaan baut perbulan.

1.d Tujuan

Tujuan dilakukannya penelitian ini adalah : guna mengetahui sejauhmana *Clustering K-Means* dapat membantu dalam menentukan jumlah baut berdasarkan ukuran diameter, panjang, dan pitch.

1.e Manfaat Penelitian

Adapun manfaat dari penelitian ini yaitu : membantu toko untuk mengetahui jumlah baut yang ada dari setiap spesifikasinya.

1.f Metode Penelitian

Metode penelitian yang penulis gunakan yaitu, metode Algoritma *Clustering K-Means* dengan menganalisa prodak baut yang ada pada Toko Ali Baut.

2. Dasar Teori

2.a Definisi Data Mining

Menurut Pregibon (dalam Prasetyo 2014) "Data mining adalah campuran dari statistik, kecerdasan buatan, dan riset basis data yang masih berkembang".

Tujuan dari teknik data mining adalah berusaha mencari manfaat dari sekumpulan data tersebut. Dilihat dari disiplin ilmu yang digunakan, data mining merupakan sebuah ilmu multi disiplin yang menyangkut berbagai disiplin ilmu seperti *database*, *artificial intelligence* (kecerdasan buatan), *information science* (ilmu informasi), *high performance computing*, visualisasi, *machine learning*, statistik, *neural networks* (jaringan syaraf tiruan), pemodelan matematika, *information retrieval* dan *information extraction* serta pengenalan pada sebuah pola. Saat ini data mining juga berkembang menjadi salah satu dari berbagai konsep disiplin ilmu lain, seperti web mining dan text mining (Dini, 2014).

2.b Definisi Diameter, Panjang, dan Pitch

Diameter (d) adalah garis lurus yang menghubungkan dua titik pada lengkungan lingkaran dan melalui titik pusat. Garis AB pada lingkaran O merupakan diameter lingkaran tersebut. Dengan kata lain, nilai diameter merupakan dua kali nilai jari-jarinya, atau ditulis dengan $d = 2r$.

Panjang adalah dimensi suatu benda yang menyatakan jarak antar ujung. Panjang dapat dibagi menjadi tinggi, yaitu jarak vertikal, serta lebar, yaitu jarak dari satu sisi ke sisi yang lain, diukur pada sudut tegak lurus terhadap panjang benda. Dalam ilmu fisika dan teknik, kata "panjang" biasanya digunakan secara sinonim dengan "jarak", dengan simbol "l" atau "L" (singkatan dari bahasa Inggris *length*).

Pitch adalah jarak dari satu ujung ulir ke ujung ulir berikutnya. Juga dapat diartikan jarak yang ditempuh ulir dalam satu kali putaran.

2.c Definisi Algoritma klasifikasi K-Means

K-Means merupakan algoritma *clustering* yang berulang-ulang. Algoritma K-Means dimulai dengan pemilihan secara acak K, K disini merupakan banyaknya *cluster* yang ingin dibentuk. Kemudian tetapkan nilai-nilai K secara random, untuk sementara nilai tersebut menjadi pusat dari *cluster* atau biasa disebut dengan centroid, mean atau "means". Hitung jarak setiap data yang ada terhadap masing-masing centroid menggunakan rumus Euclidian hingga ditemukan jarak yang paling dekat dari setiap data dengan centroid. Klasifikasikan setiap data berdasarkan kedekatannya dengan centroid. Lakukan langkah tersebut hingga nilai centroid tidak berubah (stabil

3. Pembahasan

3.a Data Pengujian

Contoh data awal sebelum dilakukan perhitungan untuk pengelompokan baut dan

ukuran baut dari masing-masing data dapat dilihat pada Tabel 1.1

Tabel 1.1 Data Ukuran Baut

No	Diameter (MM)	Panjang Baut (MM)	Pitch (MM)
1	3	10	1
2	3	20	1
3	3	30	1
4	4	40	1
5	4	50	1
6	12	40	1.25
7	12	45	1.25
8	12	50	1.25
9	14	30	1.25
10	14	50	1.25
11	16	15	1.25
12	16	20	1.25
13	13	5	1.25
14	13	25	1.25
15	8	40	1
16	8	35	1
17	8	35	1
18	10	20	1
19	10	15	1
20	10	15	1

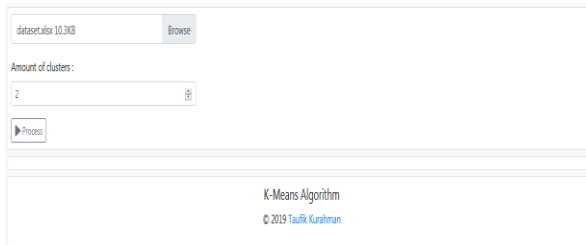
Data pada Tabel 3.1 kemudian digunakan untuk menghitung penentuan klasterifikasi ukuran baut. Hasil dari perhitungan manual dapat dilihat pada Gambar 3.2.

	Cluster_1	Cluster_2	Cluster_1	Cluster_2	Cluster_1	Cluster_2	Cluster_1	Cluster_2
	0	10	7.666159404	26.12951	8.70485	29.84663	9.632122	31.04017
	10	0	10.89816498	16.6158	8.70485	20.18633	8.399735	21.31669
	20	10	19.46201428	8.33175	16.60646	11.14246	15.7586	12.05448
30.01666204	20.02498439	28.72229099	7.138024	25.73955	5.399492	24.74312	5.021904	
40.01249805	30.01666204	38.53530848	15.62961	35.53202	12.07564	34.49638	10.7763	
31.32191725	21.93313703	28.04607816	5.34374	25.0707	2.796637	23.93988	3.037303	
36.13948671	26.57183659	33.03910562	10.0111	30.05894	6.415178	28.93107	5.447913	
41.00076219	31.32191725	38.03396508	14.89594	35.05054	11.15741	33.92486	10.00671	
22.82679347	14.8681707	18.3570831	7.044304	15.49323	10.28694	14.33745	11.58948	
41.4856903	31.9540686	38.17044013	15.37169	35.2142	11.79638	34.07518	10.77656	
13.93063172	13.93063172	6.354722653	21.37808	5.877077	25.06966	5.667279	26.39986	
16.4031247	13.00240362	9.76639647	16.69398	7.716219	20.29338	6.782842	21.6196	
11.18313462	18.02948973	7.468768305	30.5465	10.40625	34.36308	11.40206	35.67973	
18.02948973	11.18313462	13.25829929	10.94329	10.40625	14.63402	9.25	15.96438	
30.41381265	20.61552813	28.10284683	4.87354	25.09033	1.577291	24.01388	1.104321	
25.49509757	15.8113883	23.12509459	1.443857	20.11279	4.376587	19.04673	5.546619	
25.49509757	15.8113883	23.12509459	1.443857	20.11279	4.376587	19.04673	5.546619	
12.20655562	7	8.010617954	15.34551	5.002441	19.17866	3.91578	20.47929	
8.602325267	8.602325267	9.028200786	20.34252	0.15625	24.17618	1.20185	25.47443	
8.602325267	8.602325267	9.028200786	20.34252	0.15625	24.17618	1.20185	25.47443	
New Centroid								
Cluster_1	10.4	12	1.1					
Cluster_2	9.4	35.33333333	1.116667					
Cluster_1	10.125	15	1.09375					
Cluster_2	9.333333333	39.16666667	1.125					
Cluster_1	10.44444444	16.11111111	1.111111					
Cluster_2	9	40.45454545	1.113636					

Gambar 3.2. Hasil Perhitungan Manual

3.b Rancangan Program

Program membutuhkan input data ukuran baut dalam format excel, outputnya berupa table yang berisi datasheet beserta centroid tiap kluster, adapun rancangan programnya sebagai berikut :



Gambar 3.3 Tampilan Input Data pada Program

Rancangan program terdiri atas 3 *button*, 1 elemen input excell, 1 elemen input jumlah kluster, dan 1 *button process*. Aturan cara kerjanya yaitu:

- Masukkan file excel pada tombol browse.
- Masukan jumlah kluster yang akan ditentukan.
- Mengklik tombol *Process* untuk melakukan perhitungan secara otomatis dan akan muncul hasil dari perhitungan tersebut.

3.c Pengkodean Program

Adapaun kode program sebagai berikut :

```
<?php
require_once('vendor/autoload.php');

$target_dir = "dataset/";
$target_file = $target_dir . basename($_FILES["file_import"]["name"]);
$file_type = pathinfo($target_file, PATHINFO_EXTENSION);
$new_name = $target_dir . "dataset." . $file_type;

if (move_uploaded_file($_FILES["file_import"]["tmp_name"], $new_name)) {
    $dataset = array();
    $response["table"] = "";

    $input_file_name = 'dataset/dataset.xlsx';

    $input_file_type = \PhpOffice\PhpSpreadsheet\IOFactory::identify($input_file_name);
    $reader = \PhpOffice\PhpSpreadsheet\IOFactory::createReader($input_file_type);
    $reader->setReadDataOnly(true);
    $spreadsheet = $reader->load($input_file_name);

    $worksheet = $spreadsheet->getActiveSheet();

    $highest_column = $spreadsheet->getActiveSheetIndex(0)->getHighestColumn();

    $highest_column_index =
    \PhpOffice\PhpSpreadsheet\Cell\Coordinate::columnIndexFromString($highest_column);

    $response["table"] = '<div class="table-responsive" id="main_table_div"><table class="table table-
    striped table-bordered" style="width:100%; border-collapse: collapse;">';
    $row_index = 0;
    foreach ($worksheet->getRowIterator() as $row) {
        $response["table"] .= '<tr id="row_' . $row_index++ . '">';
        $cell_iterator = $row->getCellIterator();
        $cell_iterator->setIterateOnlyExistingCells(FALSE); // This loops through all cells,
        foreach ($cell_iterator as $cell) {
            $dataset[] = $cell->getValue();
            if (is_string($cell->getValue())) {
                $response["table"] .=
                '<th class="text-center">';
                $cell->getValue();
                '</th>';
            }
            elseif (is_numeric($cell->getValue())) {
                $response["table"] .=
                '<td class="text-right">';
                $cell->getValue();
                '</td>';
            }
        }
        $response["table"] .= '</tr>';
    }
    $response["table"] .= '</div>';
}

$dataset = array_values($dataset);

$num_cols = $highest_column_index;
$num_rows = count($dataset);

$smax_index = $num_rows - $num_cols;

for ($i=0; $i <= $smax_index; $i++) {
    $dataset[$i] = array();
}

for ($i=1; $i <= $num_cols - 1; $i++) {
    $dataset[$i] = array();
}

$dataset = array_values($dataset);

$rows = array();
$data_attributes = $num_cols - 1;

$rows = array_chunk($dataset, $data_attributes);
$num_rows = count($rows);

$clusters_amount = $_POST['clusters_amount'];

$associative_cluster_centroid = array();

for ($i=0; $i < $clusters_amount; $i++) {
    $rand_val = array();
    for ($j=0; $j < $data_attributes; $j++) {
        $rand_val[] = $rows[$i][$j];
    }
    array_push($associative_cluster_centroid, $rand_val);
}

$finish = false;
$data_init = process($num_rows, $clusters_amount, $rows, $associative_cluster_centroid,
$data_attributes);
$data = process($num_rows, $clusters_amount, $rows,
$data_init['new_associative_cluster_centroid'], $data_attributes);

while ($finish == false) {
    $data = process($num_rows, $clusters_amount, $rows, $data['new_associative_cluster_centroid'],
$data_attributes);
    if ($data['data_index'] == $data['data_index']) {
        $finish = true;
    }
}

$response['clusters_amount'] = $data['clusters_amount'];
$response['num_rows'] = $num_rows;
$response['cluster_centroid'] = $data['cluster_centroid'];
$response['min_val'] = $data['min_val'];
$response['data_index'] = $data['data_index'];
$response['cluster_elements'] = $data['cluster_elements'];
$response['clustered_rows'] = $data['clustered_rows'];
$response['new_centroid'] = $data['new_associative_cluster_centroid'];

echo json_encode($response);
}

function process($num_rows, $clusters_amount, $rows, $associative_cluster_centroid,
$data_attributes)
{
    $centroid = array();
    $cluster_centroid = array();

    for ($i=0; $i < $num_rows; $i++) {
        $sub_array_1 = array();
        for ($j=0; $j < $clusters_amount; $j++) {
            $sub_array_2 = array();
            for ($k=0; $k <= $clusters_amount; $k++) {
                $sub_array_2[] = pow($rows[$i][$k] - $associative_cluster_centroid[$j][$k], 2);
            }
            array_push($sub_array_1, array(
                $sub_array_2
            ));
        }
        array_push($centroid, array(
            $sub_array_1
        ));
    }

    for ($i=0; $i < $num_rows; $i++) {
        $sub_array = array();
        for ($j=0; $j < $clusters_amount; $j++) {
            $sub_array[] = round(sqrt(array_sum($centroid[$i][0][$j][0])), 2);
        }
    }
}
```

```

    }
    array_push($cluster_centroid, $sub_array);
}

$min_val = array();
$data_index = array();

for ($i=0; $i < $num_rows; $i++) {
    $min_val[] = min($cluster_centroid[$i]);
    if (($key = array_search(min($cluster_centroid[$i]), $cluster_centroid[$i]) !== false) {
        $data_index[] = $key;
    }
}

$cluster_elements = array_count_values($data_index);

$cluster_elements_index = array();

for ($i=0; $i < $clusters_amount; $i++) {
    $sub_array = array();
    for ($j=0; $j < $num_rows; $j++) {
        if ($data_index[$j] == $i) {
            $sub_array[] = $j;
        }
    }
    array_push($cluster_elements_index, $sub_array);
}

$clustered_rows = array();

for ($i=0; $i < $clusters_amount; $i++) {
    $sub_array_1 = array();
    for ($j=0; $j < $cluster_elements[$i]; $j++) {
        $sub_array_2 = array();
        for ($k=0; $k < $data_attributes; $k++) {
            $sub_array_2[] = $rows[$cluster_elements_index[$i][$j]][$k];
        }
        array_push($sub_array_1, $sub_array_2);
    }
    array_push($clustered_rows, $sub_array_1);
}

$data_by_cluster = array();

for ($i=0; $i < $clusters_amount; $i++) {
    $sub_array_3 = array();
    for ($x=0; $x < $data_attributes; $x++) {
        $sub_array_1 = array();

        for ($j=0; $j < $cluster_elements[$i]; $j++) {
            $sub_array_1[] = array_shift($clustered_rows[$i][$j]);
        }
        array_push($sub_array_3, $sub_array_1);
    }
    array_push($data_by_cluster, $sub_array_3);
}

$new_associative_cluster_centroid = array();

for ($i=0; $i < $clusters_amount; $i++) {
    $sub_array_1 = array();
    for ($j=0; $j < $data_attributes; $j++) {
        $sub_array_2 = array();
        $sub_array_1[] = round(array_sum($data_by_cluster[$i][$j]) / $cluster_elements[$i], 2);
    }
    array_push($new_associative_cluster_centroid, $sub_array_1);
}

$data = array(
    'clusters_amount' => $clusters_amount,
    'cluster_centroid' => $cluster_centroid,
    'min_val' => $min_val,
    'data_index' => $data_index,
    'cluster_elements' => $cluster_elements,
    'clustered_rows' => $clustered_rows,
    'new_associative_cluster_centroid' => $new_associative_cluster_centroid
);

return $data;
}
?>

```

Gambar 3.4 Tampilan Kode Program

3.d Hasil Pembahasan

Hasil pembahasan meliputi contoh hasil eksekusi program dan pembahasannya. Pertama mengisi input data dalam bentuk excell pada tombol

browse, lalu menginput jumlah klaster, setelah itu mengklik tombol *process*, maka akan tampak seperti berikut :

No.	Diameter (MM)	Panjang Baut (MM)	Pitch (MM)	Cluster_0	Cluster_1
1	3	10	1	8,71	29,85
2	3	20	1	8,71	20,19
3	3	30	1	16,61	11,14
4	4	40	1	25,74	5,4
5	4	50	1	35,53	12,07
6	12	40	1,25	25,07	2,8
7	12	45	1,25	30,06	6,41
9	14	30	1,25	15,49	10,29
10	14	50	1,25	35,21	11,79
11	16	15	1,25	5,87	25,07
12	16	20	1,25	7,71	20,3
13	13	5	1,25	10,4	14,37
14	13	25	1,25	10,4	14,64
15	8	40	1	29,09	1,37
16	8	35	1	20,11	4,38
17	8	35	1	20,11	4,38
18	10	20	1	5	19,38
19	10	15	1	0,36	24,18
20	10	15	1	0,36	24,18
CLUSTERS ELEMENTS				9	11

Gambar 3.5 Tampilan Hasil Perhitungan Program

4. Penutup

4.a Kesimpulan

Telah dibuat program untuk mengelompokan baut berdasarkan diameter, panjang, dan pitch menggunakan metode *K-Means*. Program dapat dikembangkan dengan menambah fitur visualisasi seperti tampilan *Plot View*.

5. Pustaka

- [1] **Agusta, Yudi. Pebruari 2007.** "K-Means Penerapan, Permasalahan dan Metode Terkait". Jurnal Sistem dan Informatika Vol.3 : 47-60.
- [2] **Prasetyo, E. 2014.** Data Mining Mengolah Data Menjadi Informasi Menggunakan Matlab. Andi. Yogyakarta.
- [3] **Santoso, B. (2007), Data Mining: Teknik Pemanfaatan Data untuk Keperluan Bisnis,** Yogyakarta: Graha Ilmu.
- [4] **Tim Penyusun Kamus Pusat Pembinaan dan Pengembangan Bahasa. 1990,** Kamus Besar Bahasa Indonesia, Jakarta: Balai Pustaka.
- [5] **Wicaksana, I. W. S. (2013).** *Belajar Data Mining Dengan Rapid Miner.*
- [6] **Witten, Ian H. dan Frank, Eibe. 2005.** *Data Mining Practical Machine Learning Tools and Techniques, Second Edition.* Morgan