

## IMPLEMENTASI METODE COSINE SIMILARITY UNTUK REKOMENDASI PRODUK PADA APLIKASI PENJUALAN BERBASIS MOBILE

M.Bagus Sujasman<sup>1</sup>, Diana<sup>2</sup>, Ahmad Syazili<sup>3</sup>

Fakultas Komputer, Universitas Binadarma

E-mail: <sup>1</sup>[mbagussujasman27@gmail.com](mailto:mbagussujasman27@gmail.com), <sup>2</sup>[diana@binadarma.ac.id](mailto:diana@binadarma.ac.id),  
<sup>3</sup>[syazili@binadarma.ac.id](mailto:syazili@binadarma.ac.id)

### ABSTRAK

Pada sistem penjualan online, seringkali produk yang dicari oleh pembeli tidak terdapat pada sistem, namun ada produk yang memiliki kemiripan dengan produk yang dicari. Sistem rekomendasi ini memberikan informasi kemiripan produk dengan produk lain, contohnya shampo, obatan, dan produk perawatan lainnya, kemiripan tersebut dapat dijadikan perbandingan sehingga membuka kemungkinan seseorang menyukai produk yang ditawarkan. Langkah penelitian ini mengikuti langkah pada metode pengembangan aplikasi RUP yakni *inception, elaboration, construction dan transition*. Hasil penelitian ini berupa aplikasi rekomendasi produk dengan menerapkan metode *Cosine Similarity* berbasis mobile. Pada tahap uji coba, dilakukan ujicoba efektifitas aplikasi dengan melakukan 5 kali uji coba dengan melihat kemiripan komposisi produk. Hasil uji coba menunjukkan bahwa hasil perhitungan persentase kemiripan produk secara manual sama dengan hasil perhitungan persentase kemiripan produk pada aplikasi rekomendasi. Metode *Cosine Similarity* dapat diterapkan pada aplikasi rekomendasi produk berbasis *mobile* untuk mencari angka persentase kemiripan produk dengan hasil kemiripan yang tinggi.

**Kata kunci:** *Rekomendasi Produk, Cosine Similarity, Aplikasi Mobile*

### ABSTRACT

*In an online sales system, often the product sought by the buyer is not found in the system, but there are products that have similarities to the product sought. This recommendation system provides product similarity information with other products, for example shampoo, medicine, and other care products, the similarity can be used as a comparison so that it opens the possibility for someone to like the product offered. This research step follows the steps in the RUP application development method namely inception, elaboration, construction and transition. The results of this study are in the form of a product recommendation application by applying the Cosine Similarity method based on mobile. In the trial phase, the effectiveness of the application is tested by conducting 5 trials to see the similarity of the product composition. The trial results show that the calculation result of product prosentase similarity manually is the same as the result of calculation of product similarity prosentase in the recommendation application. Cosine Similarity method can be applied to a product recommendation based mobile application to search for similarities percentage point product with high similarity results.*

**Keywords:** *Product Recommendations, Cosine Similarity, Mobile Applications*

## 1. PENDAHULUAN

Sistem rekomendasi merupakan suatu aplikasi untuk menyediakan dan merekomendasikan suatu item dalam membuat suatu keputusan yang diinginkan oleh pengguna. [1]. Sistem rekomendasi sudah banyak digunakan pada website yang menjual produk atau layanan dalam jumlah banyak namun setiap produk atau layanan tersebut memiliki kemiripan-kemiripan dengan produk lain, contohnya shampo, obatan, dan produk perawatan lainnya. Kemiripan tersebut dapat dijadikan pembandingan untuk menilai kemungkinan seseorang menyukai produk. Dalam kasus ini, metode *Cosine Similarity* dapat diimplementasikan untuk menghitung nilai kemiripan antar kalimat dan menjadi salah satu teknik untuk mengukur kemiripan teks.

*Cosine Similarity* adalah perhitungan kesamaan antara dua vektor n dimensi dengan mencari kosinus dari sudut diantara keduanya dan sering digunakan untuk membandingkan dokumen dalam *text mining*[2]. Untuk penelitian ini, peneliti akan mengambil metode *Cosine Similarity* untuk mengukur kemiripan suatu produk dengan produk lainnya. Karena metode *Cosine Similarity* bekerja dengan menghitung tiap-tiap bobot suatu produk yang nantinya diperoleh dari p term/token (TF-IDF). Hasil dari metode tersebut merupakan barang-barang yang selanjutnya direkomendasikan, yang mana diharapkan pembeli dapat membeli produk yang memiliki kandungan atau komposisi atau bahan yang sama dengan merek yang berbeda pada setiap produk. Hal ini juga diharapkan mampu meningkatkan angka penjualan nantinya. *Cosine similarity* adalah perhitungan kesamaan antara dua vektor n dimensi dengan mencari kosinus dari sudut diantara keduanya dan sering digunakan untuk membandingkan dokumen dalam *text mining* [2].

$$Sim = \frac{\sum_{k=1}^l weight_{lk} * weight_{qk}}{\sqrt{\sum_{k=1}^l (weight_{lk}^2 * weight_{qk}^2)}} \quad (1)$$

Dimana :  $weight_{lk}$  = bobot dari setiap dokumen

$weight_{qk}$  = invest frekuensi dokumen  $\log_{10} (n/df)$

Untuk membandingkan suatu dokumen, tentunya dibutuhkan informasi sebagaimana informasi tersebut dapat digunakan untuk dijadikan pembandingan untuk mencari kemiripan suatu produk. Informasi atau data yang dicari dapat berupa berupa teks, image, audio, video dan lain-lain. Koleksi data teks yang dapat dijadikan sumber pencarian juga dapat berupa pesan teks, seperti e-mail, fax, dan dokumen berita, bahkan dokumen yang beredar di internet. Dengan jumlah dokumen koleksi yang besar sebagai sumber pencarian, maka dibutuhkan suatu sistem yang dapat membantu user menemukan dokumen yang relevan dalam waktu yang singkat dan tepat.

Berdasarkan referensi dijelaskan bahwa *Information Retrieval* merupakan suatu pencarian informasi yang didasarkan pada suatu *query* yang diharapkan dapat memenuhi keinginan *user* dari kumpulan dokumen yang ada. *Information Retrieval* adalah materi pencarian (biasanya dokumen) yang bersifat tidak terstruktur (biasanya teks) yang memenuhi kebutuhan informasi dari dalam kumpulan informasi yang besar (biasanya disimpan di komputer)[3].

*Information Retrieval* (IR) adalah Pencarian informasi berkaitan dengan proses yang terlibat dalam representasi, penyimpanan, mencari dan menemukan informasi yang relevan dengan kebutuhan informasi yang diinginkan oleh pengguna manusia. IR adalah bidang penelitian inti dalam ilmu informasi. Tujuan adalah untuk mempelajari dan memahami proses IR untuk merancang, membangun dan pengambilan tes sistem yang dapat memfasilitasi komunikasi yang efektif dari informasi yang diinginkan antara generator manusia dan pengguna manusia[4].

Setelah informasi serta *query* didapatkan, untuk memproses penghitungan kemiripan pada suatu produk dibutuhkan metode penghitungan nilai frekuensi dari suatu produk. Untuk itu, penggunaan TF-IDF bisa digunakan dalam penghitungan nilai tersebut. *Term Frequency* dan *Document Frequency* (TF-IDF) digunakan untuk menentukan nilai frekuensi sebuah kata dalam

sebuah dokumen atau artikel dan juga frekuensi di dalam banyak dokumen. Perhitungan ini menentukan seberapa *relevan* sebuah kata di dalam sebuah dokumen. Penggunaan *term frequency* dan *Dument frequency* untuk memberikan peringkat pada dokumen secara ekstensif, khususnya oleh [5], untuk model ruang vektor. [6] berpendapat bahwa kata yang muncul dalam dokumen harus diberi nilai sebanding dengan frekuensi kata dan terbalik sebanding dengan frekuensi dokumen. Pembobotan skema yang mengikuti pendekatan ini disebut *TF x IDF* (*term frequency x inverse Dument frequency*).

Proses dalam implementasi TF-IDF terdapat perbedaan kecil di dalam semua aplikasinya, tetapi pendekatannya kurang lebih sama. *TF* (*Term Frequency*) merupakan frekuensi sebuah kata terdapat di dalam sebuah dokumen. Nilai dari *TF* didapat menggunakan persamaan :

$$TF(t) = f_{t,d} / \sum t, d \quad (2)$$

dimana  $f_{t,d}$  merupakan frekuensi sebuah kata ( $t$ ) muncul di dalam dokumen  $d$ , sedangkan  $\sum t, d$  merupakan total keseluruhan kata yang terdapat di dalam dokumen  $d$ . Kemudian untuk menghitung nilai *IDF* (*inverse Dument frequency*) dari sebuah kata di dalam kumpulan dokumen menggunakan persamaan :

$$IDF(t) = \log(|D|/f_{t,D}) \quad (3)$$

$|D|$  merupakan jumlah dokumen yang ada dalam koleksi, sedangkan  $f_{t,D}$  merupakan jumlah dokumen dimana / muncul di dalam  $D$ [5]. Dalam koleksi dokumen  $D$ , sebuah kata  $t$  dan dokumen individu  $d \in D$ , dapat dihitung nilai *TF-IDF* menggunakan rumus :

$$TF - IDF(t) = TF(t) * IDF(t) \quad (4)$$

Diasumsikan bahwa  $|D| \sim f_{t,D}$ , ukuran dari kumpulan dokumen hampir sama dengan frekuensi  $t$  di dalam  $D$ . Jika  $1 < \log(|D|/f_{t,D}) < c$  untuk sebuah konstanta  $c$  dengan nilai yang kecil, maka  $w_d$  akan lebih kecil daripada  $f_{t,d}$  tetapi tetap bernilai positif. Hal ini berarti bahwa  $w$  merupakan relasi yang biasa dengan seluruh dokumen tetapi tetap menyimpan beberapa informasi yang penting [7].

## 2. METODOLOGI PENELITIAN

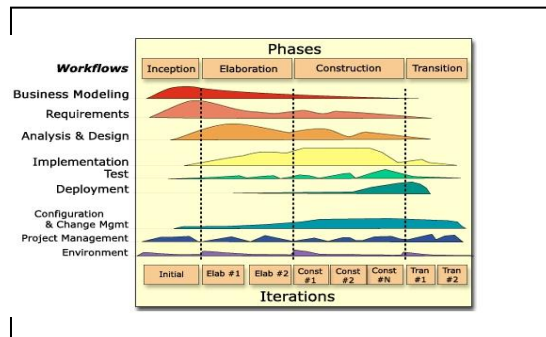
### 2.1. Data penelitian

Data set pada penelitian ini adalah contoh produk-produk yang akan dijual. Data penelitian berupa data kode produk, nama produk, keterangan atau komposisi, dan gambar.

### 2.2. Metode Pengembangan Aplikasi

Langkah-langkah penelitian mengikuti tahapan pengembangan sistem *Rational Unified Process* (RUP). [8], RUP merupakan proses rekayasa perangkat lunak yang memberikan pendekatan terdisiplin untuk menugaskan tugas dan tanggung jawab dalam organisasi pengembangan. Tujuannya adalah untuk memastikan produksi perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna akhir, dalam jadwal dan anggaran yang dapat diprediksi. RUP menurut [9] bukanlah model pengembangan yang konkrit. Melainkan dimaksudkan untuk menjadi adaptif dan disesuaikan dengan kebutuhan spesifik proyek, tim, atau organisasi. *Rational Unified Process* didasarkan pada beberapa gagasan mendasar, seperti fase pembangunan dan blok-blok pembangun, yang menentukan siapa, apa, kapan, dan bagaimana pembangunan akan terjadi.

Dalam bentuk yang paling sederhana RUP terdiri dari beberapa tahapan kerja dasar, dalam gambar ditunjukkan pada poros vertikal (*organization along content*), untuk lebih jelasnya berikut adalah gambar denah RUP dalam pengembangan pada perangkat lunak:



**Gambar 1.**Fase dan Model Proses RUP

Dengan menggunakan model ini, RUP membagi tahapan pengembangan perangkat lunak ke dalam 4 fase sebagai berikut:

- Inception*, merupakan tahap untuk mengidentifikasi sistem yang akan dikembangkan. Aktivitas yang dilakukan pada tahap ini antara lain mencakup penentuan arsitektur global, identifikasi kebutuhan, perumusan persyaratan fungsional, konsep GUI serta perumusan kebutuhan pengujian level performansi dan fungsionalitas aplikasi rekomendasi produk.
- Elaboration*, merupakan tahap untuk melakukan desain secara lengkap berdasarkan hasil analisis di tahap *inception*. Aktivitas yang dilakukan pada tahap ini antara lain mencakup pembuatan *database*, desain dan antarmuka/tampilan.
- Construction*, merupakan tahap untuk mengimplementasikan hasil desain dan melakukan pengujian hasil implementasi. Pada tahap awal *construction*, ada baiknya dilakukan pemeriksaan ulang hasil analisis dan desain. Apabila desain yang dibuat telah sesuai dengan analisis sistem, maka implementasi dengan bahasa pemrograman tertentu dapat dilakukan. Aktivitas yang dilakukan pada tahap ini antara lain mencakup pengujian hasil analisis, pendataan kebutuhan implementasi lengkap (berpedoman pada identifikasi kebutuhan di tahap analisis), penentuan *coding pattern* yang digunakan, pembuatan program, penulisan kode metode cosine similarity, pengujian pencarian produk dan melihat hasil rekomendasi serta pendataan berbagai kemungkinan pengembangan / perbaikan lebih lanjut.
- Transition*, merupakan tahap untuk menyerahkan sistem aplikasi ke konsumen (*roll-out*), yang umumnya mencakup pelaksanaan pelatihan kepada pengguna dan testing beta aplikasi terhadap ekspektasi pengguna.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Koding Cosine Similarity

Sebelum mengimplementasikan Cosine Similarity dibutuhkan kode program untuk menghitung TF IDF agar aplikasi penjualan dapat berjalan dengan baik, berikut merupakan kode menghitung tf-idf :

```
foreach($token as $tokenname => $tot)
{
    $total=0;
    foreach($d as $docid => $deskr)
    {
        $jml[$docid] = $this->get_tf($d[$docid],$tokenname);
        $jml = $jml[$docid];
        // if ($docid!=0)
        // {
        //     if ($jml >= 1)
        //     {
        //         $jt=1;
        //     }
        //     else
        //     {
        //         $jt=0;
        //     }
        // }
        $token[$tokenname]+=$jt;
    }
}

$idf = $this->IDF($total,$token[$tokenname]);
$sql = "insert into idf(term,idf)values('$tokenname','$idf')";
mysql_query($sql)or die(trigger_error(mysql_error()));

foreach($d as $docid => $deskr)
{
    $wd[$tokenname][$docid] = $this->bobot($jml[$docid],$idf);
    $sql = "insert into bobot(term_id_doc,bobot)values('$tokenname','$docid','$wd[$tokenname][$docid]' .
    mysql_query($sql)or die(trigger_error(mysql_error()));
}
}
```

Gambar 2. Tampilan Kode Menghitung TF-IDF

Dengan kode yang dibuat pada gambar diatas, kita akan mendapatkan hasil dari nilai TF-IDF dari suatu produk pada aplikasi penjualan. Hasil dari nilai TF-IDF ini nantinya akan digunakan untuk mencari nilai kemiripan atau mencari nilai *cosine similarity* pada aplikasi penjualan.

Setelah kode tf-idf dibuat, barulah kita membuat kode untuk rumus *cosine similarity*. Kode *cosine similarity* dibuat bertujuan untuk mencari nilai kemiripan suatu produk dengan produk lainnya. Untuk lebih jelasnya, berikut gambar untuk kode *cosine similarity*:

```
$w = $this->total_bobot($wd,$token,$count($d));
$wv = array();
foreach($token as $tokenname => $tot)
{
    foreach($d as $docid => $deskr)
    {
        $wv[$tokenname][$docid] = $wd[$tokenname][$docid];
        $wv[$tokenname][$docid] = $wv[$tokenname][$docid]*$wv[$tokenname][$docid];
    }
}

//var_dump($wv);
$vector = $this->total_vector($wv,$token,$d);
//var_dump($vector);
$vectorscalar = $this->total_jarakvector($wd,$token,$d);

$fd = array();
foreach($d as $docid => $deskr)
{
    if ($docid!="0")
    {
        $sh = (sqrt($vector["0"])*sqrt($vector[$docid]));
        if ($sh!=0){
            $fd[$docid]= $vectorscalar[$docid]/$sh;
        }
        else {
            $fd[$docid]=0;
        }
    }
}

$fd=$this->sorting_desc($fd);
```

Gambar 3. Tampilan Koding Metode Cossin Similarity

Tahapan pengujian pada aplikasi penjualan pada android ini meliputi beberapa aspek yaitu halaman *Splash Screen* pada tampilan ini akan menunjukkan tampilan halaman awal saat membuka aplikasi penjualan online, selanjutnya ada halaman *login* pada tampilan ini akan menunjukkan halaman untuk melakukan *login*, selanjutnya ada halaman pendaftaran dimana tampilan ini akan menunjukkan halaman untuk melakukan pendaftaran atau registrasi untuk pelanggan baru, selanjutnya ada halaman daftar produk/item pada tampilan ini akan menunjukkan daftar produk yang akan dijual, selanjutnya ada tombol pencarian produk pada tampilan ini merupakan proses untuk mencari produk atau query, dan selanjutnya ada halaman hasil Cosine Similarity pada tampilan ini menunjukkan hasil persentase dari pencarian produk atau query

### 3.2. Perhitungan Cosine Similarity

Pada bagian ini diuraikan simulasi perhitungan implementasi metode Cosine Similarity untuk melihat kemiripan antara beberapa produk berdasarkan komposisi masing-masing produk.

**Tabel 1.** Data Produk

Kode	Nama Produk	Keterangan / Komposisi
Q	Dove Shampoo	Sodium, laureth, sulfate
D1	Pantene Shampo	Ammonium, lauryl, laureth
D2	Zinc Shampo	Sodium, sulfata, lauryl

Berdasarkan komposisi produk pada tabel produk diatas diperoleh 5 buah term, yakni sodium, laureth, sulfata, ammonium dan lauryl.

**Tabel 2.** Frekuensi Term Dalam Dokumen

No	Term	Q	D1	D2	F <sub>t,d</sub>
1	Amonium	0	1	0	1
2	Laureth	1	1	0	2
3	Laury	0	1	1	2
4	Sodium	1	0	1	2
5	Sulfate	1	0	1	2

Angka 1 memiliki arti bahwa term pada query terdapat pada dokumen, sebaliknya angka 0 memiliki arti bahwa term pada query tidak terdapat pada dokumen. Jumlah dokumen ada 3 buah, sehingga nilai D = 3, selanjutnya mencari nilai IDF, menggunakan persamaan (2).

$$IDF(Amonium) = \log\left(\frac{3}{1}\right) = 0,477$$

$$IDF(Laureth) = \log\left(\frac{3}{2}\right) = 0,176$$

$$IDF(Laury) = \log\left(\frac{3}{2}\right) = 0,176$$

$$IDF(Sodium) = \log\left(\frac{3}{2}\right) = 0,176$$

$$IDF(Sulfate) = \log\left(\frac{3}{2}\right) = 0,176$$

Nilai IDF pada tiap term yang telah diperoleh melalui  $\log(n/F_{t,d})$  akan mengganti nilai 1 pada tiap term yang ada pada tabel 2.

**Tabel 3. Frekuensi Term IDF**

No	Term	Q	D1	D2
1	Amonium	0	0,477	0
2	Laureth	0,176	0,176	0
3	Laury	0	0,176	0,176
4	Sodium	0,176	0	0,176
5	Sulfate	0,176	0	0,176

Selanjutnya setelah mendapat nilai IDF, langkah selanjutnya adalah dengan kuadratkan setiap term yang ada pada dokumen lalu dijumlahkan.

**Tabel 4. Kuadrat, Penjumlahan dan Akar Bobot Penyebut**

No	Q	D1	D2
1	0	0,227	0
2	0,031	0,031	0
3	0	0,031	0,031
4	0,031	0	0,031
5	0,031	0	0,031
<b>Jumlah</b>	<b>0,093</b>	<b>0,289</b>	<b>0,093</b>
<b>Diakarkan</b>	<b>0,305</b>	<b>0,537</b>	<b>0,305</b>
<b>jumlah akar (Q*D)</b>		<b>0,164</b>	<b>0,093</b>

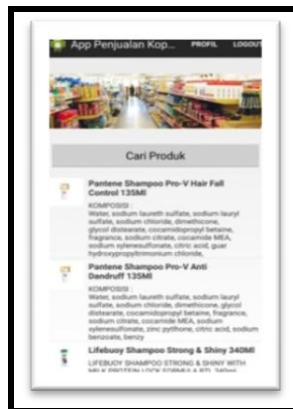
Setelah nilai jumlah akar didapat, maka selanjutnya mencari nilai penjumlahan pada tabel 3 lalu mencari hasil dari cosine similarity.

**Tabel 5. Penjumlahan Bobot Dokumen dan Query**

No	D1	D2
1	0	0
2	0,031	0
3	0	0
4	0	0,031
5	0	0,031
<b>Jumlah</b>	<b>0,031</b>	<b>0,062</b>
<b>Cosim (Jumlah akar tabel 4*jumlah tabel 5)</b>	<b>0,188</b>	<b>0,666</b>
<b>Hasil %</b>	<b>18,89</b>	<b>66,61</b>

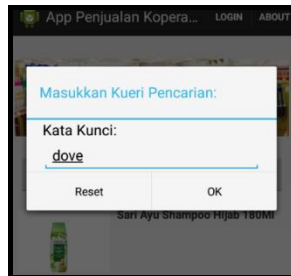
### 3.3. Aplikasi Rekomendasi Produk

Untuk tahap awal pada simulasi pembuktian dalam aplikasi mobile ini adalah dengan membuka aplikasi pada handphone, lalu buka menu awal atau home, seperti yang terlihat pada gambar berikut:



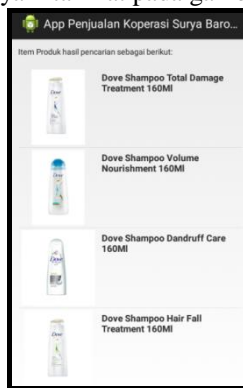
**Gambar 4 Tampilan Awal atau Home**

Selanjutnya kita klik tombol pencarian untuk mencari produk dengan kata kunci nama produk (*Dove*). Untuk lebih jelasnya langsung saja lihat pada gambar aplikasi penjualan berikut ini :



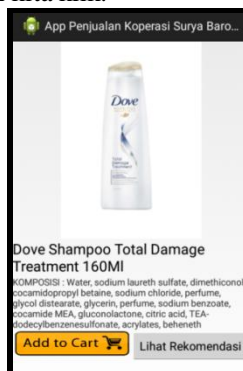
**Gambar 5**Tampilan Pada Pencarian

Selanjutnya kita masukkan kata kunci *dove* sebagai produk yang akan kita cari dan dijadikan *queri*. Setelah kita selesai melakukan pencarian maka akan muncul beberapa produk yang memiliki nama *dove*. Untuk lebih jelasnya kita lihat pada gambar berikut:



**Gambar 6**Tampilan Pencarian Kata Dove

Selanjutnya, karena pada pengujian pada hitungan manual menggunakan *queri Dove Shampoo Total Damage*, maka untuk pembuktian pada aplikasi mobile ini juga akan menggunakan *query Dove Shampoo Total Damage*. Berikut gambar *Dove Shampoo Total Damage* yang ada pada aplikasi penjualan setelah kita klik:



**Gambar 7.**Tampilan Produk yang Menjadi Query

Selanjutnya, pada gambar diatas ada tombol “lihat rekomendasi”, tombol ini merupakan tombol untuk melihat hasil *cosine similarity* pada peroduk *Dove Shampoo Total Damage*. Untuk hasilnya langsung saja lihat pada gambar aplikasi penjualan berikut ini :





**Gambar 8.** Hasil *Cosine Similarity* dari Aplikasi Mobile

Dari hasil gambar diatas, hal ini membuktikan bahwa hasil cosine similarity pada aplikasi mobile, yaitu produk dengan nilai tertinggi pada semua hitungan adalah Dove Shampoo Volume Nourishment dengan persentase 64,5%.

### 3.4. Uji Efektifitas Aplikasi

Uji efektifitas dilakukan untuk melihat seberapa besar tingkat keberhasilan implementasi metode *Cosine Similarity* pada aplikasi yang dibangun. Uji efektifitas aplikasi rekomendasi produk dilakukan dengan melakukan perbandingan hasil 5 kali ujicoba aplikasi dan 5 kali uji coba perhitungan manual.

**Tabel 6.** Uji coba Efektifitas

Uji Coba	Jumlah Term	Jumlah Dokumen (Q,D1,D2)	Hitungan Manual		Perhitungan Aplikasi		Keterangan
			D1	D2	D1	D2	
1	5	3	18,89	66,6	18,89	66,6	Berhasil
2	7	3	18,46	66,17	18,46	66,17	Berhasil
3	9	3	18,03	65,74	18,03	65,74	Berhasil
4	12	3	17,6	65,31	17,6	65,31	Berhasil
5	14	3	17,17	64,87	17,17	64,87	Berhasil

## 4. KESIMPULAN DAN SARAN

Dari hasil penelitian dan pembahasan pada penelitian ini dapat disimpulkan sebagai berikut:

- 1) Metode *Cosine Similarity* dapat diterapkan pada aplikasi penjualan berbasis *mobile* untuk mencari angka persentase kemiripan produk dengan hasil kemiripan yang tinggi.
- 2) Hasil dari persentase kemiripan produk dapat kita jadikan acuan untuk mencari produk lain yang memiliki komposisi yang sama.

## DAFTAR PUSTAKA

- [1] B. T. Wahyu U and A. W. Anggriawan, "Sistem Rekomendasi Paket Wisata Se-Malang Raya Menggunakan Metode Hybrid Content Based dan Collaborative," *J. Ilm. Teknol. dan Inf. ASIA*, 2015.
- [2] Z. Yan, R. Dijkman, and P. Grefen, "Fast business process similarity search with feature-based similarity estimation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in*

- Bioinformatics*), 2010.
- [3] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval Introduction*. 2008.
  - [4] K. Järvelin and P. Ingwersen, "Information seeking research needs extension toward tasks and technology," in *Information Research*, 2004.
  - [5] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manag.*, 1988.
  - [6] M. Dillon, "Introduction to modern information retrieval," *Inf. Process. Manag.*, 2002.
  - [7] J. Ramos, J. Eden, and R. Edu, "Using TF-IDF to Determine Word Relevance in Document Queries," *Rutgers Univ.*, 1999.
  - [8] I. Akil, "Rekayasa Perangkat Lunak Dengan Model Unified Process Studi Kasus : Sistem Informasi Journal," *J. Pilar Nusa Mandiri*, 2016.
  - [9] A. Powell-Morse, "Waterfall Model: What Is It and When Should You Use It?," *Airbrake*, 2016. .