SISTEM PENCARIAN KODE SUMBER BERDASARKAN *CLASS* DAN *METHOD* MENGUUNAKAN METODE PROBABILISTIK

VIRSEP ADITYA

Jurusan Teknik Informatika, Universitas Muhammadiyah Jember
Email: virsep.adityapw50@gmail.com
NIM : 09 1065 1154

Abstrak

Suatu universitas sangat memerlukan saran dan kritik dari para mahasiswabaik mengenai fasilitas, kinerja dosen, keuangan, proses pelayanan, dll.Saran dan kritik tersebut akan digunakan universitas untuk meninjau dan memperbaiki kinerja dan proses pelayanan yang selama ini telah dilakukan demi kepentingan universitas untuk lebih maju lagi kedepannya. Terkadang saran dari mahasiswa tersebut kurang mendapat respon dan tanggapan dari universitas. Saran telah diberikan tetapi untuk waktu yang lama universitas tersebut tidak secara baik meninjau. Hal ini terjadi karena proses yang dilakukan secara manual dan terkadang penggolongan terhadap saran dan kritik terhadap sub bidang proses universitas tersebut sangat sulit untuk diimplementasikan. Hal ini dikarenakan sistem peninjau harus mencari satu persatu saran tersebut secara manual, walaupun telah menggunakan media komputer untuk merekam informasi tersebut. Model Vektor dan Probabilistik sangat tepat digunakan untuk mencari perbandingansaran dan kritik dari mahasiswa pada penentuan kemiripan query dan penggolongan kritik dan saran tersebut. Model yang dikembangkan dan diusulkan telah diterapkan dan di implementasikan pada salah satu perusahaan multinasional di Indonesia yang oleh penulis tidak disebutkan dalam paper ini.

Kata Kunci: probabilistik model, pencarian dokumen.

1. Pendahuluan

Pengembangan perangkat lunak dalam organisasi besar dapat dilakukan secara terpisah dengan team yang berbeda untuk mengembangkan bagian dari perangkat lunak mereka. Masing-masing team memiliki sekumpulan operasi yang perlu dilakukan dalam urutan tertentu untuk membantu suatu proyek perangkat lunak. Pengembang perangkat lunak sering melakukan duplikasi pekerjaan pada bagian kode program yang relatif sama yaitu beberapa orang bekerja pada team yang berbeda dan melakukan pekerjaan yang sama. Oleh karena itu, pengembangan perangkat lunak berdasarkan repositori contoh-contoh kode program dapat mengatasi permasalahan diatas, juga memberikan keuntungan reuse terhadap kode sumber dan efisiensi pekerjaan bagi pengembangan perangkat lunak selanjutnya.

Metode pencarian dokumen kode sumber yang sederhana, namun memiliki akurasi relatif tinggi salah satunya adalah *probabilistik model*. *Probabilistik model* merupakan suatu teknik penghitungan kemiripan yang tidak menggunakan proses normalisasi. Namun, untuk meningkatkan presisi dari dokumen yang diindeks diperlukan suatu sistem pencarian bagian dokumen kode sumber berdasarkan *class* dan *method*.

Penelitian ini mengajukan satu solusi kolaboratif dengan melakukan pencarian contoh-contoh kode sumber pada repositori dokumen kode program menggunakan *Probabilistik model* untuk mengukur derajat kemiripan kata-kata pada suatu dokumen kode sumber. Oleh karena itu penulis mengambil sebuah judul yaitu "SISTEM PENCARIAN KODE SUMBER BERDASARKAN *CLASS* DAN *METHOD* MENGGUNAKAN METODE PROBABILISTIK".

2. Landasan Teori

2.1 Temu Kembali Informasi

Information Retrieval adalah suatu sistem yang menemukan (*retrieve*) informasi yang sesuaidengan kebutuhan pengguna dari kumpulan informasi secara otomatis. Salah satu aplikasi dari *Information Retrieval* adalah mesin pencari yang dapat diterapkan diberbagai bidang. Pada mesin pencari dengan *information retrieval* pengguna dapat memasukkan *query* yang bebas dalam arti kata *query* yang sesuai dengan bahasa manusia dan Sistem dapat menemukan dokumen yang sesuai dengan *query* yang ditulis oleh user.

Pada proposal ini dikenalkan konsep sistem temu kembali (*Information Retrieval System*) dan sistem basis informasi (*Information base system*) yang konsepnya hampir sama dengan basis data, yaitu menyimpan informasi didalamnya (bukan extraksi data). Konsep ini tentunya memiliki kebebasan dalam berekspresi baik dalam penyimpanannya maupun dalam *query* yang diberikan. Pada proposal ini penulis tidak akan mengutarakan lebih jauh tentang konsep yang penulis ajukan (*Information base*) karena akan penulis berikan lebih lanjut pada penelitian penulis berikutnya. Sistem temu kembali informasi melakukan proses-proses tertentu untuk mendapatkan hasil yang baik, beberapa proses yang telah kita kenal adalah *string gathering* atau pemilahan kata-kata dalam sekumpulan *sentence*, penghilangan kata-kata yang memiliki frekuensi kemunculan sangat tinggi (*stoplist*) seperti *in, are,go, is, were, without, break*, dan lain-lain, proses *stemming*, yaitu menghilangkan *suffix, prefix* dan*infix* dari suatu kata, sehingga menjadi kata dasar, yang sangat baik untuk normalisasi dan penyamaan arti kata nantinya. Proses pemberian bobot dan pembuatan index kata.

2.2. Penghimpunan kata (indexing)

subsistem adalah proses subsistem yang mempresentasikan koleksi doumen ke dalam bentuk tertentu untuk memudahkan dan mempercepat proses pencarian dan penentuan kembali dokumen yang benar.

Pembangunan penghimpunan kata dari koleksi dokumen merupakan tugas pokok pada tahapan praproses di dalam temu kembali informasi. Kualitas penghimpunan kata mempengaruhi efektifitas dan efisiensi sistem temu kembali informasi. Penghimpunan kata dokumen adalah himpunan kata yang menunjukkan isi atau topik yang dikandung oleh dokumen. Penghimpunan kata kan membedakan suatu dokumen dari dokumen lain yang berada di dalam koleksi. Ukuran penghimpunan kata yang kecil dapat memberikan hasil buruk dan mungkin beberapa item yang benar terabaikan. Penghimpunan kata yang besar memungkinkan jumlah ditemukan banyak dokumen yang benar tetapi sekaligus dapat menaikkan jumlah dokumen yang tidak benar dan menurunkan kecepatan pencarian, (G. Salton, (1989).

Pembuatan *inverted* penghimpunan kata harus melibatkan konsep praproses yang bertujuan mengekstrak kata-kata penting dari dokumen yang direpresentasikan sebagai *big-of-words.* Ekstraksi kata biasanya melibatkan dua operasi utama berikut:

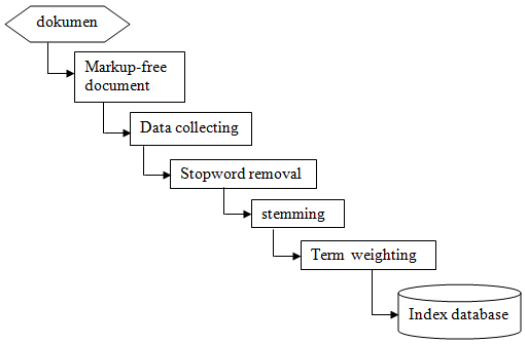
1. Proses pertama yang harus dilakukan adalah pengumpulan data.
2. Penghapusan *(stop word).*
3. Pembentukan kata dasar *(stemming).*

2.3. Konversi Kata Bentuk Dasar *(stemming)*

Stemming adalah proses konversi kata ke bentuk umumnya, sebagaimana dijelaskan sebelumnya. Dokumen dapat pula diekspansidengan mencarikan sinonim bagi kata-kata tertentu di dalamnya. Sinonim adalah kata-kata yang mempunyai pengertian serupa tetapi berbeda dari sudut pandang morfologis. Seperti *stemming,* operasi ini bertujuan menemukan suatu kelompok kata terkait. Akan tetapi sinonim bekerja berdasarkan pada *thesaurus,* tidak berbagi-pakai kata sistem. Jika pengguna memasukkan *query* diekspansi untuk mengakomodasi semua sinonim dari disease seperti *operating, system, windows,* dan lain-lain. Ambara F. Horasi,(1997).

2.4. Pemberian Bobot Terhadap Kata *(Weighting)*

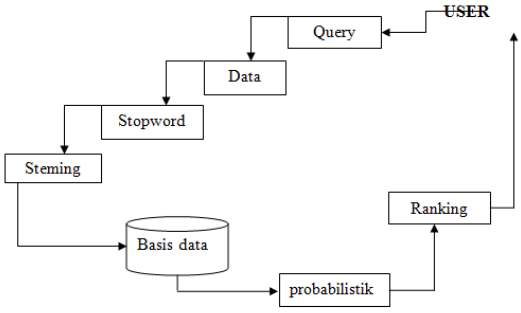
Setiap kata diberikan bobot sesuai dengan skema pembobotan yang dipilih, apakah pembobotan lokal, global atau kombinasi keduanya. Banyak aplikasi menerapkan pembobotan kombinasi berupa perkalian bobot lokal kata *frequency* dan global *inverse* dokumen *frequency,* ditulis *tf.idf.*



Gambar 2.1 Proses Pembobotan Kata

2.5. Proses pencarian

Di bawah ini adalah gambar ilustrasi pada proses pencarian dalam sistem temu kembali informasi.



Gambar 2.2 Proses Pencarian

2.6. Java Source Code Search

Bagian ini menjelaskan penerapan alat JSearch.Sebuah repositori kode sampel yang dibuat oleh pengembang menyerahkan kode mereka.Sebuah pengembang ingin melaksanakan tugas memasuki string kueri ke dalam alat JSearch.Alat ini menyajikan pengguna dengan hasil peringkat mereka berdasarkan pertandingan antara permintaan dan indeks kode sumber. Pengembang scan melalui hasil yang yang berisi potongan kode dan dapat

menggunakannya untuk menyelesaikan tugas atau bagian dari tugas.

2.7. Probabilitas

Dokumen dianggap sebagai sebuah *vektor d* begitu pun *query* dipandang sebagai *vektor q.* Dalam penelitian ini yang dimaksud dengan dokumen adalah dokumen mengenai kritik dan saran mahasiswa, sedangkan *query* adalah masukan kata kunci dari pencari informasi dokumen. (Alberto. M, (1997))

Tingkat kemiripan *query-document* bisa didapatkan dengan membandingkan antara kedua dokumen yang bersesuaian dengan persamaan yang akan digunakan jika perhitungan bobot kata telah dilakukan normalisasi dengan jangkauan nilai bobot 0 sampai dengan 1. Sedangkan bila proses pembobotan tiak menggunakan proses normalisasi maka proses kemiripan menggunakan persamaan Probabilitas.

$$\mathbf{RSV_{t,d}} = \sum_{t \in q} \log\left[\frac{N}{dft}\right] \cdot \frac{(k1+1)TFtd}{k1(1-b)+bx(\frac{Ld}{Lavg}))+TFtd}$$

Keterangan:

RSV_{td} : bobot kata terhadap dokumen tertentu
$_{t \in q}$: kata t merupakan elemen dari *query* q yang diinputkan oleh pengguna
N :jumlah dokumen
df_t : jumlah dokumen yang mengandung kata *t*
tf_{td} : frekuensi kata *t*padadokumen *d*
ld : jumlah kata yang terdapat pada setiap dokumen
lavg : jumlah rata-rata kata pada dokumen
k_1 : konstanta untuk tetapan = 1,2
b : konstanta untuk tetapan = 0,2

Sebuah pencocokan dapat melakukan perbandingan global antara *query* dan dokumen untuk mendapatkan tingkat kemiripan antara keduanya. Menghitung skor dokumen tertentu, dengan rumus:

$$\mathrm{RSV_d} = \sum_{i=1}^{n} RSV_{\mathrm{t,d}}$$

Keterangan:

RSV_d : skor total suatu dokumen
$RSV_{t,d}$: bobot kata t terhadap dokumen d
n : jumlah kata *query* yang diinputkan pengguna

2.8. Ranking

Ranking (perankingan) merupakan pencarian dokumen-dokumen yang benar terhadap *query* dan mengurutkan dokumen tersebut berdasarkan kesesuaiannya dengan *query* (Mandala, 2004). Sistem temu kembali informasi menerima *query* dari pengguna, kemudian melakukan perankingan terhadap dokumen pada koleksi berdasarkan kesesuaiannya denga *query*. Hasil perankingan yang diberikan kepada pengguna merupakan dokumen yang menurut sistem benar dengan *query* dan metode untuk perankingan tersebut dilakukan dengan menghitung nilai kemiripan antara dokumen dan *query.* Setelah dilakukan perankingan, dokumen hasil pencarian akan ditampilkan sesuai urutan rankingnya yaitu semakin tinggi nilai kemiripan sebuah dokumen maka emakin tinggi juga rankingnya.

2.9. Pengujian Precision dan Recall

Ukuran efektifias pencarian ditentukan oleh *precision* dan *recall. Precision* adalah rasio jumlah dokumen benar yang ditemukan dengan total jumlah dokumen yang ditemukan oleh mesin pencari. *Precision* mengindikasikan kualitas himpunan jawaban, tetapi tidak memandang total jumlah dokumen yang benar dalam kumpulan dokumen (Cleland-Huang, 2006).

$$\mathbf{Precision} = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

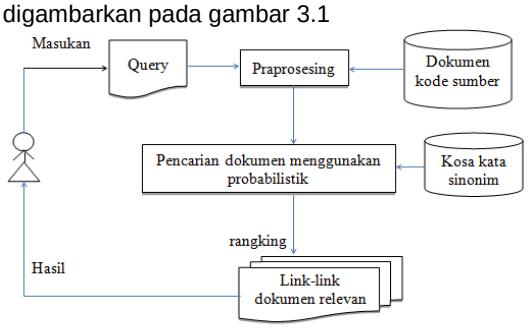
Recall adalah rasio jumlah dokumen benar yang ditemukan kembali dengan total jumlah dokumen dalam kumpulan dokumen yang dianggap benar.

$$\mathbf{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

3. Metodelogi Penelitian

3.1 Arsitektur sistem

Sistem temu kembali informasi memiliki bagian yang membangun sistem secara keseluruhan. Gambaran bagian-bagian yang terdapat pada sistemtemu kembali informasi digambarkan pada gambar 3.1



Gambar 3.1Arsitektur Sistem Pencarian Kode Sumber

4. Hasil Pembahasan

4.1. Skenario Pengujian

Pada penelitian ini digunakan 36 dokumen contoh kode sumber yang berasal dari *algoritma semut dan rekam medis*. Skenario pengujian dilakukan tiga kali percobaan dengan memasukkan *query* pengguna yang berbeda yaitu berdasarkan class, method serta gabungan dari class dan method.

1. Skenario 1
   Memasukkan query berupa nama suatu class untuk memperoleh link dokumen yang berkaitan dengan query.
2. Skenario 2
   Memasukkan query berupa nama suatu method untuk memperoleh link dokumen yang berkaitan dengan query.
3. Skenario 3

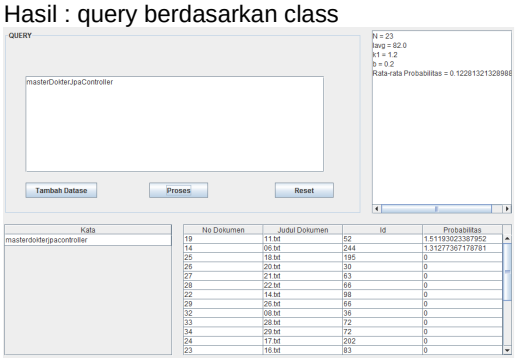
Memasukkan query gabungan antara class dan method untuk memperoleh link dokumen yang berkaitan dengan query.

4.2. Hasil Pengujian

1. Skenario 1
   Memasukkan query berdasarkan class
   Dokumen pengujian yang benar adalah sebagai berikut :

No Dokumen	Judul Dokumen
19	05.txt
14	11.txt

Query 1 : ***masterDokterJpaController***
Hasil : query berdasarkan class



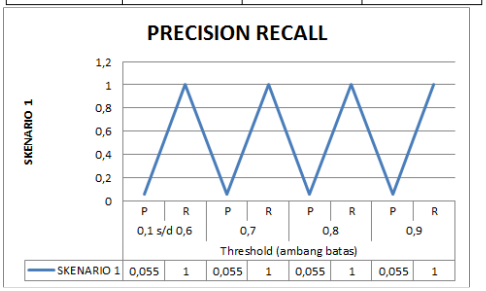
Pada pengujian skenario 1 diperoleh hasil skor dokumen yang merupakan nilai kemiripan antara *query* dan dokumen dalam *dataset*. Nilai kemiripan dari hasil perhitungan query berdasarkan classakan dilakukan perhitungan *precision-recall*berdasarkan threshold (ambang batas). Berikut hasil perhitungannya :

a. Untuk ambang batas (threshold) yang mempunyai nilai probabilitas 0,1 s/d 0,9

Precision	Recall
2/36 = 0,055	2/2 = 1

Tabel 4.1 : Tabulasi metrik *precision* dan *recall* dari skenario 1

	Threshold	Precision	Recall
Skenario 1	0,1	0,055	1
	0,2	0,055	1
	0,3	0,055	1
	0,4	0,055	1
	0,5	0,055	1
	0,6	0,055	1
	0,7	0,055	1
	0,8	0,055	1
	0,9	0,055	1



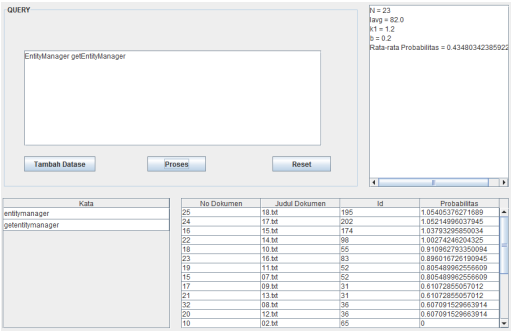
Gambar 4.1 : Grafik metrik *precision* dan *recall* dari skenario 1

Grafik di atas menunjukkan hasil *precision-recall* dari skenario pengujian 1, dari hasil percobaan tersebut threshold (ambang batas) yang mempunyai nilai 0,1 s/d 0,9 mempunyai nilai hasil precision recall yang sama yaitu P=0,055 dan R=1.

2. Skenario 2
   Memasukkan query berdasarkan method
   Dokumen pengujian yang benar adalah sebagai berikut :

No Dokumen	Judul Dokumen
17	09.txt
18	10.txt
19	11.txt
25	18.txt

Query 2:***EntityManager getEntityManager***Hasil : query berdasarkan method



Pada pengujian skenario 2 diperoleh hasil skor dokumen yang merupakan nilai kemiripan antara *query* dan dokumen dalam *dataset*. Nilai kemiripan dari hasil perhitungan query berdasarkan classakan dilakukan perhitungan *precision-recall*berdasarkan threshold (ambang batas). Berikut hasil perhitungannya :

a. Untuk ambang batas (*threshold*) dengan nilai probabilitas 0,1 s/d 0,6

Precision	Recall
4/36 = 0,11111111	4/4 = 1

b. Untuk ambang batas (*threshold*) dengan nilai probabilitas 0,7

Precision	Recall
3/36 = 0,08333333	3/4 = 0,75

c. Untuk ambang batas (*threshold*) dengan nilai probabilitas 0,8

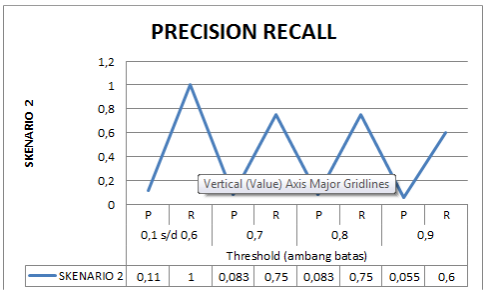
Precision	Recall
3/36 = 0,08333333	3/4 = 0,75

d. Untuk ambang batas (*threshold*) dengan nilai probabilitas 0,9

Precision	Recall
2/36 = 0,05555556	2/4 = 0,5

Tabel 4.2 : Tabulasi metrik *precision* dan *recall* dari skenario 2

	Threshold	Precision	Recall
Skenario 2	0,1	0,11	1
	0,2	0,11	1
	0,3	0,11	1
	0,4	0,11	1
	0,5	0,11	1
	0,6	0,11	1
	0,7	0,083	0,75
	0,8	0,083	0,75
	0,9	0,055	0,5



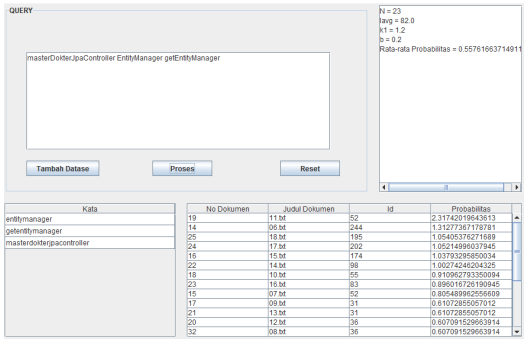
Gambar 4.2 : Grafik metrik *precision* dan *recall* dari skenario 2

Grafik di atas menunjukkan hasil *precision-recall* dari skenario pengujian 2, dari hasil percobaan tersebut dapat disimpulkan bahwa rata-rata *precision* tertinggi terdapat pada *threshold* yang mempunyai nilai probabilitas 0,1 s/d 0,6 dengan P=0,11dan R = 1.

3. Skenario 3
   Memasukkan query gabungan antara class dan method
   Dokumen pengujian yang benar adalah sebagai berikut :

No Dokumen	Judul Dokumen
17	09.txt
18	10.txt
19	11.txt
23	16.txt
25	18.txt

Query3:***masterDokterJpaController EntityManager getEntityManage***
Hasil query gabungan antara class dan method:



Pada pengujian skenario 3 diperoleh hasil skor dokumen yang merupakan nilai kemiripan antara *query* dan dokumen dalam *dataset*. Nilai kemiripan dari hasil perhitungan query berdasarkan classakan dilakukan perhitungan *precision-recall*berdasarkan threshold (ambang batas). Berikut hasil perhitungannya :

a. Untuk ambang batas (*threshold*) dengan nilai probabilitas 0,1 s/d 0,6

Precision	Recall
5/36 = 0,13	5/5 = 1

b. Untuk ambang batas (*threshold*) dengan nilai probabilitas 0,7

Precision	Recall
4/36 = 0,11	4/5 = 0,8

c. Untuk ambang batas (*threshold*) dengan nilai probabilitas 0,8

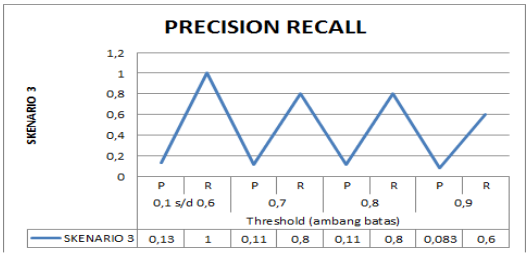
Precision	Recall
4/36 = 0,11	4/5 = 0,8

d. Untuk ambang batas (*threshold*) dengan nilai probabilitas 0,9

Precision	Recall
3/36 = 0,083	3/5 = 0,6

Tabel 4.3 : Tabulasi metrik *precision* dan *recall* dari skenario 3

	Threshold	Precision	Recall
Skenario 3	0,1	0,13	1
	0,2	0,13	1
	0,3	0,13	1
	0,4	0,13	1
	0,5	0,13	1
	0,6	0,13	1
	0,7	0,11	0,8
	0,8	0,11	0,8
	0,9	0,083	0,6



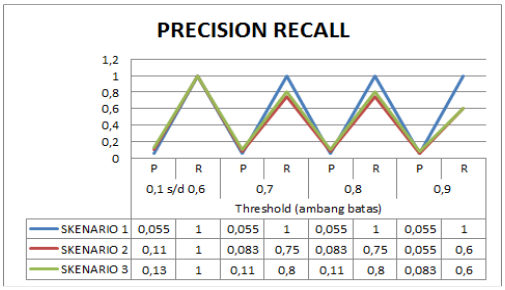
Gambar 4.3 : Grafik metrik *precision* dan *recall* dari skenario 3

Grafik di atas menunjukkan hasil *precision-recall* dari skenario pengujian 3, dari hasil percobaan tersebut dapat disimpulkan bahwa rata-rata *precision* tertinggi juga terdapat pada *threshold* yang mempunyai nilai probabilitas 0,1 s/d 0,6, dengan P=0,13dan R =1.

4.3. Analisa Pengujian

Tabel 4.4 : Tabulasi metrik *precision* dan *recall* dari *skenario 1, 2 dam 3*

Sken	Threshold (ambang batas)							
	0,1 s/d 0,6		0,7		0,8		0,9	
	P	R	P	R	P	R	P	R
1	0,055	1	0,055	1	0,055	1	0,055	1
2	0,11	1	0,083	0,75	0,083	0,75	0,055	0,6
3	0,13	1	0,11	0,8	0,11	0,8	0,083	0,6



Gambar 4.4 : Grafik metrik *precision* dan *recall* dari *skenario 1, 2 dan 3*

Grafik di atas menunjukkan bahwa hasil *precision-recall* dari analisa pengujian 1, 2 dan 3 dapat disimpulkan bahwa rata-rata *precision*dan *recall*tertinggi terdapat pada gabungan antara query class dan method yaitu pada skenario 3, dengan dengan ambang batas 0,1 – 0,6 yang mempunyai nilai P=0,13 dan R=1.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Dari uji coba dan analisa yang telah dijelaskan dalam bab sebelumnya, penulis mengambil beberapa kesimpulan sebagai berikut :

1. Sistem dapat membuat tools pengambilan kode sumber berdasarkan class dan method dengan praproses,
2. Skenario 3 merupakan pengujian dengan kinerja terbaik pada keseluruhan ambang batas dibandingkan dengan skenario 1 dan skenario 2. Hal ini menunjukkan bahwa query yang melibatkan *class* dan *method* memiliki hasil yang lebih baik,
3. *Threshold* 0,1 sampai dengan 0,9 pada skenario 3 memiliki hasil *precision* dan *recall* lebih baik daripada *threshold* yang lain.

5.2. Saran

Penulis ingin memberikan beberapa saran yang mungkin dapat membantu dalam pengembangan Tugas Akhir ini adalah sebagai berikut :

1. Sistem dikembangkan untuk mencari contoh kode sumberdengan berbagai macam kategori.
2. Untuk pengembangan aplikasi selanjutnya diharapkan dilakukan pengujianmenggunakan data yang lebih banyak sehingga validitas hasil lebih baik.

6. Pustaka

Alberto. M, 1997, *Compound Key Word Generation from Document Basis data Using A Hierarchical Clustering ART Model,* Intelligent Data analysis.

Ambara F. Horasi, 1997. *Rancangan Kamus Kata untuk Pemeriksaan Ejaan elektronis, Tugas Akhir fakultas Ilmu Komputer*, Universitas Indonesia.

Cleland-Huang, 2006. *Precision and Recall from Dokument Pencarian,* Intelligent Data analysis.

Gerard Salton, (1989), "*Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley, Reading, Mass.

Mandala, 2004. *Perangkingan dokumen untuk sistem pencarian.*