

BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1 Implementasi Sistem

Pada bab ini, dijelaskan secara rinci tentang mekanisme kerja sistem pendukung keputusan berbasis *web* yang dirancang, mencakup tahapan *input* data, proses pengolahan data menggunakan metode *Weighted Product* (WP), hingga menghasilkan *output* berupa rekomendasi mobil bekas terbaik.

4.1.1 *Input*

Input pada sistem ini adalah data yang dimasukkan oleh pengguna melalui antarmuka *web*. Data ini terdiri dari:

1. Informasi *Login*

Pengguna memasukkan *username* dan *password* untuk mengakses sistem. Validasi *input* dilakukan untuk memastikan pengguna yang masuk adalah yang terdaftar.

```
// Fungsi Validasi Login
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    $query = "SELECT * FROM users WHERE username
= ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $result = $stmt->get_result();
    $user = $result->fetch_assoc();
```

```

if ($user && password_verify($password,
$user['password'])) {
    $_SESSION['user_id'] = $user['id'];
    $_SESSION['username'] =
$user['username'];
    $_SESSION['login'] = true;
    header('Location:
add_default_criteria.php');
} else {
    echo "Login gagal.";
}
}

```

Gambar 4. 1 Kode Validasi Login

2. Bobot Kriteria

Pengguna dapat mengatur bobot untuk setiap kriteria, seperti Harga, Tahun, Jarak Tempuh, dan Kondisi Mobil. Bobot ini merepresentasikan tingkat kepentingan masing-masing kriteria dalam proses pengambilan keputusan.

```

<div class="card">
    <div class="card-header">Edit Bobot
Kriteria</div>
    <form action="edit_kriteria.php"
method="POST" enctype="multipart/form-data">
        <div class="form-group">
            <input type="hidden" name="id"
value="<?php echo $kriteria['id']; ?>">
        </div>
        <div class="form-group">
            <label for="nama">Nama
Kriteria:</label>
            <input type="text" name="nama"
id="nama" value="<?php echo
$kriteria['nama']; ?>" required readonly>
        </div>
        <div class="form-group">
            <label for="bobot">Bobot:</label>
            <select name="bobot" id="bobot"
required>

```

```

        <option value="" disabled
selected>Pilih Bobot</option>
        <?php for ($i = 1; $i <= 5;
$i++): ?>
            <option value="<?php echo
$i; ?>" <?php echo ($kriteria['bobot'] == $i)
? 'selected' : ''; ?>>
                <?php echo $i; ?>
            </option>
        <?php endfor; ?>
    </select>
</div>
    <button type="submit" name="update"
class="submit-button">Update</button>
    <a href="kriteria.php" class="delete-
button">Batal</a>
</form>
</div>
<?php // Fungsi Untuk Mengambil Data dari ID
function fetchCriteriaById($conn, $id)
{
    $query = "SELECT * FROM criteria WHERE
id='$id' AND user_id="
$_SESSION['user_id'];
    $result = mysqli_query($conn, $query);
    return mysqli_fetch_assoc($result);
}

// Fungsi Ubah Data Kriteria
function updateCriteria($conn, $id, $bobot)
{
    $query = "UPDATE criteria SET bobot =
'$bobot' WHERE id = $id";
    return mysqli_query($conn, $query);
}
?>

```

Gambar 4. 2 Kode Ubah Data Kriteria

3. Data Alternatif

Data alternatif mobil bekas dimasukkan melalui *form input* yang mencakup informasi seperti Nama Mobil, Harga, Tahun, Jarak Tempuh, dan Kondisi. Data ini akan menjadi dasar untuk perhitungan dan rekomendasi.

```

<form action="tambah.php" method="POST"
enctype="multipart/form-data">
  <div class="form-group">
    <label for="name">Nama</label>
    <input type="text" id="name"
name="name" required />
  </div>
  <div class="form-group">
    <label for="price">Harga</label>
    <input type="number" id="price"
name="price" required />
  </div>
  <div class="form-group">
    <label for="year">Tahun</label>
    <input type="number" id="year"
name="year" required />
  </div>
  <div class="form-group">
    <label for="mileage">Jarak Tempuh
(km)</label>
    <input type="number" id="mileage"
name="mileage" required />
  </div>
  <div class="form-group">
    <label
for="conditions">Kondisi</label>
    <select id="conditions"
name="conditions" required>
      <option value="10">Seperti
Baru</option>
      <option value="9">Sangat
Baik</option>
      <option value="8">Baik</option>
      <option value="7">Cukup
Baik</option>
      <option value="6">Cukup</option>
    </select>
  </div>
  <button type="submit" class="submit-
button">Tambah Data</button>
  <a href="alternatif.php" class="delete-
button">Batal</a>
</form>
<?php
//function tambah
function tambah($data)
{

```

```

global $conn;
$name = htmlspecialchars($data["name"]);
        $price           =
htmlspecialchars($data["price"]);
        $year = htmlspecialchars($data["year"]);
        $mileage       =
htmlspecialchars($data["mileage"]);
        $conditions = ($data["conditions"]);

    // masukan data
    $query = "INSERT INTO alternatives (name,
price, year, mileage, conditions, user_id)
VALUES      ('$name',      '$price',      '$year',
'$mileage',      '$conditions',      "
.$_SESSION["user_id"] . ")";
    mysqli_query($conn, $query);
    return mysqli_affected_rows($conn);
}
?>

```

Gambar 4. 3 Kode *Input* Data Alternatif

4.1.2 Proses

Proses pada sistem melibatkan pengolahan data *input* menggunakan metode *Weighted Product* (WP). Tahapan proses yang dilakukan adalah sebagai berikut:

1. Normalisasi Nilai bobot

Normalisasi bobot dihitung dengan membagi bobot kriteria dengan total bobot. Nama tampilan diambil dari *array \$criteriaNames*, dan jika tidak ada, nama kriteria itu sendiri digunakan. Pangkat dihitung berdasarkan tipe kriteria: jika tipe adalah '*cost*', pangkatnya adalah negatif dari bobot yang dinormalisasi; jika '*benefit*', pangkatnya adalah bobot yang dinormalisasi itu sendiri. Setiap kriteria kemudian ditampilkan dalam baris tabel dengan kolom yang sesuai.

```

<h2>Normalisasi Bobot (W)</h2>
<?php
    $criteria = $conn->query("SELECT * FROM
criteria      WHERE      user_id      =      "
$_SESSION['user_id']);
    $weights = [];
    $criteriaNames = [
        'price' => 'C1 Harga',
        'year' => 'C2 Tahun',
        'mileage' => 'C3 Jarak Tempuh',
        'conditions' => 'C4 Kondisi Mesin'
    ];
    $totalWeight = 0;
    while ($row = $criteria->fetch_assoc()) {
        $weights[$row['nama']] = [
            'weight' => $row['bobot'],
            'type' => $row['tipe']
        ];
        $totalWeight += $row['bobot'];
    }
    echo "<table border='1'>
    <thead>
        <tr>
            <th>Kriteria</th>
            <th>Bobot</th>
            <th>Normalisasi Bobot</th>
            <th>Pangkat</th>
        </tr>
    </thead>
    <tbody>";
    foreach ($weights as $name => $data) {
        $weights[$name]['normalized'] =
$data['weight'] / $totalWeight;
        $displayName = $criteriaNames[$name]
?? $name;
        $power = ($data['type'] === 'cost') ?
-$weights[$name]['normalized'] :
$weights[$name]['normalized'];
        $weights[$name]['power'] = $power;
        echo "<tr>
            <td>$displayName</td>

```

```

        <td>{$data ['weight' ]}</td>
        <td>{$weights[$name] ['normalized
' ]}</td>
        <td>$power</td>
    </tr>";
}

```

Gambar 4. 4 Kode Normalisasi Nilai Bobot

2. Perhitungan Nilai S

Dalam bagian ini, kode menggunakan *loop while* untuk iterasi melalui setiap alternatif yang diambil dari *database*. Untuk setiap alternatif, nilai (S) diinisialisasi dengan 1. Kemudian, kode menggunakan *loop foreach* untuk iterasi melalui setiap kriteria dalam *array* \$weights. Di dalam *loop* ini, nilai dari alternatif untuk kriteria tertentu diambil dan disimpan dalam variabel \$value. Pangkat dihitung berdasarkan tipe kriteria: jika tipe adalah 'cost', pangkatnya adalah negatif dari bobot yang dinormalisasi; jika 'benefit', pangkatnya adalah bobot yang dinormalisasi itu sendiri. Nilai (S) dihitung dengan mengalikan (S) dengan nilai kriteria yang dipangkatkan. Lalu menampilkan nama alternatif dan nilai (S) dalam baris tabel.

```

<h2>Menghitung Nilai Vektor (S)</h2>
<?php
    $stmt = $conn->prepare("SELECT * FROM
alternatives WHERE user_id = ?");
    $stmt->bind_param("i", $user_id);
    $stmt->execute();
    $alternatives = $stmt->get_result();

    $vectorS = [];
    echo "<table border='1'>
        <thead>
            <tr>
                <th>Alternatif</th>
                <th>Vector S</th>
            </tr>

```

```

        </thead>
        <tbody>";
        while ($alt = $alternatives-
>fetch_assoc()) {
            $S = 1;
            foreach ($weights as $name => $data)
            {
                $value = $alt[$name];
                $power = ($data['type'] ===
'cost') ? -$data['normalized'] :
$data['normalized'];
                $S *= pow($value, $power);
            }
            $vectorS[$alt['id']] = $S;
            echo
"<tr><td>{$alt['name']}</td><td>$S</td></tr>
";
            $_SESSION['vectorS'] = $vectorS; //
Simpan Vector S ke session
        }
        echo "</tbody></table>";
    ?>

```

Gambar 4. 5 Kode Perhitungan Nilai S

3. Perhitungan Nilai V

Dalam bagian ini, kode menggunakan *loop foreach* untuk iterasi melalui setiap nilai vektor (S) dalam *array \$vectorS*. Untuk setiap alternatif, nilai (V) dihitung dengan membagi nilai (S) dengan total (S) yang telah dihitung sebelumnya. Nilai (V) kemudian disimpan dalam *array \$vectorV* dengan ID alternatif sebagai kunci.

```

<h2>Menghitung Nilai Vektor (V)</h2>
<?php
$totalS = array_sum($vectorS);
$vectorV = [];
echo "<table border='1'>
    <thead>
        <tr>
            <th>Alternatif</th>
            <th>Vector V</th>

```



```

        </tr>
    </thead>
    <tbody>";
    foreach ($vectorS as $id => $S) {
        $V = $S / $totals;
        $vectorV[$id] = $V;

        $stmt = $conn->prepare("SELECT * FROM
alternatives WHERE id = ? AND user_id = ?");
        $stmt->bind_param("ii", $id, $user_id);
        $stmt->execute();
        $alt = $stmt->get_result()->fetch_assoc();

        echo
        "<tr><td>{$alt['name']}</td><td>$V</td></tr>";
    }
    echo "</tbody></table>";
?>

```

Gambar 4. 6 Kode Perhitungan Nilai V

4.1.3 Output

Output dari sistem adalah hasil akhir berupa rekomendasi alternatif mobil bekas terbaik yang ditampilkan kepada pengguna.

```

<h2> Perankingan Nilai Vektor</h2>
<?php
arsort($vectorV);
$rank = 1;
echo "<table border='1'>
    <thead>
        <tr>
            <th>Rank</th>
            <th>Alternative</th>
            <th>Vector V</th>
        </tr>
    </thead>
    <tbody>";
$rankedAlternatives = [];
foreach ($vectorV as $id => $V) {
    $alt = $conn->query("SELECT * FROM alternatives
WHERE id=$id")->fetch_assoc();
    $rankedAlternatives[] = [

```

```

        'name' => $alt['nama'],
        'value' => $V
    ];

    echo
    "<tr><td>$rank</td><td>{$alt['nama']}</td><td>$V</td>
    </tr>";
    $rank++;
}
echo "</tbody></table>";

echo "<h2>Conclusion</h2>";
echo "<p>Dari tabel tersebut dapat disimpulkan bahwa
{$rankedAlternatives[0]['name']} mempunyai hasil
paling tinggi, yaitu
{$rankedAlternatives[0]['value']}.<br>";
for ($i = 1; $i < count($rankedAlternatives) - 1; $i++)
{
    echo "Lalu diikuti dengan
    {$rankedAlternatives[$i]['name']} dengan nilai
    {$rankedAlternatives[$i]['value']}.<br>";
}
$last = count($rankedAlternatives) - 1;
echo "Dan terakhir
{$rankedAlternatives[$last]['name']} dengan nilai
{$rankedAlternatives[$last]['value']}.</p>";
?>

```

Dalam bagian ini, kode menggunakan *loop foreach* untuk iterasi melalui setiap nilai vektor (*V*) yang telah diurutkan. Untuk setiap alternatif, kode melakukan *query* ke *database* untuk mengambil data alternatif berdasarkan ID alternatif. Hasil *query* disimpan dalam variabel *\$alt*, yang berisi informasi tentang alternatif tersebut. Alternatif yang diranking kemudian disimpan dalam *array* *\$rankedAlternatives*, yang berisi nama alternatif dan nilai vektor (*V*). Kode juga menampilkan peringkat, nama alternatif, dan nilai (*V*) dalam baris tabel. Setelah menampilkan setiap alternatif, variabel *\$rank* diinkrementasi untuk memberikan peringkat berikutnya.

4.2 Pengujian sistem

Pembahasan berisi hasil pengujian yang dilakukan dengan membandingkan perhitungan manual dan sistem untuk membuktikan bahwa hasil perhitungan sistem valid dengan menggunakan data alternatif. Pengujian ini terdiri dari dua proses yaitu proses perhitungan produk tertimbang dan proses pemeringkatan data mobil bekas.

4.2.1 Data Mobil Bekas

Pada penelitian ini, data mobil bekas menjadi salah satu elemen penting dalam sistem pendukung keputusan (SPK) yang dirancang. Data ini digunakan sebagai alternatif dalam proses perhitungan metode *Weighted Product* untuk memberikan rekomendasi kepada pengguna. Berikut 50 data mobil bekas

Tabel 4. 1 Data Mobil Bekas

No	Nama	Harga(Rp)	Tahun	Jarak(Km)	Kondisi
1	Mazda CX-5	295000000	2016	115872	7
2	BMW 320i SPORT	815000000	2020	19312	10
3	Chevrolet TRAX	205000000	2017	67592	8
4	Toyota Calya G 1.2	117000000	2017	231744	6
5	Toyota Grand Avanza G 1.3	170000000	2019	164152	6
6	Daihatsu Gran Max D 1.3	107500000	2017	286462	6
7	Toyota Grand New Avanza Veloz 1.5	165000000	2017	136793	7
8	Daihatsu Sigr X 1.2	115000000	2018	98169	8
9	Toyota Calya G 1.2	122000000	2008	165762	6
10	Toyota Grand New Avanza G 1.3	157500000	2018	196339	6
11	Daihatsu Gran Max D 1.5	132500000	2019	128747	7
12	Mitsubishi Outlander Sport PX 2.0	235000000	2017	91732	8
13	Honda Jazz S CVT 1.5	185500000	2018	94951	8

Tabel 4. 15 (Lanjutan)

14	Daihatsu Sigras R Deluxe 1.2	109500000	2017	151277	6
15	Daihatsu Gran Max D Front Facing 1.3	115000000	2018	222088	6
16	Toyota Grand New Avanza G 1.3	150000000	2017	148059	7
17	BMW X7	2275000000	2020	6437	10
18	Mercedes Benz E 300 AT (CKD)	375000000	2012	78857	8
19	Jeep GRAND CHEROKEE 5.7AT	425000000	2011	72420	8
20	Mercedes Benz R350 L AT	285000000	2007	144840	7
21	Toyota LC 100 4.7 AT	475000000	2004	233354	6
22	Daihatsu AYL1.0XAT(B100GQQFJ	103000000	2016	96	10
23	Toyota HARRIER 2.4L 2WD AT	159000000	2006	241401	6
24	BMW 528 I AT	650000000	1997	257494	6
25	Range Rover RANGE ROVER 4.2L V8 AT	535000000	2007	152	10
26	Mercedes Benz 300 GE	875000000	1993	209214	6
27	VW TOUAREG 2.5L AT	215000000	2006	88513	8
28	Jaguar XF 3.0 V6 AT	289000000	2011	120700	7
29	Audi Q7 3.0 TFSI AT	323000000	2011	193	10
30	Toyota ESTIMA 2.4 2WD AT	145000000	2007	185074	6
31	Toyota Kijang Innova Reborn G 2.0	260000000	2018	157715	6
32	Daihatsu Gran Max D 1.5	107500000	2016	165762	6
33	Honda Mobilio E CVT 1.5	142000000	2016	186683	6
34	Toyota All New Calya G 1.2	147000000	2021	90123	8
35	Daihatsu Sigras X 1.2	105000000	2016	217260	6
36	Toyota Rush S TRD Sportivo 1.5	177500000	2017	69201	8
37	Toyota Grand Avanza G 1.3	172500000	2019	148059	7
38	Toyota Grand New Avanza E 1.3	142500000	2018	215651	6
39	Toyota Kijang Innova Reborn V 2.0	300000000	2019	133575	7
40	Toyota Calya G 1.2	122500000	2018	82076	8
41	Honda JAZZ GK5	250000000	2020	24140	9
42	Toyota YARIS 1.5 S	175000000	2016	88	10
43	Daihatsu TER15RF	145000000	2016	125528	7
44	Mazda 3 5 HB 6E	340000000	2016	88513	8
45	Mitsubishi OUTL SP2.0L GLS4X2	140000000	2013	225307	6
46	Toyota FORTUNER2.7SRZ	385000000	2018	157715	6
47	Honda ACCORD CP2 2.4	205000000	2013	131965	7
48	Nissan NEW TERRAE2.5	340000000	2019	70810	8

Tabel 4. 15 (Lanjutan)

49	Daihatsu TERIOS F700RG	123000000	2012	172199	6
50	Toyota RUSH 1.5 G	205000000	2019	117481	7

Kemudian data pada tabel diatas dimasukan ke dalam sistem oleh *user* maka akan tampil seperti gambar dibawah ini, *user* dapat menambah, mengubah, dan menghapus data mobil bekas tersebut.

SPK PEMILIHAN MOBIL BEKAS METODE WEIGHTED PRODUCT

Data Kriteria Data Alternatif Perhitungan Hasil Perhitungan Logout

Alternatif
has logged in as: ADMIN
USER_ID: 20

Tambah Data

Nama	Harga	Tahun	Jarak Tempuh	Kondisi	Aksi
Mazda CX-5	Rp 295.000.000	2016	115.872 KM	Cukup Baik	Edit Hapus
BMW 320i SPORT	Rp 815.000.000	2020	19.312 KM	Seperti Baru	Edit Hapus
Chevrolet TRAX	Rp 205.000.000	2017	67.592 KM	Baik	Edit Hapus
Toyota Calya G 1.2	Rp 117.000.000	2017	231.744 KM	Cukup	Edit Hapus
Toyota Grand Avanza G 1.3	Rp 170.000.000	2019	164.152 KM	Cukup	Edit Hapus
Daihatsu Gran Max D 1.3	Rp 107.500.000	2017	266.462 KM	Cukup	Edit Hapus
Toyota Grand New Avanza Veloz 1.5	Rp 165.000.000	2017	136.793 KM	Cukup Baik	Edit Hapus
Daihatsu Sigra X 1.2	Rp 115.000.000	2018	98.169 KM	Baik	Edit Hapus
Toyota Calya G 1.2	Rp 122.000.000	2008	165.762 KM	Cukup	Edit Hapus
Toyota Grand New Avanza G 1.3	Rp 157.500.000	2018	196.339 KM	Cukup	Edit Hapus
Daihatsu Gran Max D 1.5	Rp 132.500.000	2019	128.747 KM	Cukup Baik	Edit Hapus
Mitsubishi Outlander Sport PX 2.0	Rp 235.000.000	2017	91.732 KM	Baik	Edit Hapus
Honda Jazz S CVT 1.5	Rp 185.500.000	2018	94.951 KM	Baik	Edit Hapus
Daihatsu Sigra R Deluxe 1.2	Rp 109.500.000	2017	151.277 KM	Cukup	Edit Hapus
Daihatsu Gran Max D Front Facing 1.3	Rp 115.000.000	2018	222.088 KM	Cukup	Edit Hapus
Toyota Grand New Avanza G 1.3	Rp 150.000.000	2017	148.059 KM	Cukup Baik	Edit Hapus
BMW X7	Rp 2.275.000.000	2020	6.437 KM	Seperti Baru	Edit Hapus
Mercedes Benz E 300 AT (CKD)	Rp 375.000.000	2012	78.857 KM	Baik	Edit Hapus
Jeep GRAND CHEROKEE 5.7AT	Rp 425.000.000	2011	72.420 KM	Baik	Edit Hapus
Mercedes Benz R350 LAT	Rp 285.000.000	2007	144.840 KM	Cukup Baik	Edit Hapus
Toyota LC 100 4.7 AT	Rp 475.000.000	2004	233.354 KM	Cukup	Edit Hapus
Daihatsu AYLAI.0XAT(B100GGQFJ	Rp 103.000.000	2016	96 KM	Seperti Baru	Edit Hapus
Toyota HARRIER 2.4L 2WD AT	Rp 159.000.000	2006	241.401 KM	Cukup	Edit Hapus
BMW 528 IAT	Rp 650.000.000	1997	257.494 KM	Cukup	Edit Hapus
Range Rover RANGE ROVER 4.2L V8 AT	Rp 535.000.000	2007	152 KM	Seperti Baru	Edit Hapus
Mercedes Benz 300 GE	Rp 875.000.000	1993	209.214 KM	Cukup	Edit Hapus
VW TQUAREG 2.5L AT	Rp 215.000.000	2006	86.513 KM	Baik	Edit Hapus
Jaguar XF 3.0 V8 AT	Rp 289.000.000	2011	120.700 KM	Cukup Baik	Edit Hapus
Audi Q7 3.0 TFSI AT	Rp 323.000.000	2011	193 KM	Seperti Baru	Edit Hapus
Toyota ESTIMA 2.4 2WD AT	Rp 145.000.000	2007	185.074 KM	Cukup	Edit Hapus
Toyota Kijang Innova Reborn G 2.0	Rp 260.000.000	2018	157.715 KM	Cukup	Edit Hapus
Daihatsu Gran Max D 1.5	Rp 107.500.000	2016	165.762 KM	Cukup	Edit Hapus
Honda Mobilio E CVT 1.5	Rp 142.000.000	2016	186.683 KM	Cukup	Edit Hapus
Toyota All New Calya G 1.2	Rp 147.000.000	2021	90.123 KM	Baik	Edit Hapus
Daihatsu Sigra X 1.2	Rp 105.000.000	2016	217.260 KM	Cukup	Edit Hapus
Toyota Rush S TRD Sportivo 1.5	Rp 177.500.000	2017	69.201 KM	Baik	Edit Hapus
Toyota Grand Avanza G 1.3	Rp 172.500.000	2019	148.059 KM	Cukup Baik	Edit Hapus
Toyota Grand New Avanza E 1.3	Rp 142.500.000	2018	215.651 KM	Cukup	Edit Hapus
Toyota Kijang Innova Reborn V 2.0	Rp 300.000.000	2019	133.575 KM	Cukup Baik	Edit Hapus
Toyota Calya G 1.2	Rp 122.500.000	2018	82.076 KM	Baik	Edit Hapus
Honda JAZZ GK5	Rp 250.000.000	2020	24.140 KM	Sangat Baik	Edit Hapus
Toyota YARIS 1.5 S	Rp 175.000.000	2016	88 KM	Seperti Baru	Edit Hapus
Daihatsu TER15RF	Rp 145.000.000	2016	125.528 KM	Cukup Baik	Edit Hapus
Mazda 3 5 HB 6E	Rp 340.000.000	2016	88.513 KM	Baik	Edit Hapus
Mitsubishi OUTL SP2.0L GLS4X2	Rp 140.000.000	2013	225.307 KM	Cukup	Edit Hapus
Toyota FORTUNER2.7SRZ	Rp 385.000.000	2018	157.715 KM	Cukup	Edit Hapus
Honda ACCORD CP2 2.4	Rp 205.000.000	2013	131.965 KM	Cukup Baik	Edit Hapus
Nissan NEW TERRAZ2.5	Rp 340.000.000	2019	70.810 KM	Baik	Edit Hapus
Daihatsu TERIOS F700RG	Rp 123.000.000	2012	172.199 KM	Cukup	Edit Hapus
Toyota RUSH 1.5 G	Rp 205.000.000	2019	117.481 KM	Cukup Baik	Edit Hapus

© Surya Jaya

Gambar 4. 7 Tampilan Data Alternatif Mobil Bekas

4.2.2 Proses Perhitungan Nilai W

Berikut adalah proses penghitungan nilai W dari metode *Weighted Product* dengan membandingkan hasil perhitungan manual menggunakan aplikasi *Microsoft*

excel dan perhitungan sistem disajikan pada tabel berikut ini, dan dari perbandingan ini didapat hasil yang sama.

Tabel 4. 2 Data Perbandingan Perhitungan Manual dan Sistem

Kriteria	Bobot	Nilai Normalisasi Bobot	
		Perhitungan Manual	Perhitungan Sistem
harga	4	0,2666666667	0.266666666666667
tahun	3	0,2	0,2
jarak	3	0,2	0,2
kondisi	5	0,3333333333	0.333333333333333
total	15		

4.2.3 Proses Perhitungan Vektor S

Berikut adalah proses penghitungan nilai S dari metode *Weighted Product* dengan membandingkan hasil perhitungan manual menggunakan *Microsoft excel* dan perhitungan sistem disajikan pada tabel berikut ini, dan dari perbandingan ini didapat hasil yang sama.

Tabel 4. 3 Data Perbandingan Perhitungan Nilai S Manual dan Sistem

Data Alternatif	Nilai S	
	Perhitungan Manual	Perhitungan Sistem
A1	0,004690105	0.00469010480259683
A2	0,005766728	0.005766727914190068
A3	0,006019001	0.0060190010571774906
A4	0,004963717	0.0049637168562786296
A5	0,004814768	0.004814768374058069
A6	0,004866342	0.004866341716767599
A7	0,005297838	0.005297838312399626
A8	0,006517784	0.006517783500691362
A9	0,005244171	0.005244171363834538
A10	0,004740511	0.004740511291875397
A11	0,00568663	0.005686629748416762
A12	0,005459869	0.005459869212018973
A13	0,005775939	0.005775939147401719
A14	0,005502076	0.005502075987267876

Tabel 4. 3 (Lanjutan)

Data Alternatif	Nilai S	
	Perhitungan Manual	Perhitungan Sistem
A15	0,005029717	0.0050297168828519175
A16	0,005348876	0.005348876388365311
A17	0,005463512	0.005463511545080814
A18	0,004965685	0.004965685414174989
A19	0,00488469	0.004884689819680267
A20	0,004522784	0.0045227837377418865
A21	0,003407025	0.003407024691917028
A22	0,02890817	0.02890816993480125
A23	0,004531718	0.004531718135827734
A24	0,003070405	0.0030704053343341324
A25	0,016979038	0.016979038054626226
A26	0,002955504	0.0029555041974768325
A27	0,005624853	0.005624853037312628
A28	0,004675207	0.004675207404010776
A29	0,018526083	0.01852608332405926
A30	0,004898457	0.004898457470244458
A31	0,004333118	0.0043331177376184395
A32	0,005428451	0.00542845117814311
A33	0,004921725	0.004921724987981759
A34	0,006211906	0.006211905511333756
A35	0,005174901	0.005174901497477324
A36	0,006225331	0.006225331356068581
A37	0,005154215	0.005154215289785591
A38	0,00477823	0.004778230066793246
A39	0,004539582	0.004539582424916517
A40	0,006642548	0.006642547996875599
A41	0,007297145	0.007297145203759833
A42	0,025538267	0.025538267096959274
A43	0,005578077	0.00557807731802938
A44	0,00498271	0.004982710379955523
A45	0,004756601	0.004756601391796538
A46	0,003902449	0.003902448567021652
A47	0,005033947	0.005033947164466044
A48	0,005211675	0.005211674648934043
A49	0,005195115	0.005195115299678391
A50	0,005155436	0.005155435

Berikut ini adalah tampilan dari sistem hasil perhitungan nilai S yang ditampilkan menggunakan *var_dump* pada PHP untuk membuktikan bahwa *input* pengguna berjalan pada penghitungan sistem.

```
array(50) {
  [68] => float(0.004690104802596833)
  [69] => float(0.005766727914190068)
  [70] => float(0.0060190010571774906)
  [71] => float(0.0049637168562786296)
  [72] => float(0.004814768374058069)
  [73] => float(0.004866341716767599)
  [74] => float(0.005297838312399626)
  [75] => float(0.006517783500691362)
  [76] => float(0.005244171363834538)
  [77] => float(0.004740511291875397)
  [78] => float(0.005686629748416762)
  [79] => float(0.005459869212018973)
  [80] => float(0.005775939147401719)
  [81] => float(0.005502075987267876)
  [82] => float(0.0050297168828519175)
  [83] => float(0.005348876388365311)
  [84] => float(0.005463511545080814)
  [85] => float(0.004965685414174989)
  [86] => float(0.004884689819680267)
  [87] => float(0.0045227837377418865)
  [88] => float(0.003407024691917028)
  [89] => float(0.02890816993480125)
  [90] => float(0.004531718135827734)
  [91] => float(0.0030704053343341324)
  [92] => float(0.016979038054626226)
  [93] => float(0.0029555041974768325)
  [94] => float(0.005624853037312628)
  [95] => float(0.004675207404010776)
```

Gambar 4. 8 Hasil Perhitungan Nilai S Sistem

4.2.4 Proses Perhitungan Vektor V

Berikut adalah proses penghitungan nilai V dari metode *Weighted Product* dengan membandingkan hasil perhitungan manual menggunakan *Microsoft excel* dan perhitungan sistem, disajikan pada tabel berikut ini, dan dari perbandingan ini didapat hasil yang sama.

Tabel 4. 4 Data Perbandingan Hasil Perhitungan Nilai V Manual Dan Sistem

Nilai V	
Perhitungan Manual	Perhitungan Sistem
0,014422277	0.01442227699871922
0,01773294	0.017732940063225452

Tabel 4. 18 (Lanjutan)

Nilai V	
Perhitungan Manual	Perhitungan Sistem
0,018508691	0.01850869099004645
0,015263646	0.015263646007403717
0,014805623	0.014805622922730998
0,014964213	0.014964213202823436
0,016291084	0.01629108406992299
0,020042469	0.020042468776519815
0,016126056	0.016126055860435393
0,014577279	0.014577279153576378
0,017486635	0.017486634707056792
0,016789336	0.016789336159165856
0,017761265	0.01776126500743665
0,016919124	0.016919123835450904
0,015466599	0.01546659897012552
0,016448028	0.016448028381412835
0,016800536	0.016800536492324804
0,015269699	0.015269699408865788
0,015020634	0.015020634420204546
0,013907757	0.013907757420452528
0,01047675	0.010476749649836612
0,088893884	0.0888938844383409
0,013935231	0.013935231084567576
0,00944163	0.009441630431284523
0,052211283	0.05221128318071488
0,009088304	0.00908830441982606
0,017296669	0.017296668623757252
0,014376467	0.014376466847771316
0,056968515	0.056968514915273924
0,015062971	0.015062970546669866
0,013324526	0.013324526190842283
0,016692724	0.01669272433354909
0,015134519	0.015134519179377212
0,019101881	0.019101880607171415
0,015913048	0.015913047997644172
0,019143166	0.01914316566579105
0,015849437	0.015849437006005343
0,014693266	0.014693266032934886
0,013959414	0.013959414116804646

Tabel 4. 18 (Lanjutan)

Nilai V	
Perhitungan Manual	Perhitungan Sistem
0,020426125	0.020426125048460354
0,02243904	0.022439040033866466
0,078531286	0.07853128611021462
0,017152831	0.01715283124512473
0,015322052	0.01532205192180849
0,014626757	0.014626756913193024
0,0120002	0.012000199691844547
0,015479607	0.01547960727074766
0,016026127	0.0160261270435817
0,015975206	0.01597520632177793
0,01585319	0.015853190283317477

Berikut ini adalah tampilan dari sistem hasil perhitungan nilai V yang ditampilkan menggunakan *var_dump* pada PHP untuk membuktikan bahwa *input* pengguna berjalan pada penghitungan sistem.

```

float(0.01442227699871922)
float(0.017732940063225452)
float(0.01850869099004645)
float(0.015263646007403717)
float(0.014805622922730998)
float(0.014964213202823436)
float(0.01629108406992299)
float(0.020042468776519815)
float(0.016126055860435393)
float(0.014577279153576378)
float(0.017486634707056792)
float(0.016789336159165856)
float(0.01776126500743665)
float(0.016919123835450904)
float(0.01546659897012552)
float(0.016448028381412835)
float(0.016800536492324804)
float(0.015269699408865788)
float(0.015020634420204546)
float(0.013907757420452528)
float(0.010476749649836612)
float(0.0888938844383409)
float(0.013935231084567576)
float(0.009441630431284523)
float(0.05221128318071488)
float(0.00908830441982606)
float(0.017296668623757252)
float(0.014376466847771316)
float(0.056968514915273924)
float(0.015062970546669866)
float(0.013324526190842283)
float(0.01669272433354909)
float(0.015134519179377212)
float(0.019101880607171415)
float(0.015913047997644172)
float(0.01914316566579105)
float(0.015849437006005343)
float(0.014693266032934886)
float(0.013959414116804646)
float(0.020426125048460354)
float(0.022439040033866466)
float(0.07853128611021462)
float(0.01715283124512473)
float(0.01532205192180849)
float(0.014626756913193024)
float(0.012000199691844547)
float(0.01547960727074766)
float(0.0160261270435817)
float(0.01597520632177793)

```

Gambar 4. 9 Hasil Perhitungan Nilai V Sistem

4.2.5 Perangkingan

Berikut ini adalah proses membandingkan hasil perangkingan nilai V manual dan sistem. Dari perbandingan ini didapat hasil yang sama dan didapat bahwa nilai mobil bekas yang terbaik.

Tabel 4. 5 Hasil Perangkingan dari Nilai V

Perangkingan			
Data Alternatif	Nama Mobil Bekas	Perhitungan Manual	Perhitungan Sistem
A22	Daihatsu AYLA1.0XAT(B100GQQFJ)	0,088893884	0.0888938844383409
A42	Toyota YARIS 1.5 S	0,078531286	0.07853128611021462
A29	Audi Q7 3.0 TFSI AT	0,056968515	0.056968514915273924
A25	Range Rover RANGE ROVER 4.2L V8 AT	0,052211283	0.05221128318071488
A41	Honda JAZZ GK5	0,02243904	0.022439040033866466
A40	Toyota Calya G 1.2	0,020426125	0.020426125048460354
A8	Daihatsu Sigr X 1.2	0,020042469	0.020042468776519815
A36	Toyota Rush S TRD Sportivo 1.5	0,019143166	0.01914316566579105
A34	Toyota All New Calya G 1.2	0,019101881	0.019101880607171415
A3	Chevrolet TRAX	0,018508691	0.01850869099004645
A13	Honda Jazz S CVT 1.5	0,017761265	0.01776126500743665
A2	BMW 320i SPORT	0,01773294	0.017732940063225452
A11	Daihatsu Gran Max D 1.5	0,017486635	0.017486634707056792
A27	VW TOUAREG 2.5L AT	0,017296669	0.017296668623757252
A43	Daihatsu TER15RF	0,017152831	0.01715283124512473
A14	Daihatsu Sigr R Deluxe 1.2	0,016919124	0.016919123835450904
A17	BMW X7	0,016800536	0.016800536492324804
A12	Mitsubishi Outlander Sport PX 2.0	0,016789336	0.016789336159165856
A32	Daihatsu Gran Max D 1.5	0,016692724	0.01669272433354909
A16	Toyota Grand New Avanza G 1.3	0,016448028	0.016448028381412835
A7	Toyota Grand New Avanza Veloz 1.5	0,016291084	0.01629108406992299
A9	Toyota Calya G 1.2	0,016126056	0.016126055860435393
A48	Nissan NEW TERRAE2.5	0,016026127	0.0160261270435817
A49	Daihatsu TERIOS F700RG	0,015975206	0.01597520632177793
A35	Daihatsu Sigr X 1.2	0,015913048	0.015913047997644172
A50	Toyota RUSH 1.5 G	0,01585319	0.015853190283317477
A37	Toyota Grand Avanza G 1.3	0,015849437	0.015849437006005343
A47	Honda ACCORD CP2 2.4	0,015479607	0.01547960727074766
A15	Daihatsu Gran Max D Front Facing 1.3	0,015466599	0.01546659897012552
A44	Mazda 3 5 HB 6E	0,015322052	0.01532205192180849
A18	Mercedes Benz E 300 AT (CKD)	0,015269699	0.015269699408865788
A4	Toyota Calya G 1.2	0,015263646	0.015263646007403717
A33	Honda Mobilio E CVT 1.5	0,015134519	0.015134519179377212
A30	Toyota ESTIMA 2.4 2WD AT	0,015062971	0.015062970546669866
A19	Jeep GRAND CHEROKEE 5.7AT	0,015020634	0.015020634420204546
A6	Daihatsu Gran Max D 1.3	0,014964213	0.014964213202823436

Tabel 4. 19 (Lanjutan)

A5	Toyota Grand Avanza G 1.3	0,014805623	0.014805622922730998
A38	Toyota Grand New Avanza E 1.3	0,014693266	0.014693266032934886
A45	Mitsubishi OUTL SP2.0L GLS4X2	0,014626757	0.014626756913193024
A10	Toyota Grand New Avanza G 1.3	0,014577279	0.014577279153576378
A1	Mazda CX-5	0,014422277	0.01442227699871922
A28	Jaguar XF 3.0 V6 AT	0,014376467	0.014376466847771316
A39	Toyota Kijang Innova Reborn V 2.0	0,013959414	0.013959414116804646
A23	Toyota HARRIER 2.4L 2WD AT	0,013935231	0.013935231084567576
A20	Mercedes Benz R350 L AT	0,013907757	0.013907757420452528
A31	Toyota Kijang Innova Reborn G 2.0	0,013324526	0.013324526190842283
A46	Toyota FORTUNER2.7SRZ	0,0120002	0.012000199691844547
A21	Toyota LC 100 4.7 AT	0,01047675	0.010476749649836612
A24	BMW 528 I AT	0,00944163	0.009441630431284523
A26	Mercedes Benz 300 GE	0,009088304	0.00908830441982606

Berikut ini adalah tampilan dari sistem hasil perangkingan nilai V pada gambar yang ditampilkan menggunakan *var_dump* pada PHP untuk membuktikan bahwa *input* pengguna berjalan pada penghitungan sistem serta hasil tampilan *website* ke pengguna dengan bobot default kepentingan pada harga = 4, tahun = 3, jarak = 3, dan kondisi = 5, memberikan informasi mobil bekas dengan tiga nilai tertinggi yaitu nilai V tertinggi = 0.0888938844383409 untuk Daihatsu AYLA1.0XAT(B100GQQFJ), lalu diikuti oleh nilai V tertinggi kedua = 0.07853128611021462 untuk Toyota YARIS 1.5 S, dan yang terakhir yaitu nilai V ketiga = 0.056968514915273924 untuk Audi Q7 3.0 TFSI AT.

```

[0]=>
array(2) {
  ["name"]=>
  string(29) "Daihatsu AYLA1.0XAT(B100GQQFJ"
  ["value"]=>
  float(0.0888938844383409)
}
[1]=>
array(2) {
  ["name"]=>
  string(18) "Toyota YARIS 1.5 S"
  ["value"]=>
  float(0.07853128611021462)
}
[2]=>
array(2) {
  ["name"]=>
  string(19) "Audi Q7 3.0 TFSI AT"
  ["value"]=>
  float(0.056968514915273924)
}

```

Gambar 4. 10 Hasil Perangkingan Nilai V Sistem

SPK PEMILIHAN MOBIL BEKAS METODE WEIGHTED PRODUCT		
		Data Kriteria Data Alternatif Perhitungan Hasil Perhitungan Logout
Perankingan Nilai Vektor		
Ranking	Alternatif	Vector V
1	Daihatsu AYLA1.0XAT(B100GQQFJ	0.088893884438341
2	Toyota YARIS 1.5 S	0.078531286110215
3	Audi Q7 3.0 TFSI AT	0.056968514915274
Lihat Lebih Banyak		
<p><small>Berdasarkan perhitungan yang telah dilakukan, didapatkan hasil ranking sebagai berikut: Dari tabel tersebut dapat disimpulkan bahwa Daihatsu AYLA1.0XAT(B100GQQFJ mempunyai hasil paling tinggi, yaitu 0.088893884438341. Lalu diikuti dengan Toyota YARIS 1.5 S dengan nilai 0.078531286110215. Dan terakhir Audi Q7 3.0 TFSI AT dengan nilai 0.056968514915274.</small></p>		

Gambar 4. 11 Tampilan Hasil Perangkingan Pada Sistem