

BAB IV

IMPLEMENTASI

4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan dijelaskan dalam sub bab ini adalah lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan perangkat lunak ini adalah sebagai berikut :

1. Intel® Core™ i3 CPU 540 @ 3,07GHz
2. Memori 2 GB
3. Harddisk 500 GB
4. Monitor 21"
5. Keyboard
6. Mouse

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan perangkat lunak ini adalah sebagai berikut :

1. Sistem Operasi Windows 8 Professional 32 bit
2. Notepad++ v6.3
3. XAMPP 1.8.1
4. Apache 2.4.3
5. MySQL 5.5.27
6. PHP 5.4.7
7. PhpMyAdmin 3.5.2.2

4.2 Implementasi Program

Berdasarkan analisa dan perancangan proses yang terdapat pada bab 3, maka pada subbab ini akan dijelaskan implementasi proses-proses tersebut.

4.2.1 Implementasi *Clustering*

Proses *clustering* pada aplikasi yang dibangun menggunakan algoritma *K-Means clustering*. Salah satu proses yang ada pada *K-Means clustering* adalah perhitungan jarak data dengan pusat *cluster*. Proses perhitungan tersebut menggunakan metode *Euclidian Distance*, seperti ditunjukkan pada Gambar 4.1.

```
function EuclidianDistance($data1, $data2)
{
    $jumlah = 0;
    $ind = 0;
    foreach ($data1 as $key => $value)
    {
        $jumlah += pow($value - $data2[$ind], 2);
        $ind++;
    }
    return sqrt($jumlah);
}
```

Gambar 4.1 Source Code Fungsi Hitung *Euclidian Distance*

Di dalam proses *clustering* dilakukan perhitungan nilai *ratio* antar *cluster* ($\frac{BCV}{WCV}$), perhitungan tersebut dilakukan pada setiap iterasi. Nilai *BCV* dihitung terlebih dahulu menggunakan fungsi `HitungBCV`, setelah itu nilai *WCV* dihitung menggunakan fungsi `HitungWCV`. Setelah didapatkan nilai keduanya, dilakukan perhitungan $\frac{BCV}{WCV}$ menggunakan fungsi `HitungLimit`. Implementasi langkah ini seperti ditunjukkan pada Gambar 4.2.

```
function HitungBCV($PusatCluster)
{
    $jumlahData = count($PusatCluster[0]);

    $sumAll = 0;
    for($i=0;$i<$jumlahData;$i++)
    {
        $sum = 0;
        for($j=(K_CLUSTER-1);$j>=0;$j--)
        {
            if($j == (K_CLUSTER-1))
            {
                $sum = $PusatCluster[$j][$i];
            }
            else
            {
                $sum = $PusatCluster[$j][$i];
            }
        }
        $sumAll += pow($sum, 2);
    }
    $Result = sqrt($sumAll);
    return $Result;
}

function HitungWCV($DataJarak)
{
    $sumAll = 0;
    foreach ($DataJarak as $key => $itemjarak)
    {
        $indCluster = $this->Minimum($itemjarak);

        $sumAll += pow($itemjarak[$indCluster], 2);
    }
    return $sumAll;
}

function HitungLimit($PusatCluster, $DataJarak)
{
    return $this->HitungBCV($PusatCluster)/$this->HitungWCV($DataJarak);
}
```

Gambar 4.2 Source Code Fungsi Hitung BCV/WCV

Setelah didapatkan sejumlah *cluster* dari proses *clustering* diatas, kemudian dilakukan perhitungan peluang tiap *item* terhadap *cluster* (*Pro*). Pada proses perhitungan *Pro* diperlukan nilai maksimal pada tiap *cluster*

(\$dataCluster[\$indMaks]). Perhitungan *Pro* menggunakan fungsi *Pro*. Implementasi langkah ini ditunjukkan pada Gambar 4.3

```
function Pro($dataCluster)
{
    $NewDataCluster = array();

    $indMaks = $this->Maksumum($dataCluster);
    foreach ($dataCluster as $key => $value)
    {
        $pro_ = 1 - ($value/$dataCluster[$indMaks]);
        array_push($NewDataCluster, $pro_);
    }
    return $NewDataCluster;
}
```

Gambar 4.3 Source Code Fungsi Hitung *Pro*

Dalam aplikasi yang dikembangkan, hasil dari perhitungan *clustering* (setelah dilakukan perhitungan *Pro*) disimpan dalam *database*, yaitu pada tabel *clustering*. *Query* untuk membuat tabel *clustering* ditunjukkan pada Gambar 4.4. *Field* *item_id* digunakan untuk menyimpan *id item*, sedangkan *field* *c1* sampai dengan *c10* digunakan untuk menyimpan nilai hasil perhitungan *Pro*.

```
CREATE TABLE IF NOT EXISTS `clustering`
(
    `item_id` int(11) NOT NULL,
    `c1` double NOT NULL,
    `c2` double NOT NULL,
    `c3` double NOT NULL,
    `c4` double NOT NULL,
    `c5` double NOT NULL,
    `c6` double NOT NULL,
    `c7` double NOT NULL,
    `c8` double NOT NULL,
    `c9` double NOT NULL,
    `c10` double NOT NULL,
    PRIMARY KEY (`item_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Gambar 4.4 Source Code Create Tabel *Clustering*

Untuk menyimpan hasil perhitungan *clustering* menggunakan fungsi *SavePro*. Implementasi pada langkah ini ditunjukkan pada Gambar 4.5

```
function SavePro($data)
{
    $DB      = new Database();
    $con= $DB->OpenConnection();
    foreach ($data as $key => $value)
    {
        $query = "INSERT INTO clustering values(".$key;
        foreach ($value as $key2 => $value2)
        {
            $query .= ",".$value2;
        }
        $query .= ")";
        mysql_query($query);
    }
    $DB->CloseConnection();
}
```

Gambar 4.5 Source Code Fungsi Simpan *Pro*

4.2.2 Implementasi *Similarity Item-rating*

Tahap *similarity item-rating* merupakan proses yang bertujuan untuk mendapatkan nilai *similarity* antar *item* satu dengan *item* yang lain. Sesuai dengan persamaan (2.5) pada Bab 2, proses *similarity item-rating* ditunjukkan pada *Source Code* 4.6. Apabila perhitungan pada salah satu akar (*\$jumlahakar1* atau *\$jumlahakar2*) menghasilkan nilai 0, maka hasil *similarity group* akan bernilai 0, jika tidak maka akan dilakukan perhitungan *similarity group*. Hasil perhitungan dari *similarity item-rating* disimpan dalam *array* *\$ResultArray[\$iditem][\$key]['similarity_item']*.

```
function HitungSimilarityItem($iditem, $ItemCurrent,
$DataBandingan, $Average, $User)
{
    $ResultArray = array();
    $jumlah = 0;
    $jumlahakar1 = 0;
    $jumlahakar2 = 0;

    foreach ($DataBandingan as $key => $value)
    {
        foreach ($User as $key2 => $value2)
        {
            if(isset($ItemCurrent[$iditem][$value2]) &&
            isset($value[$value2]))
            {
                $min1 = $ItemCurrent[$iditem][$value2]-
                $Average[$iditem];
                $min2 = $value[$value2]-$Average[$key];
                $perkalian = $min1*$min2;

                $jumlah += $perkalian;
                $jumlahakar1 +=
                pow(($ItemCurrent[$iditem][$value2]-
                $Average[$iditem]), 2);
                $jumlahakar2 += pow(($value[$value2]-
                $Average[$key]), 2);
            }
        }
        if($jumlahakar1 == 0 || $jumlahakar2 == 0)
        {
            $ResultArray[$iditem][$key]['similarity_item']=0;
        }
        else
        {
            $ResultArray[$iditem][$key]['similarity_item'] =
            $jumlah / (sqrt($jumlahakar1)*sqrt($jumlahakar2));
        }
        $jumlah = 0;
        $jumlahakar1 = 0;
        $jumlahakar2 = 0;
    }
    return $ResultArray;
}
```

Gambar 4.6 Source Code Fungsi Hitung Similarity Item-rating

4.2.3 Implementasi *Similarity Group-rating*

Tahap *similarity group-rating* merupakan proses yang bertujuan untuk mendapatkan nilai *similarity* antar *group item* yang satu dengan *group item* yang lain. Sesuai dengan persamaan (2.6) pada Bab 2, proses *similarity group-rating* ditunjukkan pada Gambar 4.7. Apabila perhitungan pada salah satu akar (\$jumlahakar1 atau \$jumlahakar2) menghasilkan nilai 0, maka hasil *similarity group* akan bernilai 0, jika tidak maka akan dilakukan *similarity group-rating*. Hasil perhitungan dari *similarity group-rating* disimpan dalam *array* \$ResultArray[\$iditem][\$key]['similarity_group'] .

```
function HitungSimilarityGroup($iditem, $ItemCurrent,
    $DataBandingan, $Average, $ResultArray)
{
    $jumlah = 0;
    $jumlahakar1 = 0;
    $jumlahakar2 = 0;

    foreach ($DataBandingan as $key => $value)
    {
        for($i=0;$i<$this->JUMLAHKLUSTER;$i++)
        {
            $min1 = $ItemCurrent[$iditem][$i]-
                $Average[$iditem];
            $min2 = $value[$i]-$Average[$key];
            $perkalian = $min1*$min2;

            $jumlah += $perkalian;
            $jumlahakar1 += pow(($ItemCurrent[$iditem][$i]-
                $Average[$iditem]), 2);
            $jumlahakar2 += pow(($value[$i]-$Average[$key]),
                2);
        }
        if($jumlahakar1 == 0 || $jumlahakar2 == 0)
        {
            $ResultArray[$iditem][$key]['similarity_group'] =
                0;
        }
        else
        {
            $ResultArray[$iditem][$key]['similarity_group'] =
                $jumlah / (sqrt($jumlahakar1)*sqrt($jumlahakar2));
        }
    }
}
```

```

    $jumlah = 0;
    $jumlahakar1 = 0;
    $jumlahakar2 = 0;
  }
  return $ResultArray;
}

```

Gambar 4.7 Source Code Fungsi Hitung *Similarity Group-rating*

4.2.4 Implementasi *Linier Combination of Similarity*

Tahap *linier combination of similarity* merupakan proses yang bertujuan untuk mendapatkan nilai *similarity* tiap *item*. Nilai pada proses *linier combination of similarity* didapatkan dengan menggabungkan nilai yang didapat dari proses *similarity item-rating* yang disimpan pada array `$value2['similarity_item']` dan proses *similarity group-rating* yang disimpan pada array `$value2['similarity_group']`. Nilai koefisien *c* yang digunakan pada proses ini ditetapkan sebesar 0,5 yang disimpan pada variabel `C_VALUE`. Implementasi pada langkah ini ditunjukkan pada Gambar 4.8.

```

function hitungSimilarity($ResultSimilarity)
{
    foreach ($ResultSimilarity as $key => $value)
    {
        foreach ($value as $key2 => $value2)
        {
            $ResultSimilarity[$key][$key2]['similarity'] =
                $value2['similarity_item'] * (1-$this->C_VALUE) +
                $value2['similarity_group'] * $this->C_VALUE;
        }
    }
    return $ResultSimilarity;
}

```

Gambar 4.8 Source Code Fungsi Hitung
Linier Combination of Similarity

Dalam aplikasi yang dikembangkan, hasil dari perhitungan *linier combination of similarity* disimpan dalam *database*, yaitu pada tabel `similarity`. Query untuk membuat tabel *similarity* ditunjukkan pada Gambar 4.9. Field

`similarity_item` digunakan untuk menyimpan hasil perhitungan *similarity item*, `field similarity_group` digunakan untuk menyimpan hasil perhitungan *similarity group*, sedangkan `field similarity` digunakan untuk menyimpan hasil perhitungan *linier combination of similarity*.

```
CREATE TABLE IF NOT EXISTS `similarity`
(
  `item_primary` int(11) NOT NULL,
  `item_secondary` int(11) NOT NULL,
  `similarity_item` float NOT NULL,
  `similarity_group` float NOT NULL,
  `similarity` float NOT NULL,
  PRIMARY KEY (`item_primary`,`item_secondary`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Gambar 4.9 Source Code Create Tabel Similarity

Untuk menyimpan hasil perhitungan *linier combination of similarity* menggunakan fungsi `SimpanSimilarity`. Implementasi pada langkah ini ditunjukkan pada Gambar 4.10.

```
function SimpanSimilarity($ResultSimilarity)
{
  $DB = new Database();

  $con= $DB->OpenConnection();
  foreach ($ResultSimilarity as $key1 => $item)
  {
    foreach ($item as $key2 => $data)
    {
      $query = "INSERT INTO similarity
values('".$key1."', '".$key2."',
'".$data['similarity_item']."',
'".$data['similarity_group']."',
'".$data['similarity']."'");

      mysql_query($query);
    }
  }
  $DB->CloseConnection();
}
```

Gambar 4.10 Source Code Fungsi Simpan Similarity

4.2.5 Implementasi Prediksi Rating

Tahap prediksi rating merupakan proses yang bertujuan untuk mendapatkan nilai prediksi rating pada *item*. Perhitungan prediksi rating dilakukan pada *item* yang sudah pernah di rating oleh *user* dan pada *item* yang belum pernah di rating oleh *user*. Proses perhitungan prediksi dilakukan oleh fungsi *process*. Apabila hasil prediksi rating melebihi skala rating (nilai rating > 5), rating akan diset 5. Implementasi pada langkah ini ditunjukkan pada Gambar 4.11.

```
function process($item_id, $averageRating)
{
    $notRating = $this->getUserNotRating($item_id);
    $Datasimilarity = $this->getSimilarity($item_id);
    $RatingUser = $this->getRatingUser();

    $jumUserNotRating = count($notRating);
    $jumUser = $this->getJumlahUser();

    $Result = array();

    if($jumUserNotRating != $jumUser)
    {
        //jika item yang sudah pernah di rating oleh user lain
        foreach ($notRating as $key1 => $user_id)
        {
            $jumlahAtas = 0;
            $jumlahBawah = 0;
            foreach ($RatingUser[$user_id] as $keyItem_id =>
                $valueRating)
            {
                $similarityValue = $Datasimilarity[$keyItem_id];
                $jumlahAtas += ($valueRating -
                    $averageRating[$keyItem_id]) * $similarityValue;
                $jumlahBawah += abs($similarityValue);
            }

            $Result[$user_id] = $averageRating[$item_id] +
                ($jumlahAtas / $jumlahBawah);
        }
    }
    else
    {
        //jika item belum di rating sama sekali
        foreach ($notRating as $key1 => $user_id)
        {
```

```

$jumlahAtas = 0;
$jumlahBawah = 0;

foreach ($RatingUser[$user_id] as $keyItem_id =>
$valueRating)
{
    $similarityValue = $Datasimilarity[$keyItem_id];
    $jumlahAtas += $valueRating * $similarityValue;
    $jumlahBawah += abs($similarityValue);
}

$Result[$user_id] = $averageRating[$item_id] +
($jumlahAtas / $jumlahBawah);
}
}
if ($Result[$user_id] >= 5) { $Result[$user_id]=5; }
return $Result;
}

```

Gambar 4.11 Source Code Fungsi Hitung Prediksi Rating

Hasil dari perhitungan prediksi rating disimpan dalam *database*, yaitu pada tabel *prediction*. *Query* untuk membuat tabel *prediction* ditunjukkan pada Gambar 4.12. *Field* *user_id* digunakan untuk menyimpan *id user*, *field* *item_id* digunakan untuk menyimpan *id item*, sedangkan *field* *prediction_value* digunakan untuk menyimpan nilai prediksi rating.

```

CREATE TABLE IF NOT EXISTS `prediction` (
  `user_id` int(11) NOT NULL,
  `item_id` int(11) NOT NULL,
  `prediction_value` float NOT NULL,
  PRIMARY KEY (`item_id`, `user_id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

Gambar 4.12 Create Tabel Prediction

Untuk menyimpan hasil perhitungan prediksi, menggunakan fungsi *saveRekomendasi*. Implementasi pada langkah ini ditunjukkan pada Gambar 4.13.


```
function saveRekomendasi($item_id, $Result)
{
    $DB = new Database();

    $con = $DB->OpenConnection();
    foreach ($Result as $user_id => $rating)
    {
        $query = "INSERT INTO prediction_new
        VALUES('.$user_id.', ".$item_id.', ".$rating.)";

        mysql_query($query);
    }

    $DB->CloseConnection();
}
```

Gambar 4.13 Source Code Fungsi Simpan Rekomendasi

4.3 Implementasi Antarmuka

Implementasi antarmuka pada aplikasi sistem rekomendasi film terdapat 4 antarmuka, yaitu antarmuka sistem, *home*, detail *item* dan antarmuka rekomendasi.

4.3.1 Antarmuka Sistem

Antarmuka sistem menampilkan proses-proses yang ada pada sistem rekomendasi. Proses-proses tersebut yaitu *config*, *clustering*, *similarity* dan prediksi. Menu *config* berisi *input* jumlah *cluster*, maksimum iterasi dan koefisien *c*. Menu *clustering* menampilkan hasil perhitungan *clustering*. Menu *similarity* menampilkan hasil perhitungan *similarity*. Sedangkan menu prediksi menampilkan hasil perhitungan prediksi.

Adapun antarmuka sistem ditunjukkan pada Gambar 4.14 berikut.

Sistem Rekomendasi Film

Config

Clustering

Similarity

Prediksi

Hasil Pro

Show10entries

Search:

Movie Title	C-1	C-2	C-3	C-4	C-5	C-6	C-7	C-8	C-9	C-10
'Til There Was You (1997)	0.0443	0.0162	0.2684	0.0384	0	0.0915	0.643	0.3505	0.4628	0.083
1-900 (1994)	0.1114	0.1017	0	0.0946	0.0487	0.2027	0.2861	0.612	0.1613	0.1912
101 Dalmatians (1996)	0.4168	0.3673	0	0.0676	0.0409	0.166	0.203	0.3534	0.1454	0.5665
12 Angry Men (1957)	0.061	0.0056	0.3742	0.0086	0	0.1259	0.5406	0.2636	0.9356	0.1327
187 (1997)	0.061	0.0056	0.3742	0.0086	0	0.1259	0.5406	0.2636	0.9356	0.1327
2 Days in the Valley (1996)	0.0508	0.0239	0	0.0251	0.1536	0.1904	0.0244	0.3177	0.1447	0.1629
20,000 Leagues Under the Sea (1954)	0.4309	0.1476	0	0.0437	0.0396	0.2094	0.0436	0.1808	0.0992	0.1298
2001: A Space Odyssey (1968)	0	0.0226	0.3808	0.4106	0.15	0.1972	0.1814	0.0927	0.2509	0.033
3 Ninjas: High Noon At Mega Mountain (1998)	0.4035	0.1222	0	0.0596	0.0063	0.3689	0.0429	0.2314	0.1299	0.1763
39 Steps, The (1935)	0.0411	0.0784	0.3369	0.3423	0.2133	0.4481	0	0.2945	0.1122	0.1505

Showing 1 to 10 of 1,682 entries

First

Previous

1

2

3

4

5

Next

Last

Hasil Final Cluster

Show10entries

Search:

Cluster

Movie Title

Copycat (1995) ; From Dusk Till Dawn (1996) ; Taxi Driver (1976) ; Crimson Tide (1995) ; Desperado (1995) ; Disclosure (1994) ; Dolores Claiborne (1994) ;

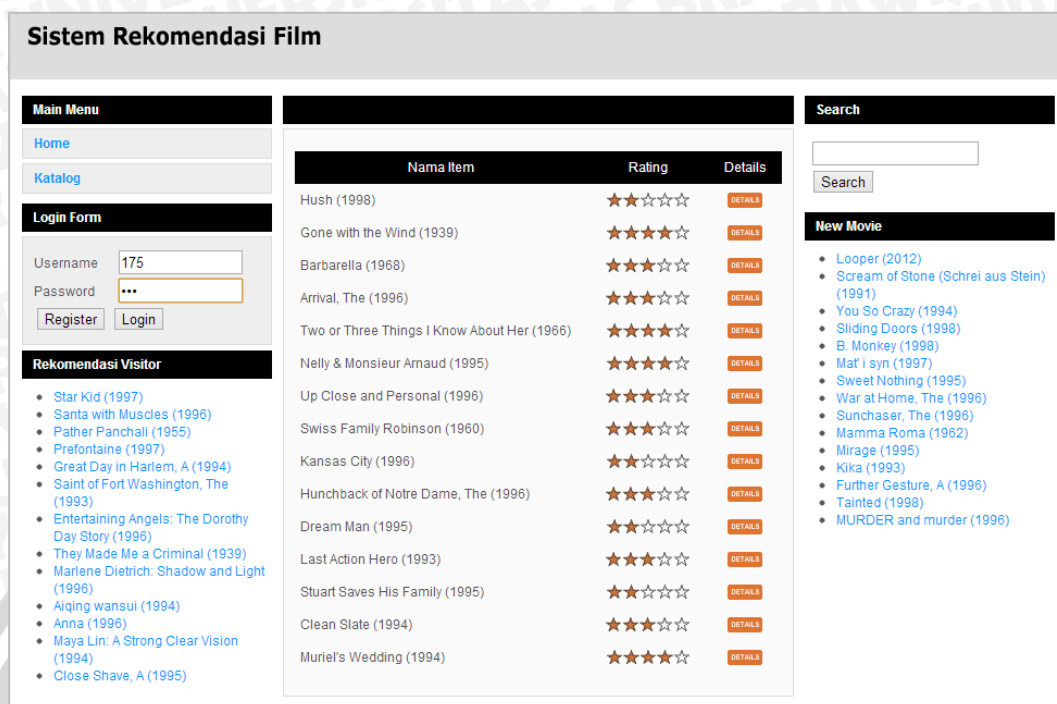
Gambar 4.14 Antarmuka Sistem

4.3.2 Antarmuka Home

Antarmuka *home* merupakan tampilan awal sistem saat *user* mengakses aplikasi. Pada antarmuka *home* terdapat sejumlah daftar *item* dimana setiap *item* yang ditampilkan disertai dengan besarnya rating dan *link* untuk melihat detail *item*.

Pada antarmuka *home* terdapat menu *katalog* dan *search*. Menu *katalog* berfungsi untuk menampilkan *katalog* semua *item* sedangkan menu *search* digunakan *user* untuk mencari suatu *item*. Dibawah menu *search* terdapat menu *new movie* yang digunakan untuk menampilkan *item* yang baru ditambahkan. Pada antarmuka *home* juga terdapat menu *login* yang digunakan oleh *user* untuk *login* pada sistem dengan memasukkan *username* dan *password*, serta menu rekomendasi *Non-Personalized* yang digunakan untuk menampilkan *item* yang direkomendasikan kepada *user* yang tidak terdaftar atau tidak *login*.

Adapun antarmuka *home* ditunjukkan pada Gambar 4.15 berikut.



Gambar 4.15 Antarmuka *Home*

4.3.3 Antarmuka *Detail Item*

Antarmuka detail *item* menampilkan detail informasi dari sebuah *item* yang mencakup *title*, *duration*, *description*, *director writers*, *starts* dan *full details item*. Pada antarmuka detail terdapat rating dari *item* serta jumlah *user* yang telah memberi rating pada *item* tersebut. Rating dari *item* digambarkan dengan jumlah bintang yaitu sebanyak 5 bintang. Pada antarmuka ini juga terdapat menu untuk *user* memberikan rating pada *item*, dengan terlebih dahulu *user* (terdaftar) melakukan *login*.

Adapun antarmuka detail *item* ditunjukkan pada Gambar 4.16 berikut.

2001: A Space Odyssey (1968)

★★★★☆ dirating oleh 259 user

Beri rating : ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5



STANLEY KUBRICK'S
2001:
a space odyssey

Duration: 141 min

Description: Humanity finds a mysterious, obviously artificial, object buried beneath the Lunar surface and, with the intelligent computer H.A.L. 9000, sets off on a quest.

Director: Stanley Kubrick

Writers: Stanley Kubrick (screenplay), Arthur C. Clarke (screenplay), 1 more credit »

Stars: Keir Dullea, Gary Lockwood, William Sylvester |
See full cast and crew

Full Details: [http://us.imdb.com/M/title-exact?2001%3A%20A%20Space%20Odyssey%20\(1968\)](http://us.imdb.com/M/title-exact?2001%3A%20A%20Space%20Odyssey%20(1968))

Gambar 4.16 Antarmuka Detail

4.3.4 Antarmuka Rating User

Pada antarmuka rating *user* menampilkan daftar *item* yang telah di rating oleh *user* beserta rating yang diberikan oleh *user* pada *item* tersebut. Dalam antarmuka rating *user* juga terdapat jumlah *item* yang telah di rating beserta rata-rata ratingnya.

Adapun antarmuka rating *user* ditunjukkan pada Gambar 4.17 berikut.

Anda login sebagai 175

Anda telah merating 38 item, dengan rata-rata 3.74

Id Item	Nama Item	Rating / 5
9	Dead Man Walking (1995)	4
11	Seven (Se7en) (1995)	5
12	Usual Suspects, The (1995)	4
31	Crimson Tide (1995)	4
50	Star Wars (1977)	5
56	Pulp Fiction (1994)	2
64	Shawshank Redemption, The (1994)	5
71	Lion King, The (1994)	4
88	Sleepless in Seattle (1993)	4
96	Terminator 2: Judgment Day (1991)	3

1 2 3 4 Next >>

Gambar 4.17 Antarmuka Rating User

4.3.5 Antarmuka Rekomendasi

Pada antarmuka rekomendasi menampilkan daftar *item* yang direkomendasikan kepada *user*. Setiap *item* yang direkomendasikan kepada *user* disertai besarnya nilai prediksi dan ditampilkan dalam urutan dari nilai prediksi terbesar sampai dengan terkecil. Jumlah *item* yang direkomendasikan kepada *user* dibatasi sebanyak 15 *item*.

Adapun antarmuka rekomendasi ditunjukkan pada Gambar 4.18 berikut.

Anda login sebagai 175

Id Item	Movie Title	Nilai Prediksi
1612	Leading Man, The (1996)	4.79
1467	Saint of Fort Washington, The (1993)	4.79
1450	Golden Earrings (1947)	4.79
1653	Entertaining Angels: The Dorothy Day Story (1996)	4.79
1122	They Made Me a Criminal (1939)	4.79
1201	Marlene Dietrich: Shadow and Light (1996)	4.79
1472	Visitors, The (Visiteurs, Les) (1993)	4.79
1536	Aiqing wansui (1994)	4.79
814	Great Day in Harlem, A (1994)	4.79
1293	Star Kid (1997)	4.79
1500	Santa with Muscles (1996)	4.79
1189	Prefontaine (1997)	4.79
851	Two or Three Things I Know About Her (1966)	4.79
1599	Someone Else's America (1995)	4.79
1449	Pather Panchali (1955)	4.48

Gambar 4.18 Antarmuka Rekomendasi