IMPLEMENTASI ALGORITMA GENETIK PADA QUESTION ANSWERING SYSTEM

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Herley Shaori Al-Ash
NIM: 115090600111008

PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PENGESAHAN

IMPLEMENTASI ALGORITMA GENETIK PADA QUESTION ANSWERING SYSTEM

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Herley Shaori Al-Ash
NIM: 115090600111008

Skripsi ini telah diuji dan dinyatakan lulus pada
8 Januari 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Drs. Achmad Ridok, M.Kom	Edy Santoso, S.Si, M.Kom
NIP: 19680825 199403 1 004	NIP: 197404014 200312 1 004

Mengetahui
Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, MT
NIP: 19670801 199203 1 001

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Januari 2016

Herley Shaori Al-Ash

NIM: 115090600111008

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

KATA PENGANTAR

Alhamdulillah Segala Puji Syukur kehadirat Allah SWT yang telah memberikan limpahan rahmat dan rahimNya sehingga penulis dapat menyelesaikan skripsi ini dengan penuh kebahagiaan. Shalawat serta salam semoga selalu tercurah kepada junjungan Nabi Besar Muhammad SAW.

Sebuah karya yang tak bisa terhitung nilainya merupakan suatu kebanggaan bagi penulis di sepanjang hidupnya, namun perlu diketahui bahwa keberhasilan tersebut juga didukung oleh pihak-pihak yang mengorbankan waktu dan tenaga materi untuk skripsi ini, sebagai bentuk rasa syukur dan berterima kasih berikut nama berbagai pihak yang telah berjasa :

1. Drs. Achmad Ridok M.Kom., selaku Dosen Pembimbing I yang memberikan banyak bimbingan, dan ilmu yang bermanfaat mulai dari awal hingga akhir penulisan serta kesabaran beliau disetiap waktu.
2. Edy Santoso, S.Si., M.Kom., selaku Dosen Pembimbing II yang tak kalah melelahkan juga untuk membimbing penulis memperoleh gelar Sarjana Komputer.
3. Drs. Marji, M.T., selaku Ketua Program Studi Ilmu Komputer beserta jajarannya yang telah memberikan kemudahan sistem birokrasi yang baik.
4. Kedua Orang Tua dan Adik yang telah mendukung penulis dengan segala usahanya, mulai dari doa, dan materi untuk keberhasilan penulis.
5. Segenap Bapak dan Ibu Dosen yang telah bersedia memberikan ilmunya serta saran dan kiritik kepada penulis.
6. Civitas Akademika FILKOM, Staf Akademik, dan Staf Laboratorium yang telah membantu setiap proses birokrasi yang ada.
7. Semua sahabat di Griya Brawijaya yang telah memberikan ilmu dan semangat dalam penyusunan skripsi ini.
8. Rekan-Rekan Ilmu Komputer baik kakak tingkat, teman seangkatan, maupun adik tingkat yang telah menjadi rekan seperjuangan dalam menuntut ilmu di Universitas Brawijaya.

Penulis menyadari bahwa masih ada kekurangan dan ketidaksempurnaan dalam penyusunan skripsi ini. Penulis dengan hati yang terbuka menerima saran, kritik, ide yang positif serta membangun guna menjadi semangat untuk penulis agar menjadi lebih baik dikemudian hari. Semoga skripsi ini dapat memberikan manfaat kepada sesama, dimana pun dan kapan pun.

Malang, 12 Januari 2016

Penulis

Email : herley.shaori@gmail.com

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRAK

Herley Shaori Al - Ash. 2015. Implementasi Algoritma Genetik pada *Question Answering System*. Skripsi Program Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Pembimbing: Drs. Achmad Ridok M.Kom. dan Edy Santoso S.Si, M.Kom.

Seiring dengan perkembangan teknologi informasi, jumlah informasi berita dalam bentuk *digital* semakin banyak. Namun, layanan untuk mendapatkan informasi yang lebih spesifik didalam berita belum diimplementasikan oleh sebagian besar penyedia berita. Berdasarkan keterangan tersebut, diusulkan sistem yang mampu memberikan jawaban yang lebih spesifik menggunakan *question answering system*.

Representasi vektor *term* setiap kalimat tanya ditentukan menggunakan algoritma genetik, dimana pada proses pembobotannya menggunakan *latent semantic indexing* sebagai fungsi *fitness* algoritma genetik.

Penelitian ini menggunakan 5 kelas sebagai tujuan, dan menggunakan 30 data. Dari pengujian yang dilakukan didapatkan akurasi sebesar 73.32%. Berdasarkan akurasi tersebut, disimpulkan bahwa algoritma genetik dapat diimplementasikan pada *question answering system*.

Kata Kunci : *Question Answering System,* Algoritma Genetik, *Latent Semantic Indexing*.

UNIVERSITAS BRAWIJAYA

repository.ub.ac.id

ABSTRACT

Herley Shaori Al - Ash. 2015. *Genetik Algorithm Implementation on Question Answering System*. Skripsi Program Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Pembimbing: Drs. Achmad Ridok M.Kom. dan Edy Santoso S.Si, M.Kom.

Within the rapid development of tehnology information, news information on digital form grow enormously. However, the service to get specific information from news has not widely implemented by news provider. Based on the example, it is urgent to develop a system that can give specific answer using question answering system.

Term vector representation on each question sentence determined by genetik algorithm, where in the weighting process latent semantic indexing is used as genetik algorithm fitness function.

This research uses 5 class as purposes, and 30 data. According to question answering system test, obtained question answering accuracy at 73.32%. Based on that accuracy, concluded that genetik algorithm can be implemented on question answering system.

***Key Words** : Question Answering System, Genetik Algorithm, Latent Semantic Indexing.*

UNIVERSITAS BRAWIJAYA

DAFTAR ISI

PENGESAHAN ii
PERNYATAAN ORISINALITAS iii
KATA PENGANTAR iv
ABSTRAK v
ABSTRACT vi
DAFTAR ISI vii
DAFTAR TABEL x
DAFTAR GAMBAR xii
DAFTAR KODE SUMBER xv
DAFTAR LAMPIRAN xvi
BAB 1 PENDAHULUAN 1
1.1 Latar belakang 1
1.2 Rumusan masalah 2
1.3 Tujuan 2
1.4 Manfaat 2
1.5 Sistematika pembahasan 2
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI 4
2.1 Kajian pustaka 4
2.2 Question answering system 6
2.3 Dekomposisi nilai singular 7
2.4 Latent semantic indexing 8
2.5 Algoritma genetik 9
2.5.1 Representasi kromosom 11
2.5.2 Reproduksi 13
2.5.3 Fungsi fitness 14
2.5.4 Seleksi 14
BAB 3 METODOLOGI PENELITIAN DAN PERANCANGAN 15
3.1 Tahapan penelitian 15
3.2 Studi literatur 15
3.3 Perancangan gambaran umum sistem 15
3.4 Perancangan corpus 16

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

3.5 Perancangan dataset 17
3.6 Perancangan document preprocessing 17
3.6.1 Perancangan parsing 19
3.6.2 Perancangan stopword removal 20
3.6.3 Perancangan stemming 22
3.6.4 Perancangan term indexing 23
3.7 Perancangan klasifikasi dokumen 24
3.8 Perancangan klasifikasi kromosom algoritma genetik 25
3.9 Perancangan parameter algoritma genetik 30
3.10 Perancangan evolusi algoritma genetik 31
3.10.1 Perancangan inisialisasi populasi 32
3.10.2 Perancangan kondisi berhenti 33
3.10.3 Perancangan reproduksi 33
3.10.4 Perancangan evaluasi 37
3.10.5 Perancangan seleksi 53
3.11 Perancangan document pool 57
3.12 Perancangan sistem basis data 59
3.13 Perancangan pengujian algoritma 60
3.14 Perancangan pengujian question answering system 61
3.15 Perancangan antarmuka 62
3.15.1 Antarmuka sistem basis data 62
3.15.2 Antarmuka algoritma genetik 67
3.15.3 Antarmuka pengujian 70
3.15.4 Antarmuka pengajuan pertanyaan 72
3.15.5 Antarmuka perolehan jawaban 73
3.16 Manualisasi 73
BAB 4 IMPLEMENTASI 82
4.1 Struktur implementasi algoritma genetik pada question answering system 82
4.1.1 Modul question answering system 83
4.1.2 Modul database 83
4.1.3 Modul weighting 84
4.1.4 Modul processor 85

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.1.5 Modul explorer 86
4.1.6 Modul main-qa 88
4.1.7 Modul installer 89
4.2 Implementasi algoritma 89
4.2.1 Term Frequency-Inverted Document Frequency 90
4.2.2 Algoritma genetik 95
4.3 Implementasi pengujian 98
4.3.1 Pengujian algoritma genetik 98
4.3.2 Pengujian question answering system 102
4.4 Spesifikasi perangkat lunak 104
4.5 Spesifikasi perangkat keras 105
BAB 5 PENGUJIAN DAN ANALISIS 106
5.1 Pengujian parameter algoritma genetik 106
5.1.1 Penegujian jumlah iterasi 106
5.1.2 Pengujian jumlah populasi 107
5.1.3 Pengujian crossover rate dan mutation rate 108
5.1.4 Pengujian dimension rate 108
5.1.5 Pengujian informative term threshold 109
5.2 Pengujian algoritma genetik 111
5.3 Pengujian question answering system 112
BAB 6 PENUTUP 116
6.1 Kesimpulan 116
6.2 Saran 116
DAFTAR PUSTAKA 117
LAMPIRAN A DATA 119
LAMPIRAN B HASIL PENGUJIAN ALGORITMA GENETIK 150
LAMPIRAN C TERM 154

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR TABEL

Tabel 2.1 Istilah dalam algoritma genetik 10
Tabel 2.2 Istilah dalam algoritma genetik (lanjutan) 11
Tabel 3.1 Parameter algoritma genetik 30
Tabel 3.2 Perancangan pengujian algoritma genetik 61
Tabel 3.3 Perhitungan precision dan recall 61
Tabel 3.4 Perancangan pengujian question answering system 61
Tabel 3.5 Manualisasi perancangan dataset kelas siapa 74
Tabel 3.6 Daftar term 74
Tabel 3.7 Daftar term (lanjutan) 75
Tabel 3.8 Term pada kromosom 75
Tabel 3.9 Koleksi term pada kelas siapa 76
Tabel 3.10 Perhitungan term frequency inverted document frequency pada dataset 76
Tabel 3.11 Term frequency inverted document frequency pada query 77
Tabel 3.12 Perhitungan panjang vektor 78
Tabel 3.13 Vektor dokumen ternormalisasi 78
Tabel 3.14 Vektor dokumen ternormalisasi (lanjutan) 79
Tabel 3.15 Vektor kueri ternormalisasi 79
Tabel 3.16 Manualisasi parameter algoritma genetik 80
Tabel 3.17 Manualisasi kromosom proses evolusi selanjutnya 81
Tabel 5.1 Parameter algoritma genetik untuk pengujian iterasi 106
Tabel 5.2 Hasil pengujian jumlah iterasi 107
Tabel 5.3 Parameter algoritma genetik untuk jumlah populasi 107
Tabel 5.4 Parameter algoritma genetik untuk jumlah populasi (lanjutan) 108
Tabel 5.5 Hasil pengujian jumlah populasi 108
Tabel 5.6 Hasil pengujian crossover rate dan mutation rate 108
Tabel 5.7 Hasil pengujian informative term threshold 109
Tabel 5.8 Parameter algoritma genetik untuk informative term threshold 110
Tabel 5.9 Parameter algoritma genetik untuk pengujian 111
Tabel 5.10 Hasil pengujian algoritma genetik 112
Tabel 5.11 Jawaban sistem berdasarkan pertanyaan pengguna 113

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.12 Jawaban sistem berdasarkan pertanyaan pengguna (lanjutan) 114
Tabel 5.13 Jawaban sistem berdasarkan pertanyaan pengguna (lanjutan) 115

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR GAMBAR

Gambar 2.1 Singular value decomposition 8
Gambar 2.2 Cara kerja algoritma genetik 10
Gambar 2.3 Kromosom dalam sebuah populasi 11
Gambar 2.4 Struktur kromosom 11
Gambar 2.5 Diagram alir algoritma genetik 12
Gambar 3.1 Perancangan question answering system 16
Gambar 3.2 Perancangan document corpus 17
Gambar 3.3 Perancangan document preprocessing 18
Gambar 3.4 Perancangan interface document preprocessing 18
Gambar 3.5 Perancangan parsing 19
Gambar 3.6 Perancangan stopword removal 21
Gambar 3.7 Perancangan edit sentence 22
Gambar 3.8 Perancangan hashmap 23
Gambar 3.9 Perancangan interface menggunakan algoritma genetik 24
Gambar 3.10 Perancangan klasifikasi dokumen 25
Gambar 3.11 Perancangan masalah dan solusi menggunakan algoritma genetik 26
Gambar 3.12 Perancangan indexed hashmap 26
Gambar 3.13 Perancangan struktur kromosom 27
Gambar 3.14 Perancangan struktur data kromosom 27
Gambar 3.15 Perancangan representasi struktur data kromosom 28
Gambar 3.16 Perancangan representasi kromosom 29
Gambar 3.17 Perancangan interface representasi kromosom 30
Gambar 3.18 Perancangan evolusi algoritma genetik 32
Gambar 3.19 Perancangan inisialisasi populasi 33
Gambar 3.20 Perancangan interface operator genetik 34
Gambar 3.21 Perancangan reproduksi 34
Gambar 3.22 Perancangan hierarchical crossover 35
Gambar 3.23 Perancangan mutasi 37
Gambar 3.24 Perancangan evaluasi 39
Gambar 3.25 Perancangan weighting 40
Gambar 3.26 Perancangan hashmap value 41

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 3.27 Perancangan struktur data term frequency 41
Gambar 3.28 Perancangan perhitungan *term frequency* dan *document frequency* 42
Gambar 3.29 Perancangan struktur data *matrix* 44
Gambar 3.30 Perancangan perhitungan TF-IDF 46
Gambar 3.31 Perancangan interface term frequency 47
Gambar 3.32 Perancangan klas TF 47
Gambar 3.33 Perancangan latent semantic indexing 48
Gambar 3.34 Perancangan perhitungan latent semantic indexing 50
Gambar 3.35 Perancangan perhitungan cosine similarity 51
Gambar 3.36 Perancangan perhitungan normalisasi vektor 52
Gambar 3.37 Perancangan seleksi 54
Gambar 3.38 Perancangan klas probability 55
Gambar 3.39 Perancangan seleksi 56
Gambar 3.40 Perancangan klas abstrak natural selector 56
Gambar 3.41 Perancangan question answer 58
Gambar 3.42 Entity relationship diagram 59
Gambar 3.43 Antarmuka sistem basis data 63
Gambar 3.44 Antarmuka corpus 63
Gambar 3.45 Antarmuka corpus sentence 64
Gambar 3.46 Dataset assignment 64
Gambar 3.47 Kelas pada dataset 65
Gambar 3.48 Add dataset kelas SIAPA 65
Gambar 3.49 Show dataset kelas SIAPA 66
Gambar 3.50 Stopword removal 66
Gambar 3.51 Class term 67
Gambar 3.52 Class term kelas siapa 67
Gambar 3.53 Parameter algoritma genetik 68
Gambar 3.54 Keterangan kromosom algoritma genetik 68
Gambar 3.55 Grafik kromosom terbaik pada setiap evolusi 69
Gambar 3.56 Hasil klasifikasi algoritma genetik kelas siapa 70
Gambar 3.57 Antarmuka pengujian 71
Gambar 3.58 Pengujian algoritma genetik 71

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Gambar 3.59 Pengujian question answering system 72
Gambar 3.60 Hasil pengujian question answering system 72
Gambar 3.61 Pengajuan pertanyaan 73
Gambar 4.1 Struktur modul implementasi 82
Gambar 4.2 Struktur modul question answering system 83
Gambar 4.3 Struktur modul database 83
Gambar 4.4 Struktur modul weighting 84
Gambar 4.5 Struktur modul processor 85
Gambar 4.6 Struktur modul explorer 87
Gambar 4.7 Struktur modul main-qa 88
Gambar 4.8 Struktur modul installer 89
Gambar 5.1 Akurasi implementasi algoritma genetik pada question answering system 115

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR KODE SUMBER

Kode Sumber 4.1 Perhitungan term frequency inverted document frequency... 92

Kode Sumber 4.2 Representasi matrix term frequency inverted document frequency 94

Kode Sumber 4.3 Implementasi algoritma genetik 97

Kode Sumber 4.4 Implementasi pengujian algoritma genetik 101

Kode Sumber 4.5 Question answering system 104

DAFTAR LAMPIRAN

LAMPIRAN A DATA 119
LAMPIRAN B HASIL PENGUJIAN ALGORITMA GENETIK 150
LAMPIRAN C TERM 154

BAB 1 PENDAHULUAN

1.1 Latar belakang

Seiring dengan perkembangan teknologi informasi, jumlah informasi berita dalam bentuk *digital* semakin banyak. Namun, layanan untuk mendapatkan informasi yang lebih spesifik didalam berita belum diimplementasikan oleh sebagian besar penyedia berita. Misalkan informasi spesifik yang dibutuhkan adalah peristiwa apa yang sedang terjadi, siapa yang terlibat didalam peristiwa tersebut, dan dimana peristiwa tersebut terjadi. Berdasarkan keterangan tersebut, diusulkan sistem dengan metode tertentu yang mampu memberikan jawaban yang lebih spesifik sehingga didapatkan informasi spesifik yang dibutuhkan.

Terdapat beberapa metode yang dapat digunakan dalam pengambilan informasi. Setiap metode memiliki karakteristik sehingga mungkin saja satu metode yang bagus diterapkan pada satu kasus, belum tentu dapat diterapkan pada kasus yang lain (Grossman & Frieder, 2004).

Salah satu metode yang digunakan dalam pengambilan informasi adalah metode yang menghasilkan beberapa dokumen relevan dengan kebutuhan pengguna, hal ini dibuktikan dengan *search engine* ternama seperti *Google, Yahoo, Bing, Galaxy* yang menggunakan metode ini. Dengan metode tersebut, pengguna diharapkan dapat mencari jawaban didalam dokumen tersebut, jika pengguna tidak menemukan jawaban yang diharapkan pada satu dokumen, maka pengguna harus mencari jawaban tersebut pada dokumen yang lain. Jika pengguna mengharapkan jawaban yang presisi misalnya nama orang, nama tempat, atau tanggal suatu kejadian, maka pengguna harus tetap mencari lebih dalam ke seluruh dokumen.

Metode yang lain memungkinkan pengguna untuk lebih dekat kepada jawaban yang lebih presisi. *Ask.com* dan *Answers.com* adalah contoh dari metode ini. Pengguna yang mengharapkan jawaban presisi tidak perlu menelusuri seluruh dokumen yang dikembalikan oleh mesin pencari, namun pengguna langsung mendapatkan jawaban yang diharapkan. Metode ini dikenal dengan *Question Answering System (QAS)*.

Question Answering System adalah sebuah sistem yang secara otomatis menjawab pertanyaan berdasarkan bahasa alami. Tujuan utama dari QA adalah mengembalikan hasil ringkasan dokumen yang dianggap paling relevan berdasarkan pertanyaan pengguna. Ringkasan dokumen yang dikembalikan dapat berupa kumpulan kalimat, sebuah kalimat, atau beberapa kata yang saling terkait (Barskar, et al., 2011).

Pengembangan *question answering system* membutuhkan sistem penilaian yang bertujuan untuk menilai *term* terhadap dokumen. *Term* yang digunakan bertindak sebagai kata kunci terhadap dokumen, sehingga dilakukan klasifikasi *term* untuk mendapatkan kata kunci tersebut. Metode klasifikasi *term* yang

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

digunakan adalah algoritma genetik karena algoritma genetik mampu menghasilkan solusi optimum (Mahmudy, 2013). Solusi optimum adalah *term* yang bertindak sebagai kata kunci dalam pencarian jawaban menggunakan *question answering system*.Dalam implementasinya, algoritma genetik dikombinasikan dengan *latent semantic indexing* (*lsi*). *Latent semantic indexing* adalah salah satu metode index pada *information retrieval* yang secara praktis dapat memahami makna dari dokumen. Memaknai dokumen berdasarkan pertanyaan pengguna memainkan peran yang penting dalam *Question Answering System* (Grossman & Frieder, 2004).

Berdasarkan uraian tersebut diusulkan algoritma genetik pada implementasi *question answering system*. Implementasi tersebut diharap mampu untuk mengembangkan *question answering system* dan memberikan jawaban yang lebih tepat kepada pengguna berdasarkan pertanyaan pengguna menggunakan bahasa alami.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, maka dapat ditarik rumusan masalah sebagai berikut :

1. Merancang dan mengimplementasikan algoritma genetik pada *question answering system.*
2. Menghitung akurasi implementasi algoritma genetik pada *question asnwering system.*

1.3 Tujuan

Tujuan dari penelitian ini adalah :

1. Mengimplementasikan algoritma genetik pada *question answering system*.
2. Mendapatkan akurasi dari implementasi algoritma genetik pada *question answering system*.

1.4 Manfaat

Manfaat yang diperoleh dari penelitian adalah memberikan jawaban yang spesifik berdasarkan pertanyaan pengguna dengan seoptimal mungkin tidak mengurangi makna dari pertanyaan pengguna pada *question answering system*.

1.5 Sistematika pembahasan

Sistematika pembahasan laporan skripsi disesuaikan dengan tata cara penulisan skripsi program studi Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 1 **Pendahuluan**

Terdiri atas latar belakang, rumusan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB 2 **Kajian Pustaka dan Dasar Teori**

Terdiri atas kajian pustaka *question answering system*, pengertian *question answering system*, pembobotan *latent semantic indexing*, pengukuran kesamaan menggunakan *cosine similarity*. Pengertian, keunggulan dan representasi kromosom pada algoritma genetik yang melandasi penyusunan dan perancangan dalam pembuatan skripsi.

BAB 3 **Metode Penelitian dan Perancangan**

Membahas metode dan langkah yang dilakukan pada penelitian. Terdiri atas studi literatur, analisa kebutuhan sistem, perancangan sistem, dan implementasi sistem dari penelitian *'Implementasi Algoritma Genetik pada Question Answering System'*.

BAB 4 **Implementasi**

Membahas implementasi dari penelitian *"Implementasi Algoritma Genetik pada Question Answering System"*.

BAB 5 **Pengujian dan Analisis**

Membahas tingkat akurasi dan analisa hasil dari penelitian *"Implementasi Algoritma Genetik pada Question Answering System"*.

BAB 6 **Penutup**

Berisi kesimpulan yang diperoleh dari dari pengujian dan analisis, serta saran untuk penelitian lebih lanjut.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI

Bab II menjelaskan kajian pustaka dan dasar teori. Kajian pustaka yang dijelaskan adalah *question answering system (qas)*. Penjelasan tentang *qas* meliputi pengertian *qas,* dan Gambaran umum *qas*. Dasar teori yang dijelaskan meliputi *singular value decomposition (svd), latent semantic indexing, cosine similarity, vector space model,* serta algoritma genetik.

2.1 Kajian pustaka

Pada penelitian ini, terdapat beberapa pustaka yang dibandingkan. Pustaka tersebut merupakan penelitian yang sudah dilakukan dan menjadi landasan bagi penelitian ini. Setiap pustaka menjelaskan judul, tujuan penelitian, obyek, metode, proses, dan hasil penelitian.

Penelitian yang pertama adalah Implementasi Question Answering System dengan Metode *Rule-Based* pada terjemahan Al-Qur'an Surat Al Baqarah (Aggraeny, 2007). Objek dari penelitian ini adalah surat Al-Baqarah pada Al-Qur’an. Tujuan dari penelitian ini adalah menentukan *rule* berdasarkan pola dokumen, mengimplementasikan sistem temu kembali informasi yang menggunakan suatu kueri pertanyaan dengan metode *rule-based* pada terjemahan Al-Qur'an surat Al-Baqarah, dan menganalisis tingkat akurasi sistem dalam menemukembalikan jawaban yang relevan berdasarkan kueri yang diberikan.

Langkah dalam penelitian tersebut dimulai dengan mendapatkan koleksi dokumen. Dokumen yang digunakan dalam penelitian tersebut adalah terjemahan surat Al-Baqarah yang merupakan surat terpanjang pada Al-Qur’an berdasarkan jumlah ayat (286 ayat). Satu terjemahan ayat direpresentasikan sebagai satu dokumen.

Setelah setiap ayat menjadi sebuah dokumen, dilakukan *parsing*, *stopwords removal*, dan *stemming* sehingga menghasilkan *term* yang disimpan pada Tabel penyimpanan kata.

Pengguna kemudian memasukkan kueri berupa kalimat pertanyaan. Kalimat pertanyaan pengguna diproses dengan melakukan *parsing*, *stopwords removal*, dan *stemming* sehingga menghasilkan *term*.

Term dari dokumen pada Tabel penyimpanan kata dan term dari pengguna yang sudah diproses dibandingkan dengan proses *WordMatch* kemudian masuk kedalam *rule* sesuai dengan tipe kueri yang diberikan. Setiap kalimat pada dokumen memperoleh nilai berdasar *WordMatch* dan *rule* yang sesuai, kalimat yang dikembalikan sebagai jawaban adalah kalimat yang memiliki nilai yang paling tinggi.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Hasil penelitian yang didapatkan adalah *question answering system* dengan metode *rule-based* dan *WordMatch* sehingga dihasilkan *rule* untuk setiap tipe pertanyaan. Berdasarkan evaluasi sistem, tipe pertanyaan "siapa" mempunyai kinerja paling tinggi, sedangkan tipe pertanyaan "mana" memiliki kinerja paling rendah. Akurasi rata-rata *rule* yang diberikan oleh penulis (Aggraeny, 2007) adalah 86.69%, sedangkan akurasi rata-rata *rule* yang diberikan pengguna umum adalah 53.13%.

Perbedaan penelitian tersebut dengan penelitian pada skripsi ini adalah algoritma yang digunakan oleh (Aggraeny, 2007) adalah *WordMatch* dan *rule-based*, sedangkan algoritma yang digunakan oleh penulis adalah algoritma genetik.

Penelitian yang kedua adalah *An Approach for Extracting Exact Answers to Question Answering System for English Sentences* (Barskar, et al., 2011). Tujuan dari penelitian tersebut adalah membangun *question answering system* berdasarkan *keyword search*. Untuk menggunakan sistem tersebut, pengguna memasukkan kueri berbahasa Inggris, kemudian sistem mengembalikan jawaban berdasarkan kata-kata pada kamus bahasa Inggris. Objek pada penelitian ini adalah bahasa Inggris. Metode yang digunakan adalah *first order logic* (Barskar, et al., 2011).

Langkah pada penelitian tersebut dimulai dengan memasukkan kata dalam bahasa Inggris kedalam sistem. Kata dalam kamus bahasa Inggris tersebut dianggap sebagai kata yang sudah baku, sehingga tidak dilakukan proses *stemming*, namun proses *parsing* dan *stopwords removal* tetap dilakukan. Pengguna memasukkan kueri dalam bahasa Inggris yang berupa pertanyaan.

Kueri pengguna dianggap sebagai satu kalimat yang dikonversi menjadi *predicate logic* dan *clause form* menggunakan *first order logic*. Sistem membuat aturan menggunakan algoritma *resolution* dan *unification*, algoritma tersebut juga digunakan dalam proses pengukuran kesamaan antara kueri pengguna terhadap seluruh dokumen. Setiap dokumen yang telah dikombinasikan pada *predicate logic* dan *clause form* mendapatkan nilai. Nilai yang paling tinggi dikembalikan sebagai jawaban atas pertanyaan pengguna.

Hasil dari penelitian adalah komputer dapat dengan mudah melakukan *natural language processing* dengan menggunakan *predicate logic*, waktu yang dibutuhkan untuk melakukan proses komputasi juga relatif lebih cepat.

Perbedaan penelitian tersebut dengan penelitian penulis adalah algoritma yang digunakan oleh (Barskar, et al., 2011) adalah *first order logic* untuk membuat *rule*, sedangkan algoritma yang digunakan oleh penulis adalah algoritma genetik.

Genetik Algorithms for data-driven Web Question Answering (Figueroa & Neumann, 2008). Tujuan dari penelitian tersebut adalah berusaha mendapatkan yang benar dan tepat berdasarkan algoritma genetik. Objek yang diteliti adalah *optimized web search engine* dan metode yang digunakan adalah algoritma genetik.

Jawaban di ekstrak secara langsung dari *N-Best Snippets* yang telah diidentifikasi sebelumnya pada *web search engine* menggunakan pertanyaan bahasa alami. Setiap jawaban yang dihasilkan *web search engine* tersebut diolah lebih lanjut menggunakan algoritma genetik. Algoritma genetik membangkitkan *term* secara acak dari jawaban yang dihasilkan oleh *web search engine* tersebut. Satu *term* digabungkan dengan *term* yang lain sehingga membentuk kalimat. Kalimat tersebut dicari konteksnya, kemudian dibandingkan dengan konteks seluruh jawaban yang dibangkitkan oleh *web search engine* sebelumnya dan digunakan sebagai fungsi *fitness* bersama dengan pertanyaan *pengguna*.

Hasil dari penelitian tersebut adalah membangun *question answering system* dengan algoritma genetik hasilnya sangat menjanjikan. Sistem tersebut (Figueroa & Neumann, 2008) telah diuji dengan tujuh data uji.

Perbedaan penelitian tersebut dengan penelitian penulis adalah dalam mencari konteks atau maksud dari sebuah kalimat, penelitian penulis menggunakan *latent semantic indexing*, sedangkan (Figueroa & Neumann, 2008) menggunakan *natural language processing*.

Penelitian yang ketiga adalah *Question Classification in English-Chinese Cross-Language Question Answering : An Integrated Genetik Algorithm and Machine Learning Approach* (Day, et al., 2007). Tujuan dari penelitian tersebut adalah membangun *question answering system* secara lintas bahasa, yaitu bahasa Inggris dan bahasa China. Objek dari penelitian tersebut adalah *cross language question answering system*. Algoritma yang digunakan adalah algoritma genetik, *support vector machine,* dan *conditional random field*.

Corpus data dalam bahasa Inggris dan bahasa China dimasukkan pada sistem. Pengguna kemudian dapat mengajukan pertanyaan baik dalam bahasa Inggris maupun dalam bahasa China.

Pertanyaan pengguna diproses lebih lanjut. Fitur dari pertanyaan pengguna tersebut didapatkan dengan menggunakan algoritma genetik. Fitur yang didapatkan digunakan bersama dengan *conditional random field* untuk membentuk sebuah *question informer prediction* yang bertugas untuk mendapatkan kemungkinan klasifikasi pertanyaan. Klasifikasi pertanyaan menggunakan *support vector machine,* ketika pertanyaan sudah mendapatkan kelas, jawaban yang sesuai dengan kelas tersebut dicari pada *corpus data*.

Hasil dari penelitian tersebut adalah bahwa metode yang diusulkan oleh (Day, et al., 2007) pada data uji NTCIR-6 CLQA mendemonstrasikan efisiensi dan akurasi yang baik pada *question answering system* lintasi bahasa Inggris dan bahasa China.

2.2 Question answering system

Beberapa mesin pencari ternama seperti *Google, Yahoo, Bing* memiliki kemampuan untuk memberikan jawaban atas kueri pengguna, jawaban tersebut diberikan dalam bentuk *url (uniform resource location)* yang merujuk kepada beberapa dokumen yang menurut mesin pencari tersebut paling relevan terhadap kueri pengguna. Hal tersebut merupakan konsep dari *information retrieval* yaitu

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

menemukembalikan satu atau beberapa dokumen utuh yang paling relevan berdasarkan *pengguna query* (Biswas, et al., 2014).

Meskipun mesin pencari tersebut mengembalikan satu atau beberapa dokumen kepada pengguna, belum tentu dokumen yang dikembalikan kepada pengguna adalah jawaban yang tepat berdasarkan kueri pengguna, sehingga ide dasar dari *QAS* adalah menemukembalikan informasi yang relevan dari koleksi dokumen tersebut (Biswas, et al., 2014).

Question Answering System adalah sistem yang mengijinkan *pengguna* untuk menyatakan kebutuhan informasinya dalam bentuk pertanyaan dengan bahasa alami, dan mengembalikan kutipan berupa teks singkat atau frase sebagai jawaban (Gunawan & Lovina, 2006).

Tujuan dari *question answering system* adalah untuk menjawab pertanyaan pengguna seringkas mungkin daripada mengembalikan koleksi dokumen yang dilakukan kebanyakan oleh *retrieval system* yang lain (Brill, et al., 2002).

2.3 Dekomposisi nilai singular

Singular Value Decomposition atau dekomposisi nilai singular adalah salah satu teknik dekomposisi $matrix$ yang berkaitan dengan nilai singular (*singular value*) suatu *matrix* yang merupakan karakteristik *matrix* tersebut (Ariyanti, 2010).

Singular Value Decomposition Matrix didefinisikan dengan Persamaan :

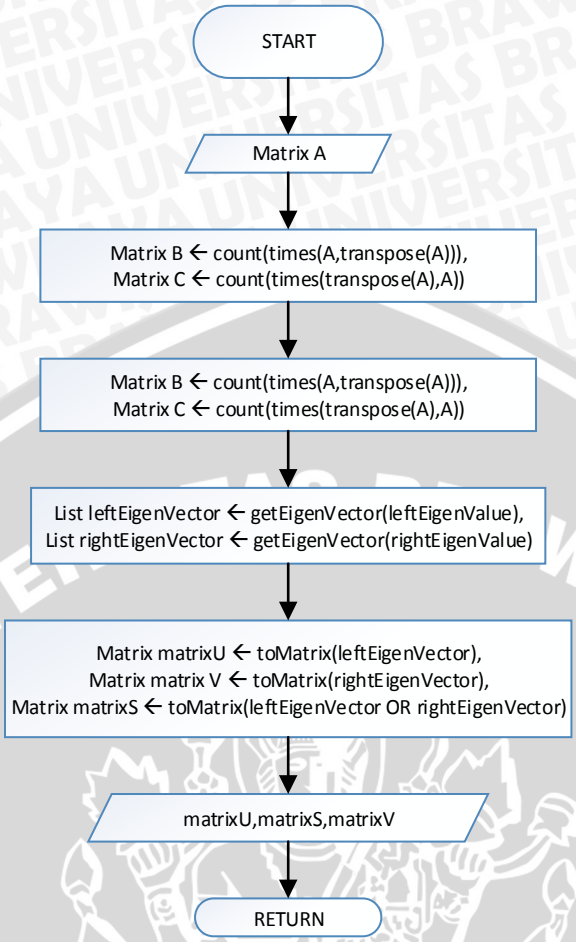
$$A = USV^T \tag{2.1}$$

Dimana A adalah *matrix riil* dengan dimensi (mxn). U adalah $matrix\ orthogonal$ dengan dimensi $(m\ x\ m)$. V adalah $matrix$ orthogonal dengan dimensi $(n\ x\ n)$. S merupakan *matrix* diagonal dengan dimensi $(m\ x\ n)$bernilai riil tidak negatif yang disebut dengan nilai singular (Ariyanti, 2010).

Matrix ortogonal adalah *matrix* yang memenuhi Persamaan :

$$AA^T = I \tag{2.2}$$

Dimana A^T adalah transpose dari $matrix\ A$, I adalah *matrix* identitas, dan A adalah *matrix* bujur sangkar (Todd, 2015). *Matrix* diagonal adalah *matrix* bujur sangkar dimana semua elemen dalam *matrix* tersebut selain diagonal utama bernilai 0 (Weisstein, 2015). Langkah untuk menghitung *singular value decomposition* $matrix$ disajikan pada *flowchart* Gambar 2.1



Gambar 2.1 Singular value decomposition

Berdasarkan $matrix\ A$, dapat dihitung $matrix\ AA^T$ dan $matrix\ A^TA$. Eigen value dan eigen vector dari kedua $matrix$ tersebut dihitung.

Eigen vector dari $matrix\ A^TA$ membentuk kolom $matrix$ V, Eigen vector dari $matrix\ AA^T$ membentuk kolom $matrix\ U$ (Ariyanti, 2010). Nilai singular dari $matrix\ U$ adalah akar pangkat dua dari nilai eigen $matrix\ AA^T$atau A^TA, nilai singular adalah elemen diagonal dari $matrix\ S$ yang diurutkan secara descending (Ariyanti, 2010).

2.4 Latent semantic indexing

Latent semantic indexing merupakan metode yang dapat secara praktis digunakan untuk memahami makna (semantik) sebuah kalimat. *Latent semantic indexing* mendeteksi kebersamaan dalam penggunaan kalimat kemudian menganggap bahwa jika kalimat yang beberapa kali terdapat kebersamaan, maka kalimat tersebut dalam satu konsep (Koepke, 2007). Misalnya terdapat *term* pada sebuah artikel yang menyebutkan *"probability"*, maka *term* yang diharapkan muncul berikutnya adalah *"distribution"* atau *"convergence"* karena kedua *term* ini dianggap dalam satu konsep dengan *"probability"* (Koepke, 2007).

LSI digunakan untuk mengoptimasi *inverted index* yang menggunakan *term frequency,* dan *inverted document frequency*, karena dalam pendekatan *inverted index* tersebut makna dari suatu kalimat terhadap kalimat yang lain tidak diperhitungkan. *Inverted index* hanya fokus pada jumlah *term* pada satu dokumen terhadap jumlah *term* yang sama pada dokumen yang lain. Hal tersebut menyebabkan *term* pada *inverted index matrix* bersifat independen, artinya *term* yang memiliki kesamaan makna atau konteks tidak diperhitungkan lebih lanjut (Grossman & Frieder, 2004), sehingga pada saat dilakukan *similarity* antara *query* dan *document, similarity* yang dicapai bukan *similarity* antara dua *term* yang memiliki kesamaan makna atau *context,* namun *similarity* antara dua *term* yang memiliki nilai yang paling besar menurut perhitungan *term frequency – inverted document index* saja (Grossman & Frieder, 2004). *Latent semantic indexing* dapat juga disebut *latent semantic analysis* (Turian, 2012).

Langkah melakukan *latent semantic indexing* dimulai dengan merubah koleksi dokumen dan kueri pengguna kedalam *term frequency-document frequency matrix*.

Matrix koleksi dokumen diolah lebih lanjut menggunakan *singular value decomposition* sehingga didapatkan $matrix\ U$, S, dan V. *Matrix* V adalah $matrix$ yang berisi vektor dokumen. Vektor kueri didapatkan dengan Persamaan 2.3.

Pengurangan dimensi merupakan salah satu fitur dalam *latent semantic indexing.* Pengurangan dimensi tersebut bertujuan untuk mengurangi kompleksitas komputasi. Besar pengurangan dimensi tersebut dilakukan dengan beberapa kali percobaan dengan sebuah konstanta k. Berdasarkan beberapa penelitian yang telah dilakukan, nilai k optimum sama dengan didalam jangkauan 100, namun konstanta ini bukan sebuah tetapan yang baku (Grossman & Frieder, 2004).

$$q = q^T U_K S_K^{-1} \tag{2.3}$$

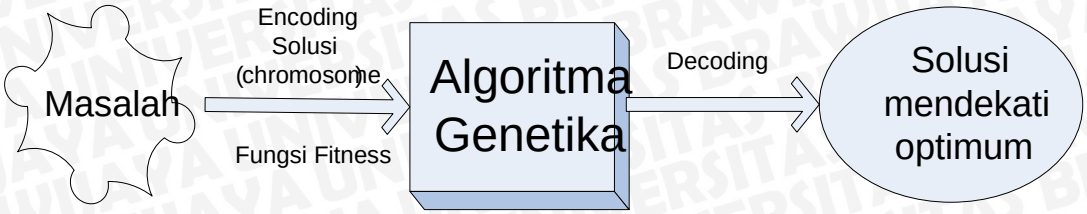
Untuk semua vektor dokumen dihitung *similarity* terhadap vektor kueri. Setiap vektor dokumen akan mendapatkan nilai berdasar kemiripan konteks. Vektor dokumen yang sudah mendapatkan nilai, diurutkan secara *descending*. Vektor dokumen teratas dianggap dokumen yang paling relevan (Grossman & Frieder, 2004).

2.5 Algoritma genetik

Algoritma genetik adalah salah satu model dalam *machine learning* yang perilakunya meniru proses evolusi di alam. Kemiripan perilaku tersebut dilakukan dengan membuat populasi. Sebuah populasi terdiri dari sekumpulan individu atau kromosom, sebuah kromosom terdiri dari sekumpulan gen. Pada level yang lebih detail, sebuah gen berisi sekumpulan alel (Anon., 1995).

Menurut (Mahmudy, 2013) langkah penyelesaian algoritma genetik secara umum direpresentasikan oleh Gambar 2.2.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 2.2 Cara kerja algoritma genetik

Sumber : (Mahmudy, 2013)

Pada Gambar 2.2, masalah direpresentasikan dalam kromosom. Solusi atas masalah tersebut direpresentasikan sebagai satu individu, dengan demikian satu individu merepresentasikan satu solusi, sehingga sebuah populasi yang berisi sekumpulan individu berarti berisi sekumpulan solusi. Representasi solusi masalah menjadi satu individu disebut *encoding* (Mahmudy, 2013).

Untuk mengukur seberapa baik solusi tersebut dalam menyelesaikan masalah, digunakan fungsi *fitness*. Semakin tinggi fungsi *fitness* tersebut berarti solusi yang dihasilkan oleh individu tersebut semakin baik.

Dengan menggunakan representasi kromosom dan fungsi *fitness,* algoritma genetik mencari solusi terbaik menirukan proses evolusi. Proses evolusi yang dilakukan adalah proses reproduksi dan proses seleksi. Setelah melalui proses tersebut, kromosom mengalami proses *decode* untuk diuraikan menjadi solusi yang diharapkan mendekati optimum (Mahmudy, 2013).

Menurut (Skinner, 2001) Dalam algoritma genetik, istilah yang digunakan dijelaskan pada Tabel 2.1 dan Tabel 2.2

Tabel 2.1 Istilah dalam algoritma genetik

No	Istilah	Penjelasan
1	Individu	Solusi yang dibangun pada algoritma genetik. Satu individu merepresentasikan satu solusi.
2	Populasi	Kumpulan individu.
3	*Search space*	Semua kemungkinan solusi terhadap masalah. Hal ini mengacu pada setiap populasi pada semua individu.
4	Kromosom	Representasi solusi terhadap individu tertentu. Merupakan tahap lanjut dari desain solusi yang diharapkan terhadap masalah.
5	Alel	Bagian terkecil dari sebuah kromosom, yang berisi informasi level dasar bagaimana kromosom tersebut direpresentasikan.

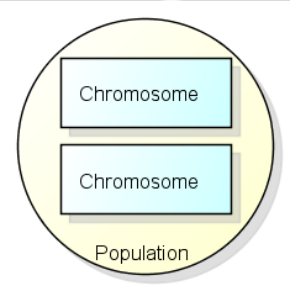
Tabel 2.2 Istilah dalam algoritma genetik (lanjutan)

No	Istilah	Penjelasan
6	Locus	Tempat gen. Merupakan *index* lokasi sebuah *genome* pada satu kromosom tertentu.
7	*Genome*	*Genome* atau gen adalah bagian dari kromosom yang terdiri dari alel. Sebuah gen dapat terdiri satu atau lebih alel.

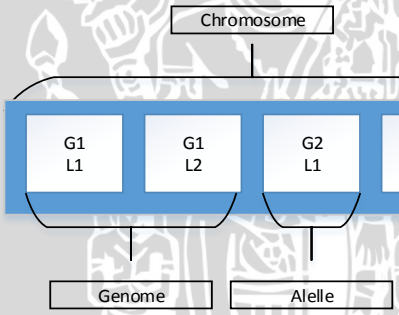
Sumber : (Mahmudy, 2013)

2.5.1 Representasi kromosom

Struktur algoritma genetik direpresentasikan oleh Gambar 2.3 dan 2.4.



Gambar 2.3 Kromosom dalam sebuah populasi



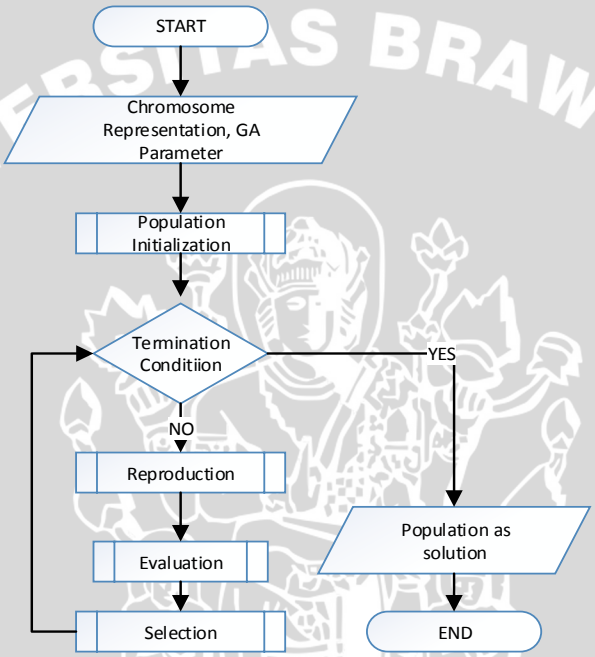
Gambar 2.4 Struktur kromosom

Pada Gambar 2.3 direpresentasikan beberapa kromosom dalam sebuah populasi. Pada Gambar 2.4, dijelaskan tentang struktur kromosom dan gen, sehingga pada level yang lebih detail $L1, L2, L3, L4$ adalah *alelle* (alel). Jika kromosom semakin panjang, maka gen dan alel berjumlah semakin banyak, tidak ada aturan baku mengenai seberapa panjang kromosom yang diijinkan. Dalam satu populasi, panjang kromosom dapat bervariasi untuk merepresentasikan solusi dengan desain solusi yang bervariasi sehingga panjang kromosom dapat tidak sama antara kromosom yang satu dengan kromosom yang lain (Kim & Wreck, 2004).

Flowchart algoritma genetik direpresentasikan pada Gambar 2.5. *Flowchart* tersebut menjelaskan bagaimana algoritma genetik bekerja untuk mendapatkan solusi, *flowchart* pada Gambar 2.5 dan penjelasan *flowchart* tersebut berdasar pada (Mahmudy, 2013).

Tahap pertama algoritma genetik dimulai dari representasi kromosom, pada tahap ini definisi masalah serta rancangan solusi yang akan dibangun sudah tersedia. Rancangan solusi tersebut kemudian melalui proses *encoding* sehingga algoritma genetik dapat membangun solusi tersebut. Hasil akhir dari proses *encoding* adalah representasi kromosom yang terdiri dari gen dan alel.

Parameter algoritma genetik juga merupakan bagian dari tahap pertama. Parameter algoritma genetik adalah nilai yang digunakan untuk menentukan *population size, crossover rate, mutation rate,* dan *termination condition*, jika *termination condition* yang digunakan adalah jumlah iterasi yang berlangsung pada algoritma genetik, maka *iteration count* termasuk parameter dalam algoritma genetik.



Gambar 2.5 Diagram alir algoritma genetik

Population size mengatur jumlah kromosom pada satu populasi tertentu pada algoritma genetik. Jumlah populasi yang diatur adalah baik jumlah kromosom untuk inisialisasi maupun jumlah kromosom setelah proses seleksi.

Crossover rate merupakan nilai yang mengatur jumlah kromosom baru yang dihasilkan dari proses *crossover*. Jumlah kromosom baru didefinisikan sebagai $jumlahKromosomBaru = crossoverRate * populationSize$. Proses dalam mendapatkan kromosom yang baru terus dilakukan pada proses *crossover* selama nilai pada $jumlahKromosomBaru$ belum terpenuhi atau belum tercapai.

Mutation rate merupakan nilai yang mengatur jumlah kromosom baru yang dihasilkan dari proses *mutation*. Jumlah kromosom baru didefinisikan sebagai $jumlahKromosomBaru = mutationRate * populationSize$. Proses dalam mendapatkan kromosom yang baru terus dilakukan pada proses *mutation* selama nilai pada $jumlahKromosomBaru$ belum terpenuhi atau belum tercapai.

Termination condition adalah kondisi yang diperlukan oleh algoritma genetik untuk menghentikan proses evolusi. *Termination condition* dapat berupa nilai *fitness* yang diraih, atau jumlah iterasi tertentu telah dicapai. Kromosom dalam populasi dikembalikan sebagai hasil dari algoritma genetik setelah melalui *termination condition*.

Pada tahap kedua, dilakukan inisialisasi populasi. Pada tahap ini sejumlah kromosom dibangkitkan, alel pada setiap kromosom dibangkitkan secara *random*. Jumlah kromosom yang dibangkitkan menurut pada *population size*.

Tahap ketiga merupakan pemeriksaan kondisi berhenti. Jika pada pemeriksaan kondisi berhenti, sudah memenuhi kriteria yang ditentukan, maka proses evolusi dihentikan dan populasi ini dikembalikan sebagai hasil dari algoritma genetik. Namun, jika belum memenuhi kriteria yang ditentukan, proses evolusi dilanjutkan hingga memenuhi kriteria yang ditentukan.

2.5.2 Reproduksi

Tahap keempat merupakan proses reproduksi, termasuk pada proses evolusi. Proses reproduksi terdiri dari dua tahap yaitu *crossover* dan *mutation*.

2.5.2.1 Pindah silang

Crossover atau pindah silang dilakukan dengan memilih dua induk (*parent*) secara acak dari populasi (Mahmudy, 2013), kemudian menghasilkan kromosom baru yang sesuai dengan *crossover rate*.

Salah satu representasi kromosom yang digunakan pada penelitian ini, memiliki panjang yang dinamis. Oleh karena itu, dibutuhkan metode *crossover* yang khusus untuk menangani panjang kromosom yang dinamis (Bentley & Wakefield, 1996).

(Bentley & Wakefield, 1996) melakukan penelitian dengan tujuan untuk melakukan *crossover* pada kromosom yang mempunyai panjang tidak sama. Metode yang diusulkan oleh (Bentley & Wakefield, 1996) adalah *hierarchical crossover*.

Hierarchical crossover merepresentasikan sebuah kromosom dalam sebuah *tree*. *Leaf* dari *tree* ini adalah alel. Titik potong yang didapatkan pada *alel* dapat diperlakukan seperti metode *single point crossover*.

2.5.2.2 Mutasi

Mutation atau mutasi dilakukan dengan memilih satu induk secara acak dari populasi, kemudian menghasilkan kromosom baru yang sesuai dengan *mutation rate* (Mahmudy, 2013).

Metode mutasi yang digunakan dalam penelitian ini adalah *reciprocal exchange mutation*. Metode tersebut melakukan mutasi dengan memilih dua orang tua yang berbeda. Kemudian, kedua alel yang dipilih secara acak pada kedua orang tua tersebut ditukar (Mahmudy, 2013).

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

2.5.3 Fungsi fitness

Tahap kelima merupakan proses evaluasi. Tahap ini bertujuan untuk mengukur seberapa baik sebuah kromosom yang berarti seberapa baik solusi tersebut terhadap masalah yang dihadapi. Evaluasi menggunakan fungsi *fitness* (Mahmudy, 2013).

Fungsi *fitness* yang digunakan pada penelitian ini dinyatakan dalam persamaan berikut :

$$Fitness = \sum_{i=0}^{i<jumlahVektorDokumen} sim(q, d_i) \quad (2.4)$$

$$d = d^T U_k S_k^{-1} \quad (2.5)$$

Dimana q sama dengan Persamaan 2.3, dan d_i sesuai dengan Persamaan 2.5. Dengan menggunakan *latent semantic indexing,* vektor dokumen yang dihasilkan pada $matrix\ V$ sama dengan Persamaan 2.5. Pada Persamaan 2.5 k sama dengan konstanta yang digunakan untuk mengurangi dimensi $matrix$ dalam *latent semantic indexing*, dan variabel d sama dengan vektor dokumen.

2.5.4 Seleksi

Tahap keenam adalah proses seleksi, proses ini bertujuan untuk memilih kromosom yang dapat berlanjut kedalam proses evolusi selanjutnya (proses generasi selanjutnya).

Metode yang digunakan dalam seleksi adalah *roulette wheel*. Metode tersebut menirukan permainan *roulette wheel* dimana setiap individu menempati potongan lingkaran pada *roulette wheel* secara proporsional sesuai dengan nilai *fitness* kromosom tersebut (Suyanto, 2007).

Pada proses ini, dipertimbangkan apakah sederetan kromosom terbaik merupakan kromosom yang dapat berlanjut kepada proses evolusi selanjutnya, ataukah memilih kromosom yang lain secara acak untuk meningkatkan keberagaman kromosom dengan harapan untuk memperoleh hasil yang lebih baik.

Pada metode *roulette wheel,* peluang kromosom atau individu dengan nilai *fitness* yang lebih besar memiliki peluang yang lebih besar juga untuk terpilih dalam generasi selanjutnya, namun karena hal ini merupakan peluang, maka tidak ada jaminan individu dengan nilai *fitness* lebih besar tersebut selalu terpilih kedalam generasi selanjutnya (Mahmudy, 2013)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 3 METODOLOGI PENELITIAN DAN PERANCANGAN

Bab III menjelaskan metode penelitian dan perancangan yang digunakan pada penelitian *Implementasi Algoritma Genetik pada Question Answering System*, dan menjelaskan tahap penelitian, studi literatur, perancangan Gambaran umum, *corpus, dataset, document preprocessing,* klasifikasi dokumen, *question answering system,* sistem basis data, dan pengujian.

3.1 Tahapan penelitian

Tahapan penelitian dijelaskan sebagai berikut :

1. Melakukan studi literatur terhadap dasar teori yang digunakan pada penelitian ini.
2. Melakukan perancangan *question answering system* dan algoritma genetik.
3. Melakukan implementasi menggunakan bahasa pemograman berdasarkan perancangan dan menguji akurasi sistem.
4. Melakukan evaluasi terhadap hasil pengujian sistem dan menberikan kesimpulan.

3.2 Studi literatur

Studi literatur menjelaskan dasar teori yang digunakan pada penelitian ini. Dasar teori yang digunakan adalah :

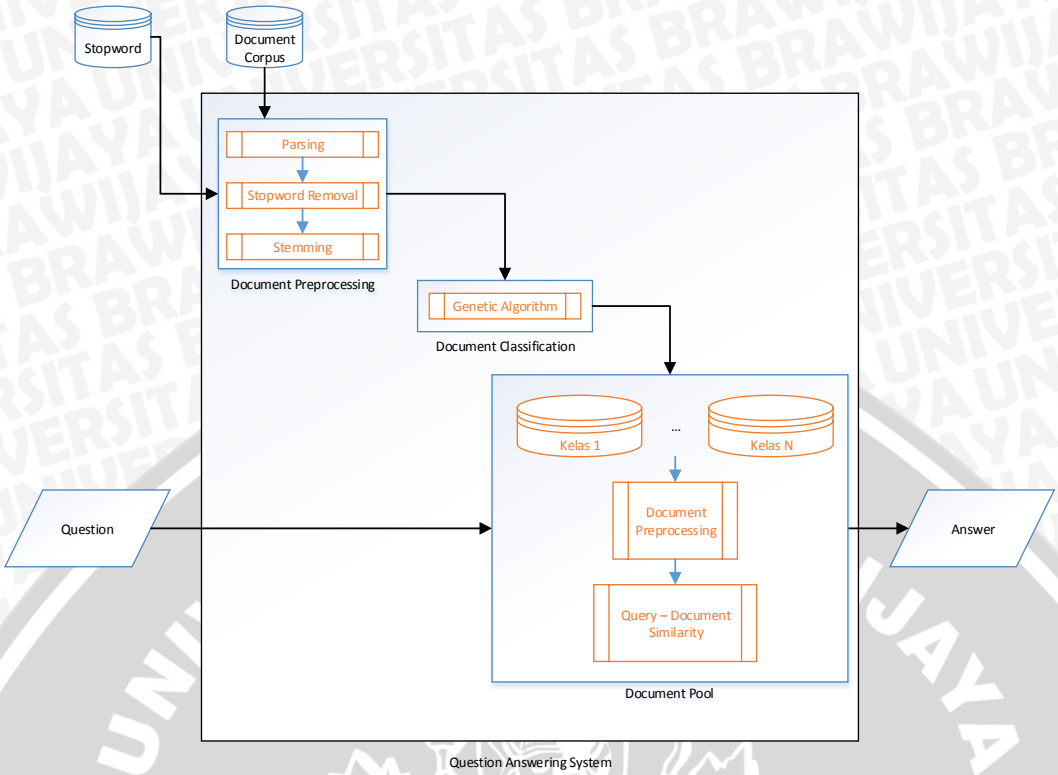
1. *Singular value decomposition*, yaitu dekomposisi *matrix* menjadi beberapa *matrix* yang lain.
2. *Latent semantic indexing,* digunakan untuk merepresentasikan setiap *term* pada setiap dokumen pada koleksi dokumen dan kueri pengguna.
3. Algoritma genetik digunakan untuk melakukan klasifikasi *term*.

3.3 Perancangan gambaran umum sistem

Untuk dapat menjawab pertanyaan pengguna, klasifikasi dokumen harus dilakukan terlebih dahulu, karena hasil klasifikasi digunakan sebagai rekomendasi jawaban atas pertanyaan pengguna.

Proses klasifikasi dimulai dengan penambahan *document corpus*, dan pemilihan *dataset* oleh admin. Kemudian, klasifikasi dilakukan menggunakan algoritma genetik. Hasil klasifikasi disimpan dalam sebuah *pool.*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 3.1 Perancangan question answering system

Pertanyaan masuk dengan format bahasa alami. Pertanyaan tersebut melalui proses awal untuk diubah menjadi sebuah vektor pertanyaan. Vektor pertanyaan mengakses *pool* untuk menemukan *term* yang sesuai. Calon kalimat jawaban direkonstruksikan dari *term* tersebut. Jawaban atas pertanyaan *pengguna* adalah kalimat yang paling relevan berdasarkan similaritas menggunakn *cosine similarity* dan *latent semantic indexing*.

Gambar 3.1 mejelaskan perancangan *question answering system* secara umum. Perancangan *document preprocessing,* klasifikasi dokumen, dan *document pool* mengacu pada perancangan *question answering system* pada Gambar 3.1

3.4 Perancangan corpus

Corpus atau *document corpus* adalah kumpulan data mentah. Data mentah adalah data yang belum dikenakan operasi apapun. Data mentah bersifat fundamental untuk sistem yang dibangun, karena tanpa data tersebut sistem tidak dapat berjalan.

Data mentah adalah dasar untuk menentukan *dataset,* dan pembentukan jawaban atas pertanyaan pengguna. Data mentah yang digunakan dalam penelitian ini adalah data berita dari *kompas.com* yang diakses pada waktu tertentu.

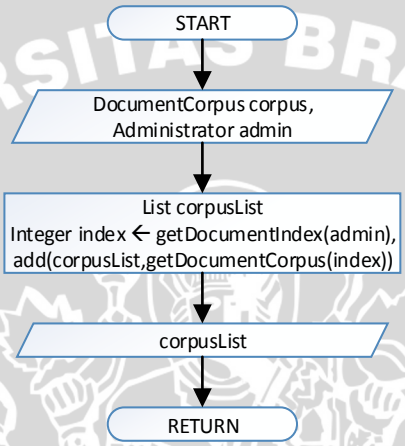
Pada penelitian ini, satu kalimat sama dengan satu dokumen.

3.5 Perancangan dataset

Dataset adalah ukuran yang digunakan sebagai pembeda antara data satu dengan data yang lain untuk dimasukkan kedalam kategori atau kelas tertentu.

Dataset dikonfigurasi oleh *admin*. *Admin* memilih beberapa kalimat dari *corpus* untuk digunakan sebagai *Dataset* dalam kelas tertentu. Dalam hal ini, *admin* dianggap orang yang mempunyai pengetahuan yang cukup untuk menentukan *Dataset*.

Representasi *Dataset* adalah kalimat dari *corpus*. Kalimat tersebut belum dikenai operasi apapun. Kalimat tersebut juga disimpan didalam sistem basis data, untuk digunakan pada setiap kelas.



Gambar 3.2 Perancangan document corpus

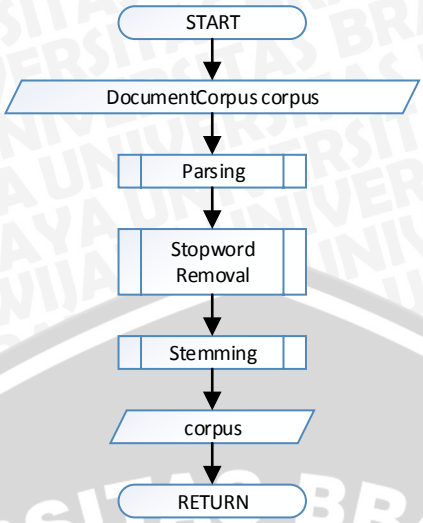
Perancangan *document corpus,* disajikan pada *flowchart* Gambar 3.2. Berdasarkan *flowchart* tersebut, dibutuhkan *document corpus,* dan seorang *administrator* sebagai masukan.

Sebuah struktur data *list* diinisialisasi. *Administrator* kemudian memilih *index* dokumen dari *document corpus*. *Index* dokumen yang dipilih oleh *administrator* kemudian disimpan didalam *list* tersebut. *List* tersebut merupakan hasil dari perancangan *document corpus*.

3.6 Perancangan document preprocessing

Document Preprocessing atau *Text Preprocessing* adalah suatu proses perubahan bentuk data yang belum terstruktur menjadi data yang terstruktur sesuai kebutuhan untuk proses *mining* atau penggalian informasi lebih lanjut (Manning, et al., 2009).

Perancangan *document preprocessing* disajikan pada *flowchart* pada Gambar 3.3



Gambar 3.3 Perancangan document preprocessing

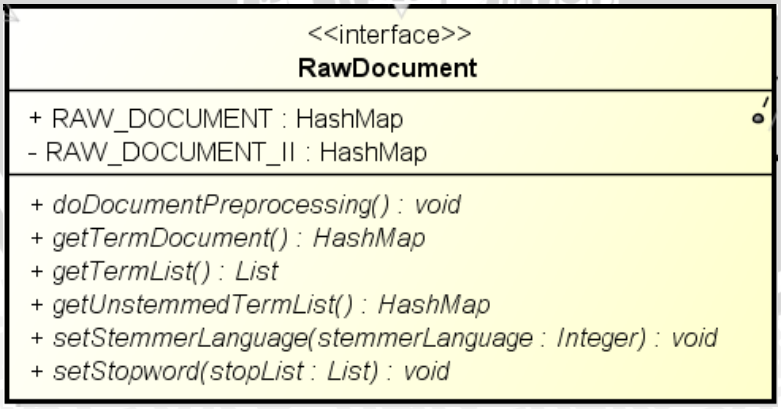
Dibutuhkan document corpus pada perancangan document preprocessing. Document corpus akan diolah dengan menggunakan document preprocessing.

Document preprocessing terdiri dari sub proses parsing, stopword removal, dan stemming yang dijelaskan pada sub bab 3.6.1, 3.6.2, dan 3.6.3.

Hasil akhir dari document preprocessing adalah daftar term yang bersifat unik. Term tersebut diurutkan secara ascending, dan term tersebut diberikan nomor urut. Nomor urut tersebut digunakan sebagai alel pada klasifikasi menggunakan algoritma genetik.

Document preprocessing berlangsung pada interface RawDocument didalam metode doDocumentPreprocessing. Metode ini dipanggil pertama kali ketika parameter algoritma genetik sudah diatur, sehingga algoritma genetik sudah siap untuk melakukan proses klasifikasi.

Interface RawDocument disajikan pada Gambar 3.4



Gambar 3.4 Perancangan interface document preprocessing

Berdasarkan Gambar 3.4 konstanta *RAW_DOCUMENT* merupakan variabel yang berisi hasil pengolahan data *corpus* hingga selesai dilakukan proses klasifikasi. Konstanta *RAW_DOCUMENT_II* menyimpan kalimat data mentah yang digunakan untuk konstruksi kalimat pada proses *question answer*.

Metode *getTermDocument* merupakan metode yang digunakan untuk mendapatkan pasangan *term-document*. Setiap *term* yang ada, dipisahkan dari kalimatnya untuk kepentingan klasifikasi. Alamat kalimat tersebut dicatat berdasarkan *term* penyusun untuk kepentingan rekonstruksi kalimat pada proses *question answer*.

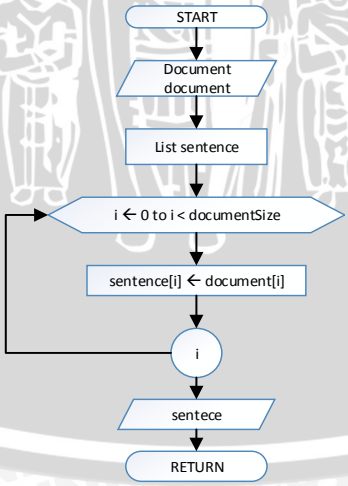
Metode *getUnstemmedTermList* merupakan metode yang digunakan untuk mendapatkan *term* sebelum *term* tersebut mengalami proses *stemming*.

Metode *setStemmerLanguage* merupakan metode yang menerima konstanta untuk pengaturan bahasa pada proses *stemming*. Pengaturan bahasa yang digunakan pada penelitian ini adalah bahasa Indonesia.

Metode *setStopWordList* merupakan metode untuk mengisikan *stopword* dari sistem basis data. Jika sudah terdapat *stopword* yang diatur, maka pengaturan tersebut akan dibatalkan, kemudian digantik oleh pengaturan *stopword* yang baru. Metode ini digunakan pada proses *stopword removal,* yaitu sebuah proses yang bertujuan untuk menghilangkan kata yang dianggap tidak informatif pada sebuah dokumen. Pada penelitian ini, satu dokumen merupakan satu kalimat.

3.6.1 Perancangan parsing

Parsing merupakan langkah yang bertujuan untuk memecah dokumen menjadi beberapa bagian tertentu. Pada tahap ini sebuah unit dokumen dibuat. Dalam penelitian ini, definisi satu dokumen adalah satu kalimat (Manning, et al., 2009).



Gambar 3.5 Perancangan parsing

Parsing membutuhkan masukan berupa dokumen dari *corpus*. Pada *flowchart* Gambar 3.5, dokumen dari *corpus* diwakili oleh objek dokumen. Objek dokumen tersebut berisi kumpulan dokumen *corpus* dari sistem basis data.

Sebuah *list* yang diwakili oleh objek kalimat dinisialisasi. Kalimat yang disimpan dalam *list* ini akan disimpan secara terurut menurut iterasi.

Langkah berikutnya adalah melakukan sebuah iterasi terhadap objek dokumen. Iterasi tersebut dilakukan dari *index* ke 0 dari objek dokumen hingga akhir dokumen *corpus* yang terdapat didalam objek dokumen.

Dalam iterasi tersebut, setiap kalimat dalam objek dokumen dipindahkan kedalam *list* tersebut. Proses ini disebut juga sebagai tokenisasi, dimana kalimat adalah *token* dari dokumen (Manning, et al., 2009).

Hasil akhir dari tahap *parsing* adalah kalimat dari dokumen tertentu pada *corpus* dengan struktur data *list*.

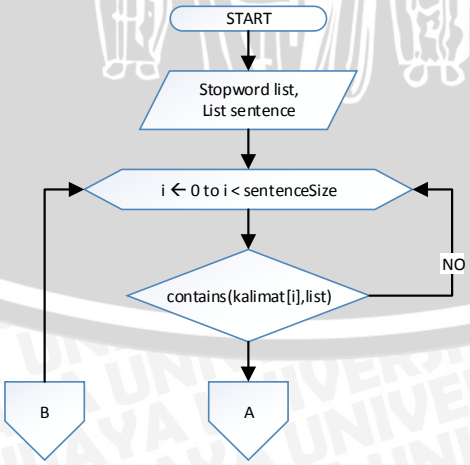
List kalimat tersebut dikembalikan sebagai hasil dari sub proses *parsing*. *Interface RawDocument* mengirim *list* kalimat kedalam sistem basis data.

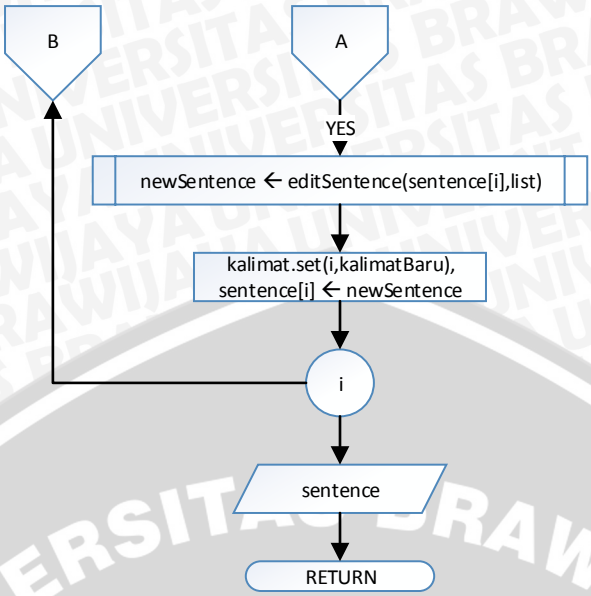
3.6.2 Perancangan stopword removal

Dalam *interface RawDocument*, *stopword removal* diimplementasikan didalam metode *doDocumentPreprocssing*. Konstanta *RawDocument* berisi kalimat dari sistem basis data yang telah mengalami *parsing*. Koleksi kalimat *RawDocument* akan diperbarui menggunakan *stopword removal* pada tahap ini.

Document stopword removal atau *document filtering* adalah tahap pengambilan dari hasil *token,* yaitu kata – kata atau *term* apa saja yang akan digunakan untuk merepresentasikan dokumen (Manning, et al., 2009).

Token pada tahap ini adalah *list* kalimat dari hasil *parsing*. Kalimat tersebut akan dipecah menjadi koleksi kata dan dilakukan seleksi kata pada tahap ini. Tahap *stopword removal* ini dijelaskan oleh *flowchart* pada Gambar 3.6





Gambar 3.6 Perancangan stopword removal

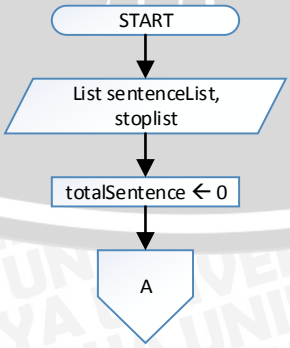
Stopword list berisi *term* menurut bahasa masing-masing, misalnya *stopword* dalam bahasa Inggris dan bahasa Indonesia berbeda.

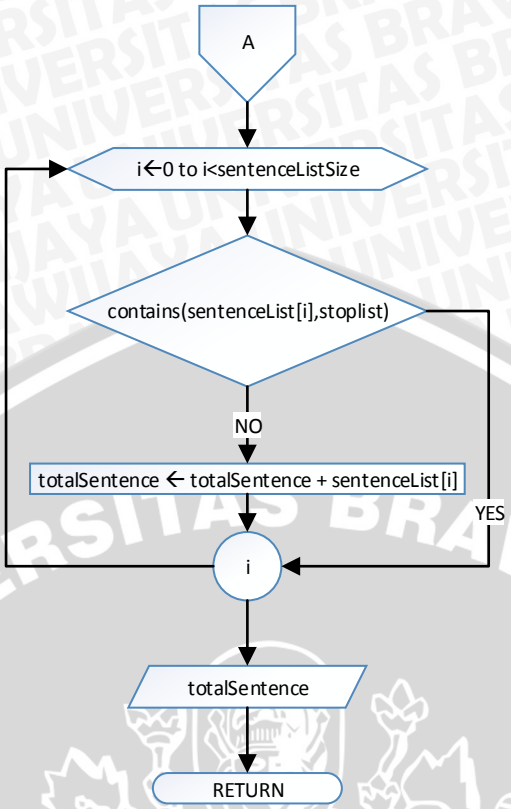
Algoritma yang digunakan pada *flowchart* 3.6 adalah algoritma *stoplist* (Manning, et al., 2009). Pada algoritma ini, *term* hasil tokenisasi atau *list* kalimat dari proses *parsing* dihilangkan jika mengandung *term* yang sama dengan *stoplist*.

Pada tahap *stopword removal,* dibutuhkan *stopword list* yang berasal dari sistem basis data, *list* kalimat yang berasal dari sub proses *parsing*.

Untuk setiap kalimat dalam *list* kalimat, dilakukan sebuah iterasi. Jika kalimat ke i pada *list* kalimat mengandung *term* dalam *list stoplist*, maka kalimat tersebut dimasukkan kedalam sub proses *editSentence* dengan parameter berupa kalimat dan *list stopword*. Kalimat baru adalah hasil dari sub proses *editSentence*.

Kalimat baru hasil sub proses *editSentence* diset kembali kedalam *list* kalimat. *List* kalimat yang sudah dimodifikasi dikembalikan sebagai hasil dari sub proses *stopword removal*.





Gambar 3.7 Perancangan edit sentence

Berdasarkan Gambar 3.7, sub proses *edit sentence* membutuhkan masukan berupa kalimat dan *stoplist*. Kalimat merupakan hasil dari sub proses *stopword removal* dan *stoplist* adalah kumpulan *term* yang akan dihilangkan pada kalimat.

Kalimat total diinisialisasi sebagai hasil yang akan dikembalikan dari sub proses *edit sentence*.

Untuk semua *term* dalam kalimat *list* kalimat, dilakukan sebuah iterasi. Jika *stoplist* tidak mengandung *term* pada kalimat ini, maka kalimat total akan direkonstruksikan dengan menambah *term* ini kepada kalimat total. Sehingga kalimat total tidak mengandung *term* pada *stoplist*.

Kalimat total dikembalikan sebagai hasil dari sub proses *edit sentence*. Kalimat total tersebut dikembalikan kepada konstanta *RawDocument* pada *interface RawDocument*.

3.6.3 Perancangan stemming

Stemming merupakan proses mengubah suatu kata tertentu menjadi kata dasar (Manning, et al., 2009). *Stemming* dalam bahasa tertentu berbeda dengan bahasa yang lain.

Pada penelitian ini, proses *stemming* diimplementasikan dalam sebuah *interface RawDocument* yang disajikan pada Gambar 3.4. Pada *interface* tersebut proses *stemming* dilakukan didalam metode *doDocumentPreprocessing*.

Proses *stemming* dapat dilakukan ketika pengaturan bahasa sudah diset. Hal ini dilakukan karena untuk menghindari ambiguitas bahasa yang dilakukan pada proses *stemming*. Beberapa bahasa tertentu memiliki kata serapan dari bahasa yang lain. Kata tersebut mendapatkan perlakukan berbeda pada satu bahasa terhadap bahasa yang lain. Perlakukan yang didapatkan antara lain adalah penambahan imbuhan, dan perubahan struktur kata dalam sebuah kalimat. Hal ini dapat menimbulkan ambigutas antara kata yang diperlakukan sebagai serapan pada satu bahasa, terhadap kata asli pada bahasa yang lain.

Algoritma *stemming* yang digunakan dalam penelitian ini adalah algoritma *porter stemmer*. Mekanisme algoritma *porter stemmer* adalah membuang kata imbuhan dalam kata tersebut (Porter, 2006).

3.6.4 Perancangan term indexing

Perancangan *indexing* merupakan langkah yang dilakukan dengan tujuan untuk memasukkan *term* yang unik kedalam struktur data *hashmap*. Perancangan *indexing* pada tahap ini tidak melalui proses *weighting* sehingga menghitung nilai *term frequency—inverted index frequency,* namun hanya menambahkan *term* yang unik dari koleksi data *corpus* untuk menjadi *key* didalam struktur data *hashmap*.

Value dari *hashmap* tersebut adalah perhitungan *term frequency—inverted index frequency* yang digunakan sebagai perhitungan *fitness* kromosom. Penjelasan lebih lanjut tentang perhitungan *term frequency—inverted index frequency* dijelaskan pada perancangan fungsi *fitness* pada klasifikasi dokumen menggunakan algoritma genetik.

Digunakan struktur data hashmap pada perancangan term indexing. Struktur data hashmap diGambarkan pada Gambar 3.8

Key	Value
String	Object

Gambar 3.8 Perancangan hashmap

Gambar 3.8 merupakan *hashmap* yang digunakan. *Hashmap* terdiri dari dua elemen utama, yaitu *key* dan *value*. *Key* dan *value* didalam *hashmap* berpasangan satu sama lain. *Key* bersifat unik. Satu *key* digunakan untuk mengambil satu *value* yang sesuai dengan *key* tersebut (Kadir, 2003).

Key yang didefinisikan pada tahap ini bertipe *string*. Isi dari *key* tersebut adalah *term* dari dokumen *corpus*. *Value* yang bertipe *object* dengan nilai *null*. Nilai

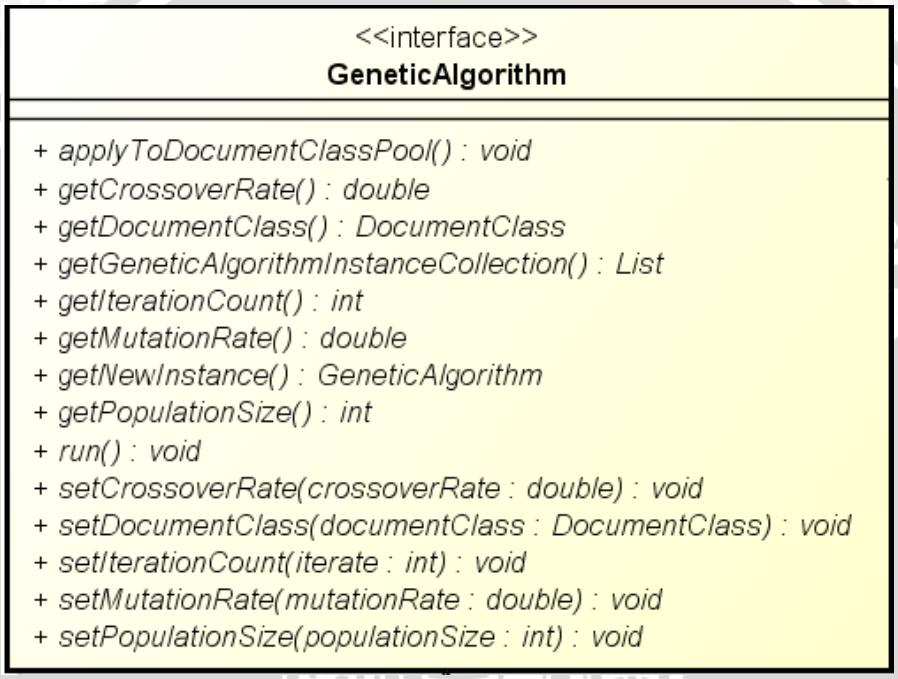
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

null tersebut akan diisi dengan perhitungan *term frequency—inverted document frequency* pada fungsi *fitness* algoritma genetik.

3.7 Perancangan klasifikasi dokumen

Pada penelitian ini, klasifikasi dokumen merupakan langkah yang dilakukan untuk mengelompokkan dokumen kedalam sebuah kelompok atau satu kelas tertentu. Dokumen yang telah dikelompokkan kedalam kelas tertentu, menjadi kunci rekonstruksi kalimat dalam pencarian jawaban.

Pada penelitian ini beberapa proses dalam klasifikasi diimplementasikan dalam sebuah klas. Rancangan klas tersebut disajikan pada Gambar 3.9



Gambar 3.9 Perancangan interface menggunakan algoritma genetik

Pada Gambar 3.9 disajikan sebuah *interface* yang bertugas untuk melakukan konfigurasi parameter algoritma genetik, dan melakukan klasifikasi *term* menggunakan algoritma genetik.

Proses klasifikasi *term* menggunakan algoritma genetik dilakukan pada metode *run*. Metode tersebut akan melakukan hal sebagai berikut :

1. Memeriksa data *corpus* didalam sistem basis data.
2. Memeriksa apakah klasifikasi dokumen sudah dilakukan terhadap *corpus* tersebut.
3. Memeriksa apakah parameter algoritma genetik sudah diatur.

Pada pemeriksaan data *corpus* , dilakukan perhitungan berapa jumlah data yang tersedia. Jika tidak ada data yang tersedia, maka sistem akan mengembalikan pesan peringatan kesalahan tidak adanya data *corpus* tersebut.

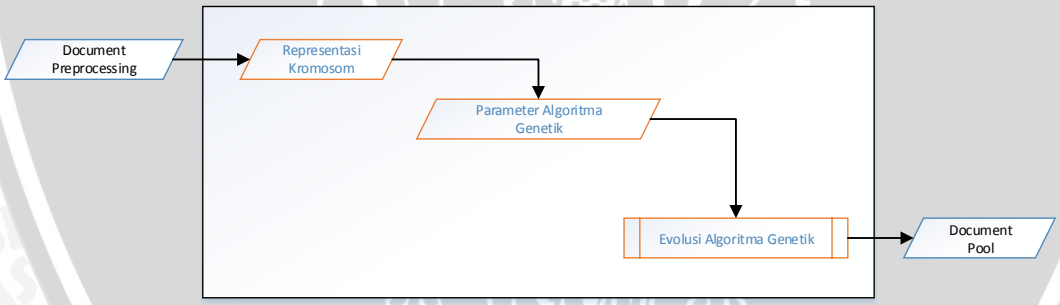
Data *corpus* disimpan dalam tipe data *binary large object* atau *blob* pada sistem basis data. Dengan menggunakan tipe data tersebut, maka sembarang tipe data yang dibutuhkan didalam fase implementasi dapat disimpan ke sistem basis data. Hal ini juga berlaku untuk semua *file* yang dibutuhkan pada tahap implementasi.

Jika data *corpus* telah tersedia, maka sistem akan melanjutkan pemeriksaan apakah klasifikasi dokumen telah dilakukan. Pada pemeriksaan tersebut dilakukan perhitungan jumlah *term* hasil klasifikasi dalam sistem basis data. Jika tidak ada *term* yang terdapat pada sistem basis data, maka dapat disimpulkan bahwa proses klasifikasi dokumen belum dilakukan.

Klasifikasi dokumen dilakukan pada metode *doDocumentClassification.* Perancangan metode klasifikasi dokumen tersebut disajikan pada Gambar 3.10.

Pada perancangan klasifikasi dokumen tersebut, secara garis besar *document preprocesing* menjadi masukan, dan *document pool* menjadi keluaran bagi sub proses klasifikasi dokumen ini. Hal ini sesuai dengan perancangan *question answering system* secara umum yang disajikan pada Gambar 3.1

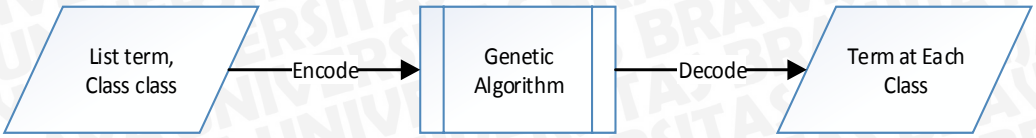
Algoritma yang digunakan dalam klasifikasi dokumen ini adalah algoritma genetik. Berdasarkan Gambar 3.10, representasi kromosom dan parameter algoritma genetik harus dipersiapkan sebelum melakukan proses evolusi algoritma genetik.



Gambar 3.10 Perancangan klasifikasi dokumen

3.8 Perancangan klasifikasi kromosom algoritma genetik

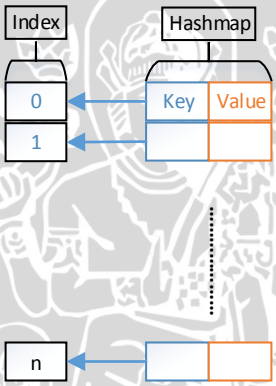
Pada perancangan kromosom, masalah dikodekan dengan kromosom, kemudian dilakukan evolusi pada kromosom tersebut, dan kromosom hasil evolusi terakhir yang mempunyai nilai *fitness* terbaik didekodekan menjadi solusi yang mendekati optimum dari masalah tersebut (Mahmudy, 2013).



Gambar 3.11 Perancangan masalah dan solusi menggunakan algoritma genetik

Perancangan masalah yang hendak dipecahkan, dan solusi yang diinginkan disajikan pada Gambar 3.11. Setelah melalui *document preprocessing,* didapatkan *term*. Masalahnya adalah bagaimana mengklasifikasikan *term* tersebut kepada setiap kelas. Solusi yang diinginkan adalah *term* pada setiap kelas.

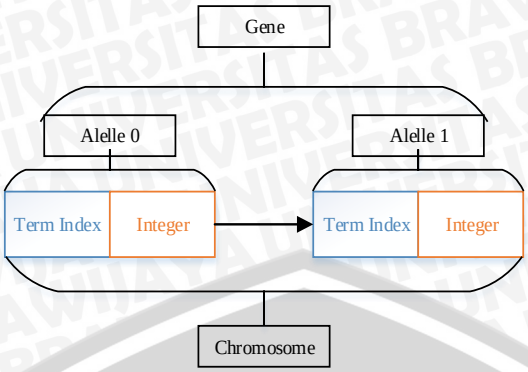
Term yang didapatkan masih didalam struktur data *hashmap.* Dilakukan pengurutan *term* secara *ascending* didalam *hashmap* secara *ascending*. Pengurutan *term* tersebut dilakukan karena tidak ada jaminan pada *hashmap* mengenai keterurutan *key* pada *hashmap* tersebut. Setelah *term* tersebut terurut, maka diberikan nomor yang disebut dengan *index* yang terkait dengan *key* dari *hashmap* tersebut. Struktur *hashmap* yang sudah diurutkan dan diberi *index* disajikan pada Gambar 3.12



Gambar 3.12 Perancangan indexed hashmap

Berdasarkan Gambar 3.12, setiap pasangan *key*−*value* sudah memiliki *index*. *Index* tersebut mulai dari 0 hingga n, sesuai dengan jumlah pasangan *key*−*value* yang terdapat pada *hashmap* tersebut.

Index pada *hashmap* tersebut digunakan untuk membentuk alel pada kromosom. Perancangan struktur kromosom yang meliputi alel, gen, dan kromosom yang digunakan pada algoritma genetik diGambarkan pada Gambar 3.13



Gambar 3.13 Perancangan struktur kromosom

Berdasarkan Gambar 3.13, 1 kromosom terdiri dari 1 gen, dan 1 gen terdiri dari 1 atau lebih alel. Gen tersebut merepresentasikan *term* yang dimuat didalam alelnya. Alel tersebut memiliki nomor *index* yang diketahui oleh gen.

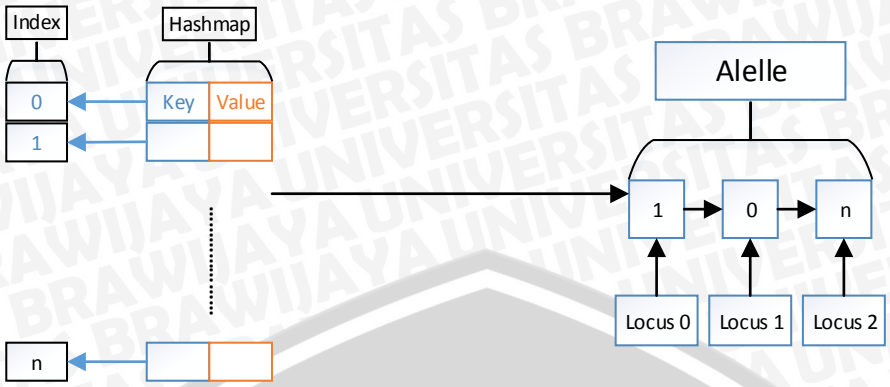
Alel dapat terdiri dari 1 atau lebih, panjang alel tersebut adalah dinamis. Digunakan alel dinamis, karena mungkin saja solusi yang baik dibentuk oleh rangkaian alel yang pendek. Jika demikian, maka alel yang panjang akan menkonsumsi lebih banyak memori. Isi dari alel tersebut adalah informasi *index term* yang terdapat pada *indexed hashmap*. *Index* pada *alel* bersifat unik, artinya jika satu *term* sudah masuk pada satu alel didalam satu gen, dan satu kromosom tertentu, maka *term* tersebut tidak akan masuk kembali menjadi anggota dari alel tersebut pada gen, dan kromosom yang sama.

Kromosom menggunakan struktur data *linked list.* Struktur data tersebut disajikan pada Gambar 3.14

Gambar 3.14 Perancangan struktur data kromosom

Berdasarkan Gambar 3.14, satu *node* terdiri dari *integer* dan *node link.* Tipe data *integer* merupakan *index* yang didapatkan dari *indexed hashmap* atau dapat disebut dengan *term index* yang sesuai dengan Gambar 3.13. *Node link* merupakan petunjuk kepada *node* yang lain.

Perancangan langkah *indexed hashmap* menjadi kromosom diGambarkan pada Gambar 3.15.

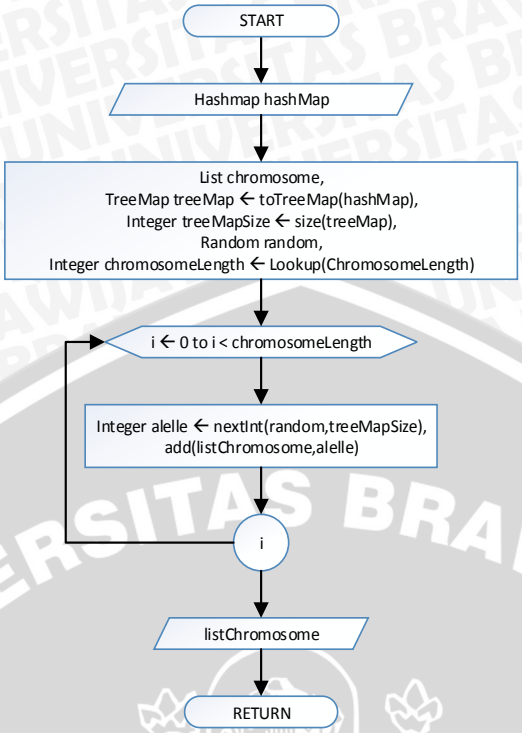


Gambar 3.15 Perancangan representasi struktur data kromosom

Berdasarkan Gambar 3.15, *hashmap* yang sudah memiliki *index* direpresentasikan kedalam struktur data *list*. *Index* dari *list* tersebut disebut dengan lokus yang berarti tempat sebuah alel berada.

Alel pada Gambar 3.15 merupakan *index* pada *hashmap* yang diambil secara acak. Batas bawah nilai pada alel sama dengan 0, karena *index* pada *hashmap* dimulai dari 0. Batas atas nilai pada alel sama dengan n, sesuai dengan jumlah *term* yang ada pada *hashmap*.

Langkah pada perancangan representasi kromosom disajikan pada *flowchart* pada Gambar 3.16



Gambar 3.16 Perancangan representasi kromosom

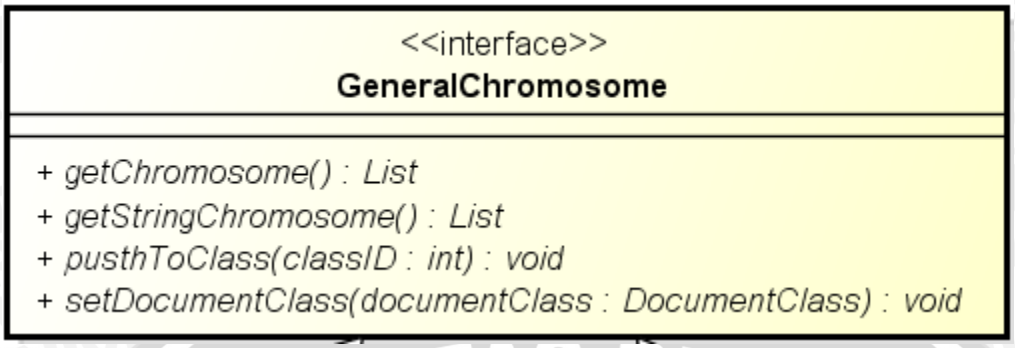
Berdasarkan Gambar 3.16, *hashmap* menjadi masukan bagi perancangan kromosom. *Hashmap* tersebut berisi *term* hasil dari *document preprocessing*. Proses yang dilakukan pada langkah selanjutnya adalah :

1. Melakukan inisialisasi pada sebuah struktur data *list*. Struktur data *list* tersebut akan dikembalikan sebagai hasil dari proses ini.
2. Menginisialisasi sebuah objek dengan tipe *Tree Map*. *Treemap* merupakan sebuah variasi dari *hashmap* dengan fitur tambahan yaitu pengurutan *key*. Dengan menggunakan *tree map,* maka *term* pada *hashmap* akan diurutkan, kemudian diberikan *index* yang sesuai (Kadir, 2003).
3. Ukuran kromosom didapatkan dari parameter panjang kromosom. Panjang kromosom yang dimaksud adalah panjang kromosom evolusi pertama algoritma genetik. Ukuran panjang kromosom pada evolusi terakhir dapat berubah.
4. Sebuah objek dengan tipe data *random* diinisialisasi. Objek ini digunakan untuk menentukan menentukan nilai alel.

Sebuah iterasi dilakukan. Iterasi tersebut dilakukan sebanyak panjang kromosom. Karena kromosom tersebut terdiri dari satu gen, maka panjang kromosom sama dengan panjang alel.

Iterasi tersebut bertujuan untuk mengisi nilai pada alel. Nilai pada alel bertipe *integer* dengan struktur data *list*. Nilai alel sama dengan nilai yang dihasilkan dari objek *random* dengan batas atas berupa ukuran *tree map*.

Setelah iterasi tersebut berakhir, maka *list* kromosom adalah hasil dari perancangan kromosom. Perancangan kromosom tersebut diimplementasikan dalam sebuah *interface* yang disajikan pada Gambar 3.17



Gambar 3.17 Perancangan interface representasi kromosom

Metode *getChromosome* diimplementasikan untuk mengambil kromosom sesuai dengan perancangan kromosom pada Gambar 3.17.

Metode *getStringChromosome* diimplementasikan untuk mengambil representasi *term* dengan tipe data *string*.

Metode *pushToClass* diimplementasikan untuk menjadikan kromosom ini menjadi bagian dari kelas tertentu. Kromosom ini diambil representasi *term* dengan tipe data *string,* kemudian *string* ini yang dijadikan anggota kelas tertentu.

Klas yang mengimplementasikan *interface* ini tidak mempunyai operasi tambahan, atau atribut tambahan.

3.9 Perancangan parameter algoritma genetik

Pada algoritma genetik dibutuhkan paramater untuk mengatur proses evolusi yang terjadi (Mahmudy, 2013). Fungsi dari parameter algoritma genetik yang digunakan dalam penelitian ini dijelaskan secara ringkas pada Tabel 3.1.

Tabel 3.1 Parameter algoritma genetik

No	Parameter	Keterangan
1	Iteration Count	Kondisi berhenti jika iterasi ini telah tercapai.
2	Population Size	Jumlah individu dalam satu populasi.
3	Crossover Rate	Mengatur banyak anak yang dihasilkan dari proses *crossover*.
4	Mutation Rate	Mengatur banyak anak yang dihasilkan dari proses *mutation*.
5	Chromosome Length	Mengatur panjang kromosom pada inisialisasi.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

6	Informative Term Threshold	Fokus kepada beberapa *term* teratas berdasarkan pengurutan menggunakan *term frequency-inverted document frequency.*
7	Dimension Rate	Fitur pengurangan dimensi pada metode *latent semantic indexing*.

Parameter nomor 1 hingga nomor 4 merupakan parameter yang mengatur evolusi algoritma genetik. Parameter nomor 5, 6 dan 7 merupakan parameter tambahan untuk mengatur performansi sistem, sehingga proses klasifikasi dapat dilakukan dengan waktu yang relatif lebih cepat dengan hasil yang relatif akurat.

Parameter nomor 5 merupakan digunakan untuk mengatur panjang kromososom.penjang kromosom harus diatur karena *term* akan memanjang seiring dengan bertambahnya dokumen *corpus*. *Term* yang mencapai lebih besar dari 1000 membutuhkan waktu yang relatif lama untuk menyusun kromosom pada performansi sistem. Jumlah *term* yang banyak menyebabkan jumlah alel juga relatif banyak, sehingga anak hasil *crossover* dan *mutation* juga berukuran banyak. Hal ini menyebabkan proses evolusi algoritma genetik berlangsung lebih lama.

Parameter nomor 6 merupakan *informative term threshold* yang digunakan untuk memilih beberapa *term* teratas yang digunakan untuk penyusunan kromosom. Konfigurasi *informative term* akan memberikan nilai tambahan kepada fungsi *fitness* algoritma genetik apabila mengandung *term* ini. Jika *dataset* yang digunakan merupakan koleksi *term* pilihan yang dikonfigurasikan oleh seseorang yang menguasai domain masalah tertentu, maka parameter *informative term* akan mengarahkan algoritma genetik pada evolusi yang selanjutnya untuk menghasilkan kromosom yang sedapat mungkin memiliki *term* yang terdapat pada *informative term*.

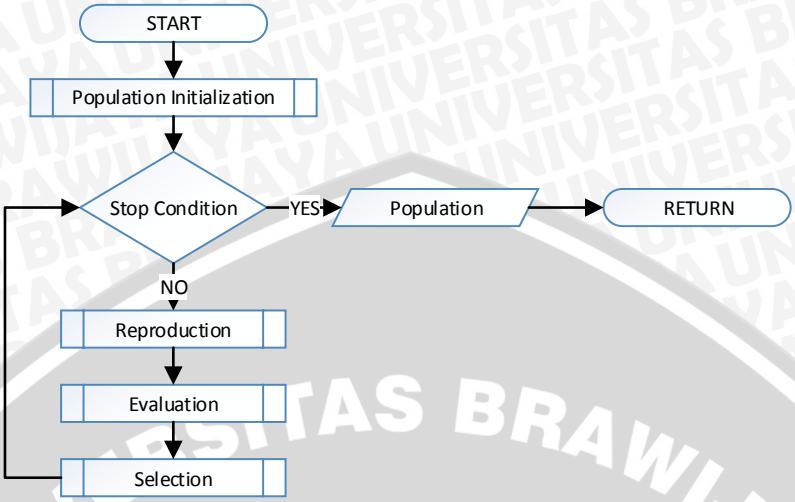
Parameter nomor 7 merupakan fitur pengurangan dimensi pada metode *latent semantic indexing*. Jika parameter tersebut diaktifkan dengan cara mengisi nilai parameter tersebut dengan nilai lebih dari 0, maka ukuran dimensi *matrix* pada metode *latent semantic indexing* mengikuti konfigurasi parameter ini.

Pengurangan dimensi *matrix* dilakukan untuk mengurangi kompleksitas operasi pada perhitungan *fitness* algoritma genetik, dan mengurangi waktu evolusi algoritma genetik. Besaran pengurangan dimensi memang ditentukan berdasarkan beberapa percobaan, namun dalam beberapa percobaan yang telah diteliti, didapatkan kesimpulan dimensi *matrix* yang mendekati 100 memberikan hasil yang baik (Grossman & Frieder, 2004).

3.10 Perancangan evolusi algoritma genetik

Ketika semua parameter algoritma genetik sudah dilengkapi dan representasi kromosom sudah siap untuk diimplementasikan, maka proses evolusi algoritma

genetik dapat dilakukan. Proses evolusi algoritma genetik disajikan didalam *flowchart* pada Gambar 3.18



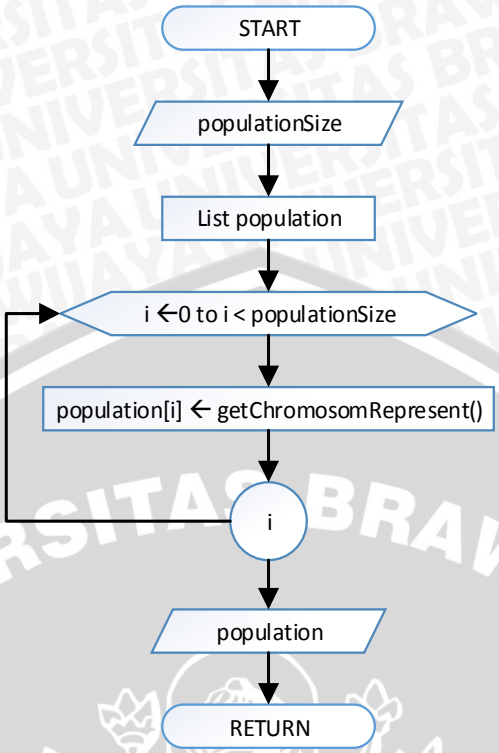
Gambar 3.18 Perancangan evolusi algoritma genetik

Langkah pertama dalam proses evolusi algoritma genetik adalah inisialisasi populasi. Tujuan dari inisialisasi populasi adalah menghasilkan representasi kromosom yang sesuai dengan solusi permasalahan dengan jumlah kromosom yang dihasilkan adalah sama dengan parameter algoritma genetik umlah populasi.

Inisialisasi populasi dilakukan apabila belum terdapat populasi pada algoritma genetik. Jika belum ada populasi, maka proses evolusi tidak dapat dilakukan. Jika sudah ada populasi, maka populasi tersebut dilakukan proses evolusi.

3.10.1 Perancangan inisialisasi populasi

Perancangan sub proses inisialisasi populasi disajikan dalam *flowchart* pada Gambar 3.19



Gambar 3.19 Perancangan inisialisasi populasi

Berdasarkan Gambar 3.19, masukan untuk inisialiasi populasi adalah parameter algoritma genetik jumlah populasi. Parameter tersebut digunakan untuk menentukan jumlah kromosom atau individu didalam satu populasi.

Dilakukan sebuah iterasi terhadap jumlah populasi. Didalam iterasi tersebut kromosom baru ditambahkan kedalam sebuah struktur data *list*. *List* tersebut merupakan sebuah populasi yang terdiri dari kromosom yang dibangkitkan. Setelah iterasi tersebut berakhir, maka *list* populasi dikembalikan sebagai hasil dari inisialisasi populasi.

3.10.2 Perancangan kondisi berhenti

Kondisi berhenti evolusi algoritma genetik yang digunakan dalam penelitian ini adalah berdasarkan nilai parameter algoritma genetik jumlah iterasi.

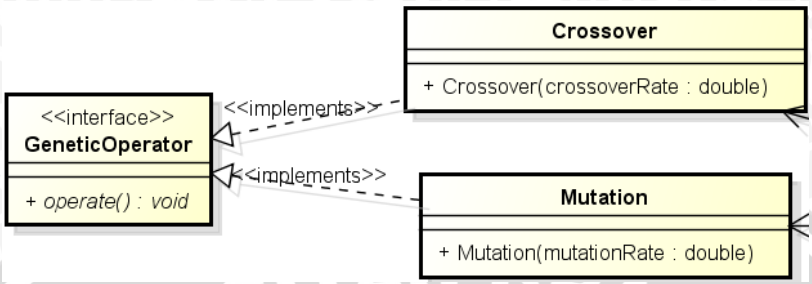
Jumlah iterasi menghitung iterasi evolusi pada algoritma genetik. Pada setiap evolusi, iterasi tersebut bertambah satu. Jika jumlah iterasi sama dengan batas yang diatur dalam parameter algoritma genetik, maka evolusi algoritma genetik dihentikan.

3.10.3 Perancangan reproduksi

Gambar 3.20 merupakan sebuah *interface* beserta klas implementornya. *Crossover* dan *mutation* termasuk kedalam operator genetik sehingga dapat

dikelompokkan dengan nama fungsi yang sama, namun dengan implementasi yang berbeda.

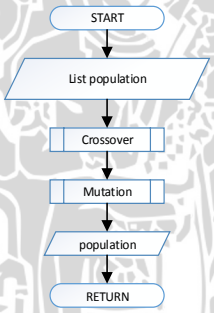
Fungsi tersebut bernama *operate,* pada klas *Crossover* fungsi tersebut melakukan proses *crossover,* sedangkan pada klas *Mutation* fungsi tersebut melakukan proses mutasi.



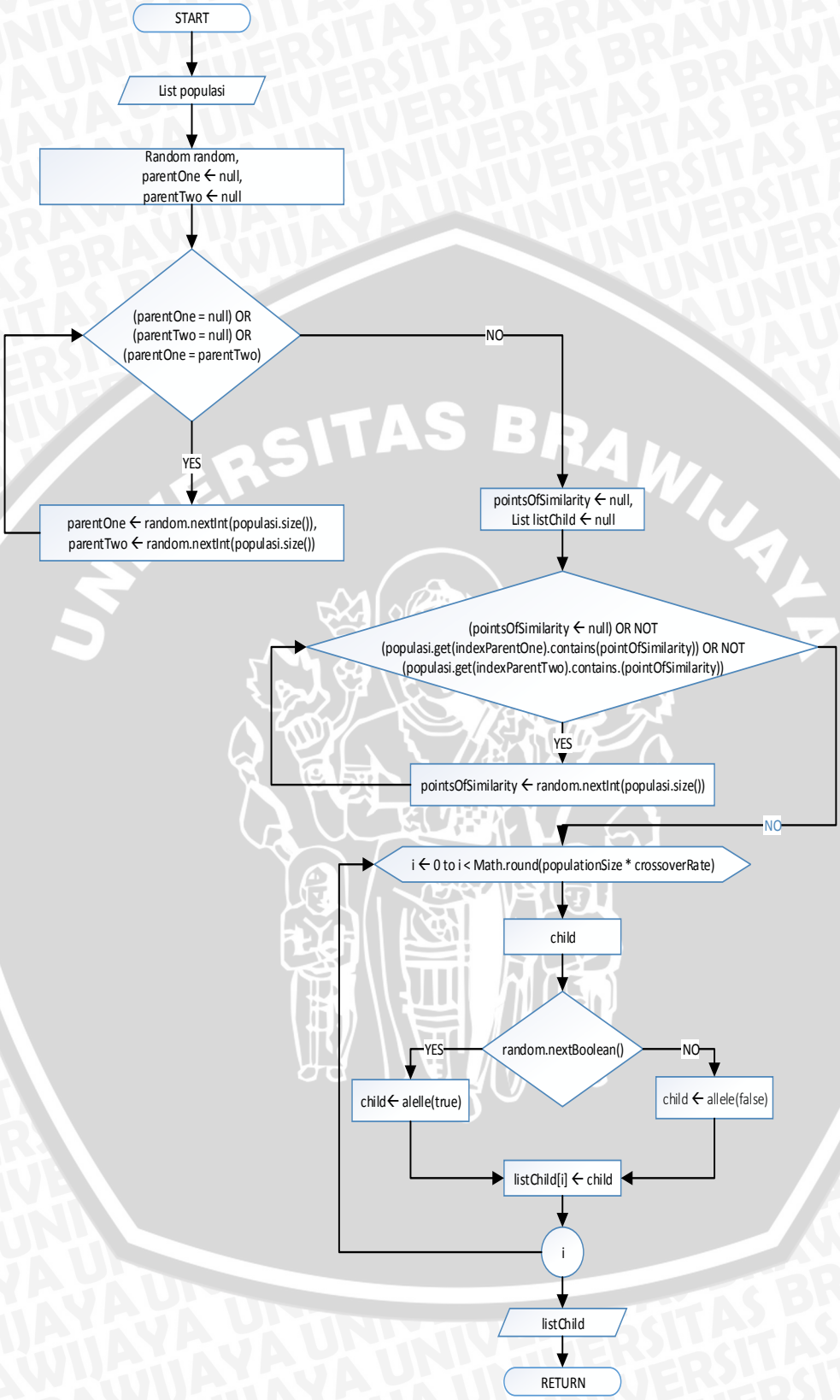
Gambar 3.20 Perancangan interface operator genetik

Reproduksi dilakukan untuk menghasilkan keturunan dari individu-individu yang terdapat didalam sebuah populasi (Mahmudy, 2013). Reproduksi diimplementasikan dengan operator genetika. Operator genetika yang digunakan dalam penelitian ini adalah *crossover* dan *mutation* (Mahmudy, 2013).

Perancangan reproduksi disajikan dalam *flowchart* pada Gambar 3.21



Gambar 3.21 Perancangan reproduksi



Gambar 3.22 Perancangan hierarchical crossover

Proses *crossover* tersebut membutuhkan *list* populasi untuk melakukan *crossover*. *List* tersebut tidak diperiksa apakah berisi kromosom atau tidak, karena *list* tersebut merupakan *instance* objek yang sama dari sub proses sebelumnya.

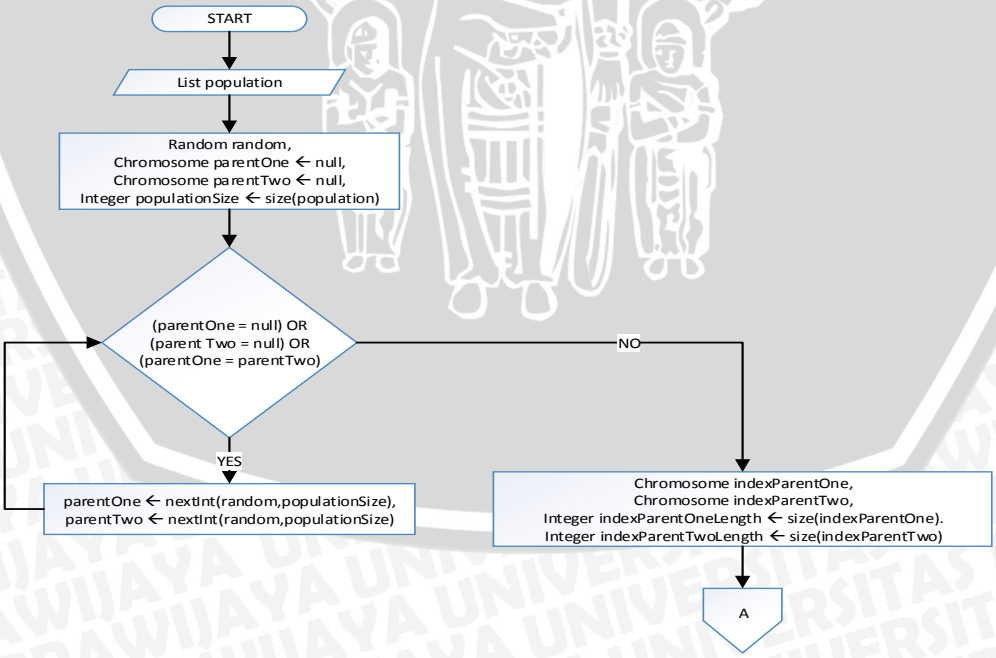
Langkah selanjutnya adalah memilih orang tua secara acak. Hal tersebut diimplementasikan dalam sebuah perulangan yang menjamin bahwa kedua orang tua tersebut didapatkan, dan kedua orang tua tersebut tidak sama.

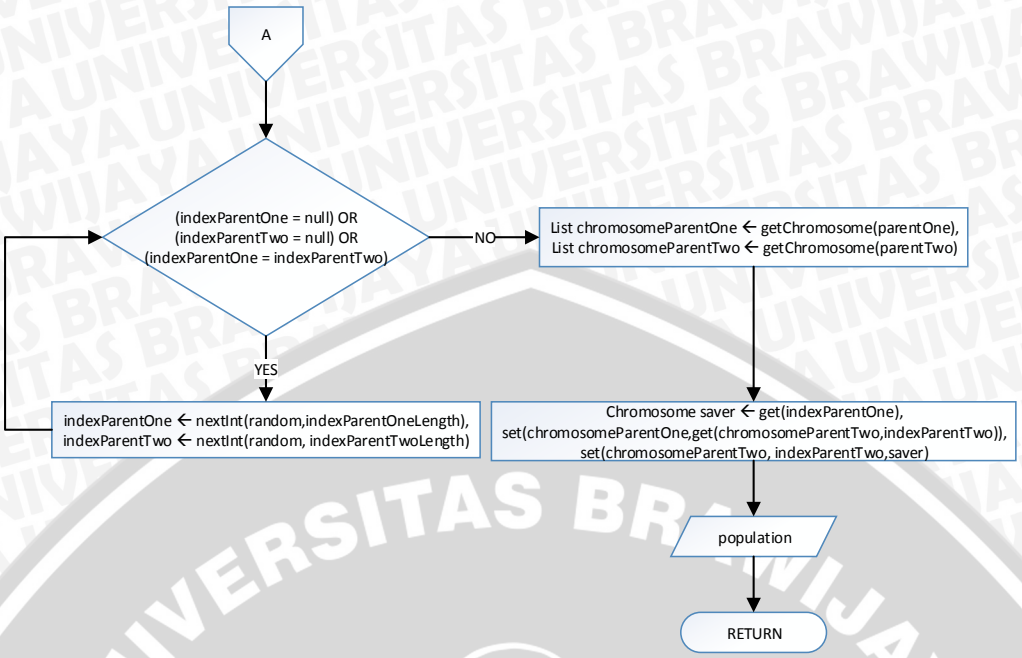
Point of similarity merupakan titik kesamaan pada kedua orang tua yang digunakan dalam *crossover*. Meskipun panjang kedua orang tua tidak sama, namun *point of similarity* harus bernilai sama.

Setelah *point of similarity* ditemukan, maka langkah selanjutnya adalah menghasilkan anak dari kedua orang tua tersebut. Hal ini diimplementasikan dengan sebuah perulangan dengan kondisi sebanyak anak yang dibutuhkan sesuai dengan parameter *crossover rate*. Dalam perulangan ini sebuah anak diinisialisasi. Kemudian dilakukan pemilihan kondisi benar atau salah. Jika kondisi benar, maka anak tersebut akan menerima alel kedua orang tua tersebut bagian kiri, namun jika kondisi salah, maka anak tersebut akan menerima alel kedua orang tua tersebut bagian kanan.

Anak yang lahir dari proses *crossover,* disimpan dalam sebuah *list child*. *List* tersebut merupakan hasil akhir dari proses *crossover*.

Berdasarkan Gambar 3.21, sub proses selanjutnya adalah mutasi. Metode mutasi yang digunakan didalam penelitian ini adalah *reciprocal exchange mutation*. Perancangan metode mutasi disajikan dalam sebuah *flowchart* pada Gambar 3.23





Gambar 3.23 Perancangan mutasi

Pada *flowchart* tersebut *list* populasi merupakan masukan untuk sub proses mutasi. *List* populasi berasal dari proses *crossover*.

Langkah selanjutnya adalah memilih orang tua secara acak. Proses pemilihan orang tua tersebut diimplementasikan didalam sebuah iterasi. Iterasi tersebut menjamin bahwa kedua orang tua yang dipilih tidak mungkin bernilai *null,* dan kedua orang tua tersebut tidak mungkin sama.

Titik tukar kedua orang tua didapatkan, titik tukar tersebut digunakan untuk menukarkan alel pada kedua orang tua dalam menghasilkan anak. Setelah kedua titik tukar lokus didapatkan, maka langkah selanjutnya adalah menukarkan alel pada setiap lokus. Hal tersebut diimplementasikan dengan mendapatkan *list* kromosom dari masing-masing orang tua. Kemudian, nilai alel di atur ulang untuk setiap orang tua.

List populasi adalah hasil akhir dari proses mutasi. *List* populasi menggunakan struktur data *list*. *Instance list* populasi pada awal proses mutasi dan akhir proses mutasi adalah sama. Jadi, tidak perlu membuat *instance* baru dengan nilai yang baru juga, karena setiap perubahan dalam *instance* yang sama akan berpengaruh terhadap *instance* tersebut.

3.10.4 Perancangan evaluasi

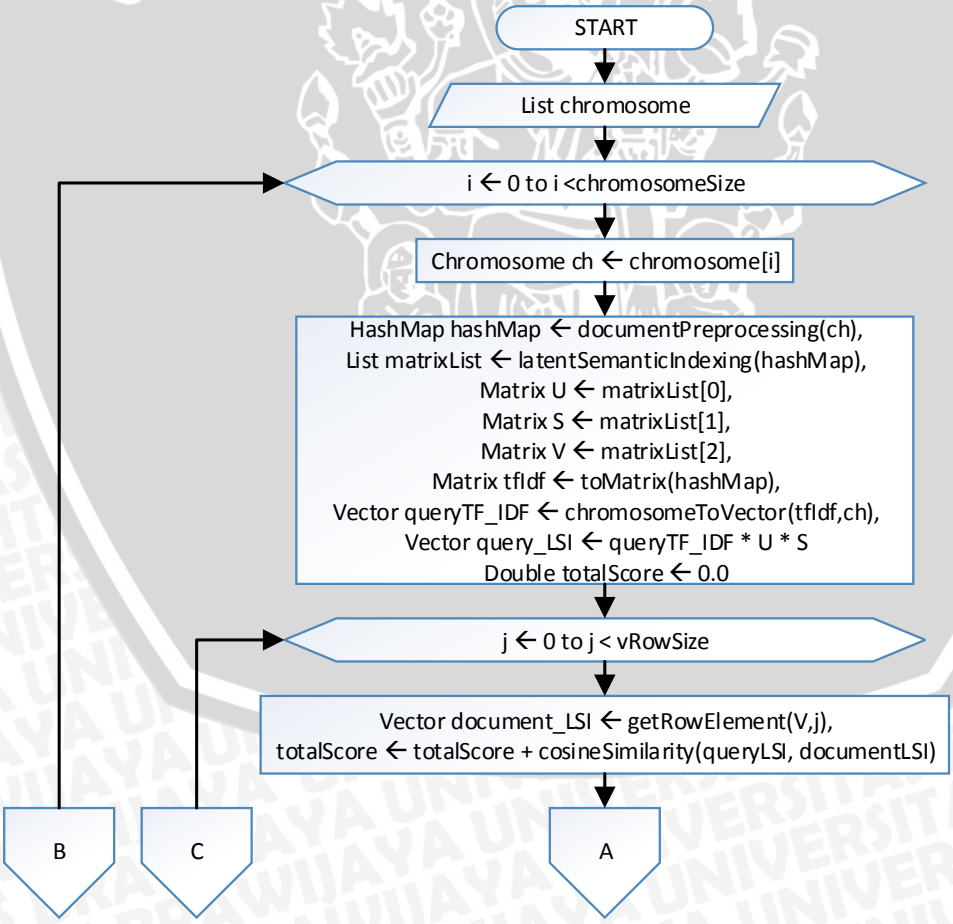
Proses evaluasi dilakukan untuk mengukur seberapa baik kromosom tersebut dalam menyelesaikan masalah. Proses evaluasi tersebut dinyatakan dalam sebuah fungsi yang disebut dengan *fungsi fitness* (Mahmudy, 2013).

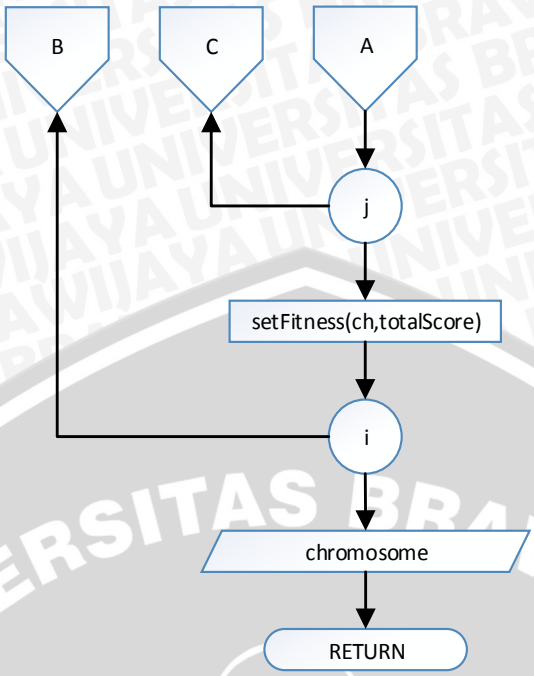
Dibutuhkan sebuah *dataset* untuk mengimplementasikan fungsi *fitness*. *Dataset* merupakan fakta awal atau informasi awal yang dijadikan referensi oleh algoritma genetik terhadap fungsi *fitness*. Dokumen yang terdapat didalam *dataset* merupakan contoh dokumen yang relevan didalam kelas tertentu, oleh karena itu dibutuhkan seseorang yang dianggap memiliki pengetahuan yang cukup didalam konfigurasi *dataset*. Klasifikasi algoritma genetik akan berusaha mendekati konfigurasi *dataset*. Semakin bagus *dataset* yang dikonfigurasi, maka semakin baik hasil klasifikasi menggunakan algoritma genetik.

Fungsi fitness yang diimplementasikan didalam peneltian ini ditulis didalam Persamaan 2.4. Persamaan tersebut melakukan iterasi kepada vektor kueri terhadap seluruh koleksi vektor dokumen. Vektor kueri dan vektor dokumen yang digunakan berasal dari pengolahan *latent semantic indexing*.

Satu kromosom dianggap sebagai satu kueri, dan koleksi dokumen adalah seluruh koleksi dokumen didalam kelas tersebut. Pada tahap awal koleksi dokumen pada setiap kelas adalah *dataset*. Namun setelah proses evolusi algoritma genetik berakhir, maka koleksi dokumen merupakan *dataset* dan seluruh koleksi dokumen dari hasil evolusi algoritma genetik.

Perancangan evaluasi kromosom disajikan pada *flowchart* Gambar 3.24





Gambar 3.24 Perancangan evaluasi

Berdasarkan Gambar 3.24, perancangan evaluasi kromosom membutuhkan masukan *list* kromosom. Pada *list* tersebut terdapat satu atau lebih kromosom dengan nilai *fitness* yang belum dihitung.

Dilakukan iterasi terhadap *list* tersebut. Didalam iterasi tersebut, satu kromosom dikeluarkan dengan representasi sebagai satu variabel. Dilakukan *document preprocessing* yang melibatkan kromosom tersebut.

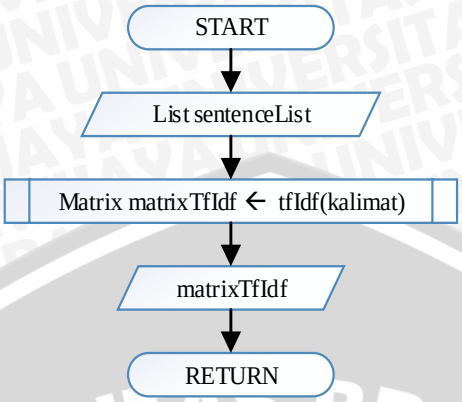
Pada *document preprocessing,* kromosom tersebut berperan sebagai *query* dan *dataset* berperan sebagai dokumen.

Koleksi dokumen yang digunakan adalah *dataset* pada setiap kelas. Operasi *parsing, stopword removal,* dan *stemming* telah dilakukan sebelumnya pada dokumen *corpus*, sehingga pada koleksi dokumen pada *dataset* juga telah mengalami operasi tersebut. Pada tahap ini dilakukan *weighting*.

Weighting merupakan proses pembobotan setiap kata terhadap setiap dokumen. Hasil dari pembobotan ini nilai yang menjadi properti kata pada kalimat tersebut. Pembobotan merupakan langkah yang sangat penting dalam *information retrieval,* karena setiap informasi yang akan muncul nantinya berdasarkan nilai pada pembobotan tersebut (Manning, et al., 2009).

Berbagai metode telah diusulkan untuk proses pembobotan. Diantaranya adalah *Jaccard Coefficient,* dan *Term Frequency – Inverted Document Frequency*. Penelitian ini menggunakan metode *Term Frequency – Inverted Document Frequency*, karena metode tersebut mampu mengukur seberapa informatif sebuah kata terhadap kata yang lain dalam koleksi dokumen, serta bobot dari

metode tersebut mampu dikombinasikan dengan *latent semantic indexing* (Paulson, 2010).



Gambar 3.25 Perancangan weighting

Weighting membutuhkan *list* kalimat sebagai masukan. *List* kalimat yang digunakan sudah melalui tahap *stemming*.

Tujuan dari sub proses *weighting* adalah untuk menghasilkan *matrix tfidf.* Dalam proses menghasilkan *matrix tfidf* , dibutuhkan struktur data *hashmap*.

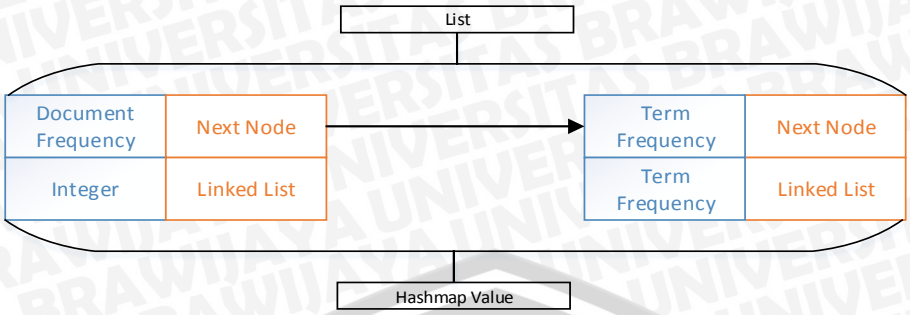
Struktur data *hashmap* tersebut digunakan karena *hashmap* dirancang untuk bersifat generik dengan pasangan kunci dan nilai (*key and value pair*), artinya *hashmap* tersebut dapat beradaptasi dengan sembarang tipe data yang diimplementasikan menjadi parameter generik *hashmap* tersebut (Kadir, 2003).

Struktur data *hashmap* disajikan dalam Gambar 3.8. *Hashmap* diimplementasikan berdasarkan pasangan kunci dan nilai. Kunci pada *hashmap* bernilai unik. Jika pada sebuah *hashmap* akan dimasukkan data dengan kunci yang sudah ada pada *hashmap,* maka data tersebut akan ditolak.

Dalam penelitian ini, *hashmap* digunakan untuk menghitung *term frequency—inverted document frequency*. Berdasarkan keperluan perhitungan tersebut, maka *key* pada *hashmap* bertipe *string* yang berisi satu kata secara unik pada koleksi dokumen, sedangkan *value* pada *hashmap* bertipe objek. Objek ini kemudian diisi dengan struktur data *list* seperti gambar 3.26

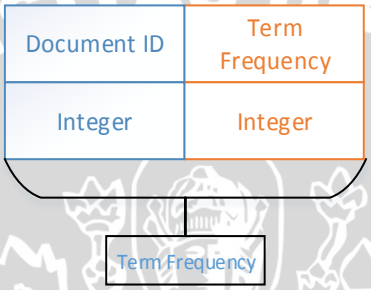
Gambar 3.26 merupakan sebuah *list*. Pada *node* pertama *list* tersebut, diimplementasikan perhitungan *document frequency*. Kemudian, pada *node* yang lain diimplementasikan perhitungan *term frequency*.

Document frequency menghitung kejadian satu *term* pada dokumen tertentu. Implementasi *document frequency* menggunakan sebuah variabel dengan tipe data *Integer*. Variabel tersebut akan ditambah nilainya berdasarkan kejadian *term* tertentu pada dokumen tertentu.



Gambar 3.26 Perancangan hashmap value

Term frequency menghitung kejadian satu *term* pada satu dokumen tertentu. *Term frequency* diimplementasikan dalam sebuah variabel dengan tipe data *Term Frequency*. Struktur data *term frequency* disajikan pada Gambar 3.27



Gambar 3.27 Perancangan struktur data term frequency

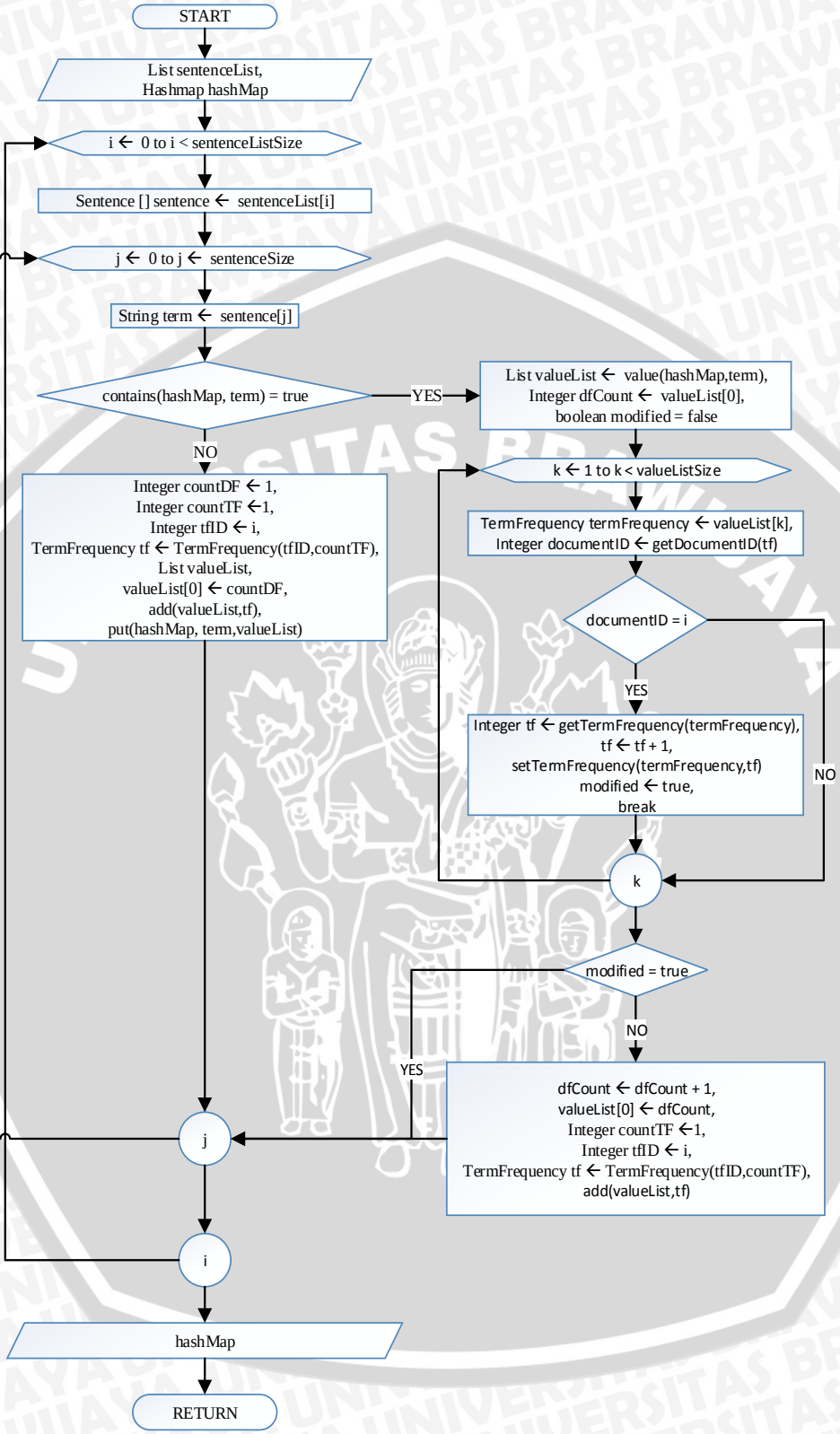
Berdasarkan Gambar 3.27, *document id* merupakan identitas dokumen, dan *term frequency* merupakan variabel yang digunakan untuk menghitung *term frequency*. Satu *term* pada satu dokumen, hanya memiliki satu struktur data yang bertipe *term frequency*.

Term frequency dan *document frequency* diimplementasikan bersama dalam proses perhitungan. Jika *term frequency* mendapatkan nilai nilai tertentu yang sesuai dengan kejadian *term* pada dokumen tertentu, maka *document frequency* untuk *term* tertentu terhadap sebuah dokumen juga akan berubah nilanya.

Perancangan perhitungan *document frequency* dan *term frequency* disajikan didalam *flowchart* pada Gambar 3.28.

Berdasarkan *flowchart* pada Gambar 3.28, dalam perhitungan *term frequency* dan *document frequency* membutuhkan sebuah struktur data *list* yang berisi koleksi kalimat sebagai masukan, dan sebuah struktur data *hashmap* sebagai hasil akhir.

Langkah selanjutnya melakukan iterasi terhadap seluruh koleksi dokumen. Iterasi tersebut berjalan satu kali terhadap koleksi dokumen. Tujuan dari iterasi tersebut adalah untuk memperbarui struktur data *hashmap* yang digunakan sebagai hasil dari proses ini.



Gambar 3.28 Perancangan perhitungan *term frequency* dan *document frequency*

Dalam iterasi tersebut diambil kalimat dari *sentenceList*. Kemudian dilakukan iterasi terhadap satu kalimat tersebut. Iterasi terhadap satu kalimat tersebut menghasilkan *term* dari kalimat tersebut.

Pada iterasi terhadap satu kalimat tersebut, langkah pertama yang dilakukan adalah memeriksa apakah *term* tersebut sudah ada didalam *hashmap.*

Jika *term* tersebut belum ada didalam *hashmap*, maka ada empat proses yang dilakukan. Proses tersebut adalah :

1. Menambahkan *term* tersebut sebagai *key* didalam *hashmap*.
2. Menghitung *document frequency* terhadap *term* yang baru tersebut. Karena *term* tersebut baru, maka nilai *document frequency* sama dengan satu. *List* pada *index* ke 0 digunakan untuk menyimpan *document frequency.*
3. Menghitung *term frequency* terhadap *term* yang baru tersebut. Karena *term* tersebut baru, maka *term frequency* bernilai sama dengan 1.
4. Menyimpan data *term frequency* dan *document frequency* didalam struktur data *hashmap*.

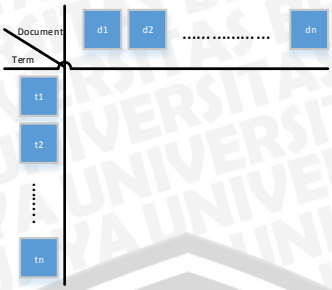
Jika *term* tersebut sudah ada didalam *hashmap,* maka dilakukan pembaruan nilai *term frequency* dan *document frequency*. Hal tersebut dilakukan dengan :

1. Mendapatkan *list* yang merupakan *value* dari *term* tersebut.
2. Mendapatkan *document frequency* pada *index* ke 0.
3. Melakukan sebuah iterasi terhadap *list* untuk menemukan objek *TermFrequency* dengan *documentID* yang sesuai dengan dokumen pada iterasi sekarang. Sebuah penanda bertipe *boolean* digunakan untuk memutuskan apakah *documentID* tersebut ditemukan atau tidak. Jika *documentID* tersebut ditemukan, maka dilakukan pembaruan terhadap *TermFrequency* tersebut, hal ini diimplementasikan dengan menambah *term frequency counter* sebesar satu. Jika *documentID* tersebut tidak ditemukan, maka *document frequency counter* dinaikkan sebesar satu, dan objek *TermFrequency* diniisialisasi lagi dengan nilai sama dengan satu untuk dokumen pada iterasi sekarang.

Iterasi kemudian dilanjutkan kepada *term* selanjutnya, dan dokumen selanjutnya, hingga seluruh dokumen telah selesai melalui iterasi.

Hashmap merupakan hasil akhir dari proses perhitungan *document frequency,* dan *term frequency*. *Hashmap* tersebut diolah lebih lanjut sehingga mendapatkan nilai *term frequency*−*inverted document frequency* dalam struktur data $matrix$. Struktur data $matrix$ tersebut disajikan pada Gambar 3.29

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 3.29 Perancangan struktur data *matrix*

Berdasarkan struktur data $matrix$ pada Gambar 3.29, kolom *matrix* tersebut adalah dokumen, dan baris $matrix$ tersebut adalah *term*. Hasil perhitungan *term frequency—inverted document frequency* disimpan pada $matrix$ ini.

Digunakan struktur data $matrix$ karena pengolahan yang dilakukan selanjutnya menggunakan metode *latent semantic indexing* yang menggunakan perhitungan $matrix$. Kemudian, perhitungan similaritas antara vektor dokumen kueri terhadap koleksi vektor dokumen menggunakan struktur data $matrix$ juga.

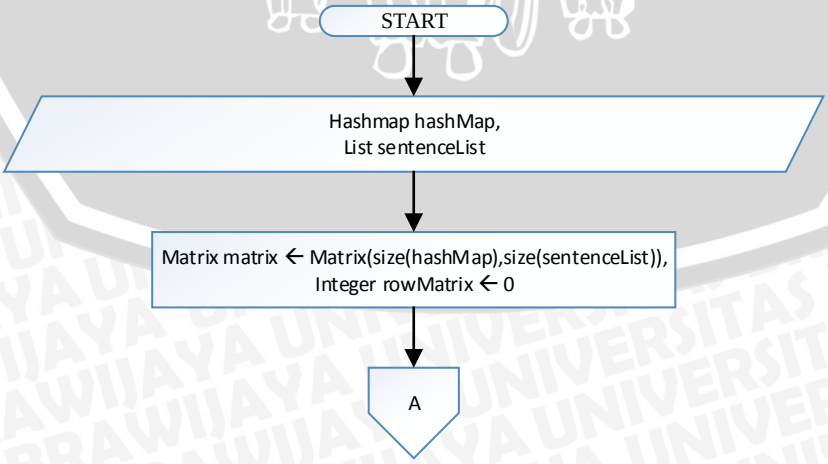
Perancangan perhitungan *term frequency—inverted document frequency* dengan struktur data *matrix* disajikan pada *flowchart* pada Gambar 3.30.

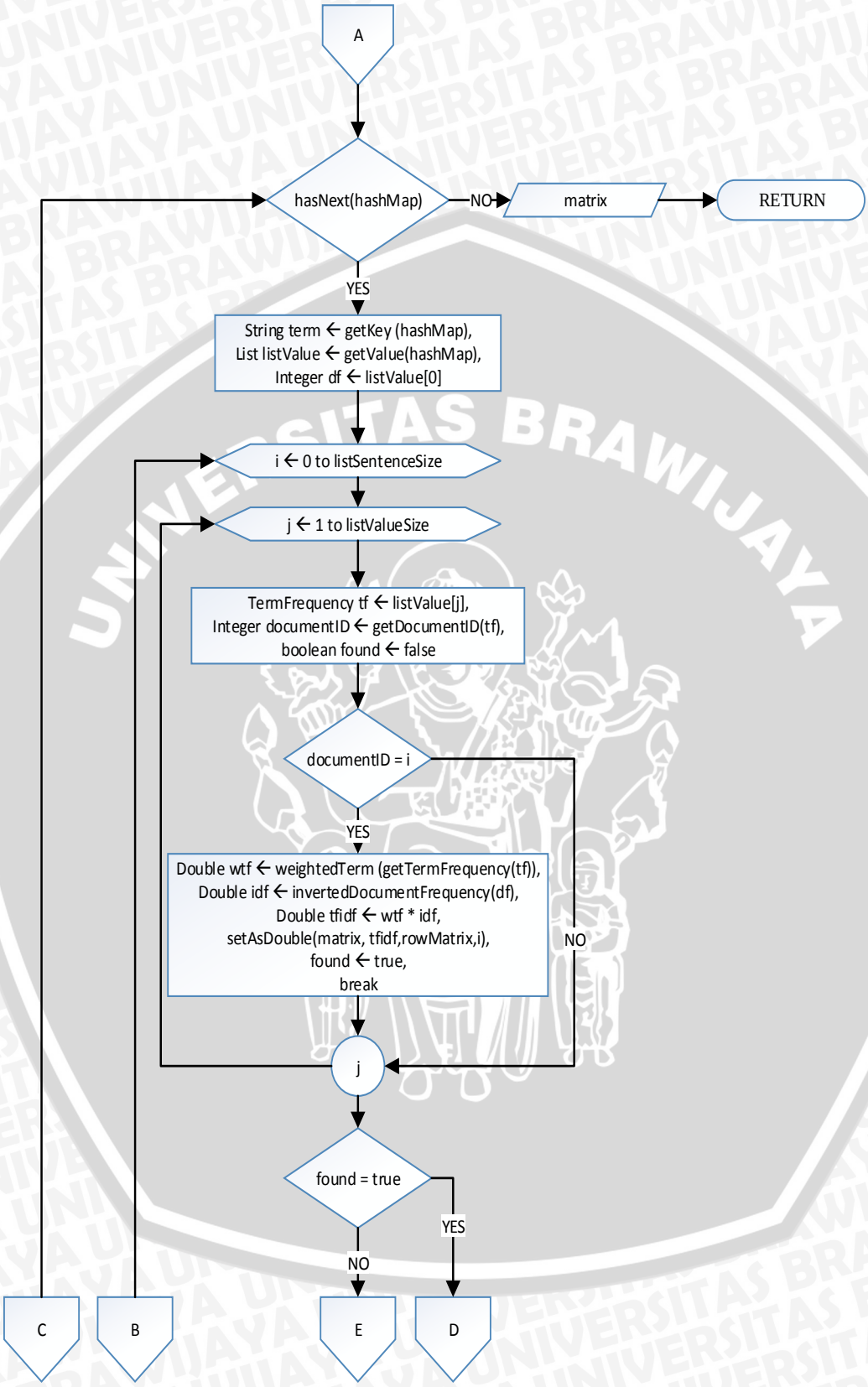
Berdasarkan *flowchart* pada Gambar 3.30, perhitungan *term frequency—inverted document frequency* dengan struktur data $matrix$, membutuhkan masukan berupa :

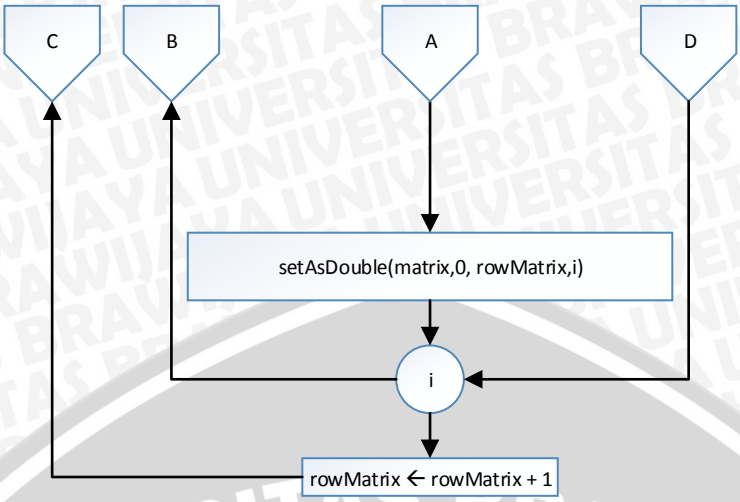
1. *Hashmap* hasil pengolahan *term frequency—document frequency*.
2. *List* kalimat yang berisi seluruh daftar dokumen.

Kemudian sebuah struktur data $matrix$ yang diimplementasikan oleh objek *matrix* dilakukan inisialisasi. *Matrix* tersebut memiliki jumlah baris sama dengan ukuran *hashmap,* dan jumlah kolom sama dengan jumlah *list* kalimat.

Sebuah variabel yang bertugas sebagai penunjuk baris pada *matrix* dinisialisasi. Setiap kenaikan satu baris *matrix,* maka nilai variabel ini ditambah sebesar satu.




A
hasNext(hashMap)
NO
matrix
RETURN
YES
String term ← getKey (hashMap),
List listValue ← getValue(hashMap),
Integer df ← listValue[0]
i ← 0 to listSentenceSize
j ← 1 to listValueSize
TermFrequency tf ← listValue[j],
Integer documentID ← getDocumentID(tf),
boolean found ← false
documentID = i
YES
Double wtf ← weightedTerm (getTermFrequency(tf)),
Double idf ← invertedDocumentFrequency(df),
Double tfidf ← wtf * idf,
setAsDouble(matrix, tfidf,rowMatrix,i),
found ← true,
break
NO
j
found = true
YES
NO
C
B
E
D



Gambar 3.30 Perancangan perhitungan TF-IDF

Diimplementasikan sebuah perulangan dengan kondisi jika masih ada *item* yang belum diiterasi pada *hashmap,* maka lakukan iterasi. *Item* tersebut terdiri dari *key* dan *value* pada *hashmap*.

Kemudian dilakukan langkah sebagai berikut :

1. Mengambil pasangan *key* dan *value,* serta menaruh kedalam variabel.
2. Dilakukan sebuah iterasi terhadap seluruh koleksi dokumen yang direpresentasikan pada *listSentence*.
3. Dilakukan sebuah iterasi terhadap seluruh koleksi dari struktur data *list* dari *value hashmap*. *List* tersebut berisi objek *term frequency* yang menyimpan informasi kejadian satu *term* didalam satu dokumen tertentu.
4. Dilakukan sebuah pemeriksaan apakah *documentID* pada iterasi sekarang sama dengan *documentID* yang terdapat didalam objek *term frequency*.
5. Apabila kondisi pada nomor 4 bernilai benar, maka dilakukan perhitungan *weighted term, inverted document frequency,* dan *term frequency—inverted document frequency,* dengan menggunakan Persamaan 3.1, 3.2, dan 3.3 secara berturut-turut. Ketiga Persamaan tersebut disajikan sebagai berikut :

$$idf_t = \log_{10}(N/df_t) \tag{3.1}$$

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

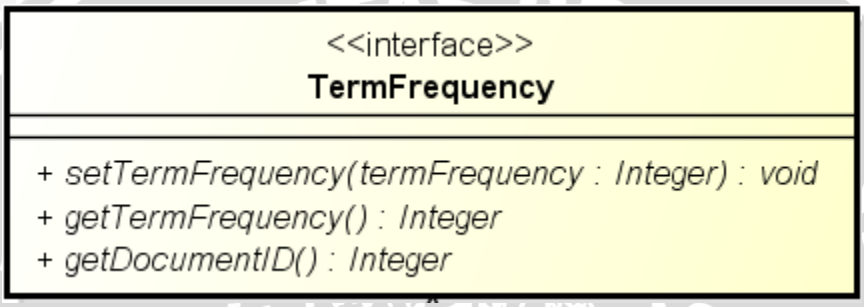
$$w_{t,d} = \log(1 + tf_{t,d}) * \log_{10}(N/df_t) \tag{3.3}$$

Kemudian menyimpan hasil perhitungan tersebut didalam *matrix* dengan *index* baris berupa variabel *rowMatrix,* dan *index* kolom berupa *index* iterasi kalimat ke j.

6. Apabila kondisi pada nomor 4 bernilai salah, elemen *matrix* pada baris ke *rowIndex* dan kolom ke i bernilai nol.
7. Diluar loop i, dilakukan penambahan *rowIndex* sebesar satu.

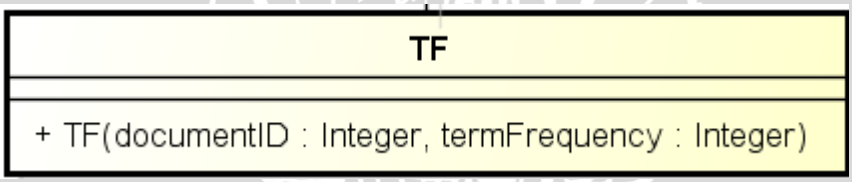
Hasil akhir dari perancangan perhitungan *term frequency—inverted document frequency* adalah objek *matrix* dengan struktur data $matrix$.

Perhitungan dalam *flowchart* pada Gambar 3.11 diimplementasikan didalam sebuah klas yang berperan sebagai *interface*. *Interface* tersebut adalah *TermFrequency*.



Gambar 3.31 Perancangan interface term frequency

Klas yang menjadi implementor *interface* pada Gambar 3.31 disajikan pada Gambar 3.32

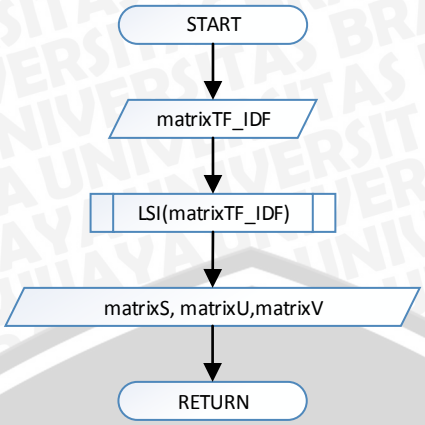


Gambar 3.32 Perancangan klas TF

Berdasarkan Gambar 3.32, konstruktor dari kelas tersebut membutuhkan argumen berupa *index* dokumen, dan nilai awal *term frequency*. *Document id* merupakan identitas dokumen yang akan dihitung, dan *term frequency* merupakan nilai inisialisasi perhitungan *term frequency*.

Setelah perhitungan *term frequency—inverted document frequency* dilakukan, maka langkah selanjutnya adalah melakukan perhitungan *latent semantic indexing*.

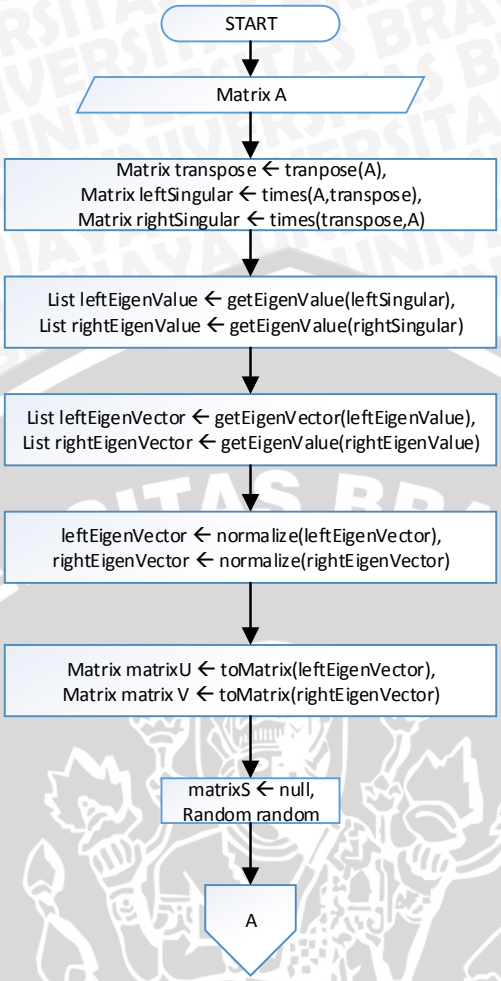
Perancangan *latent semantic indexing* disajikan dalam *flowchart* pada Gambar 3.33

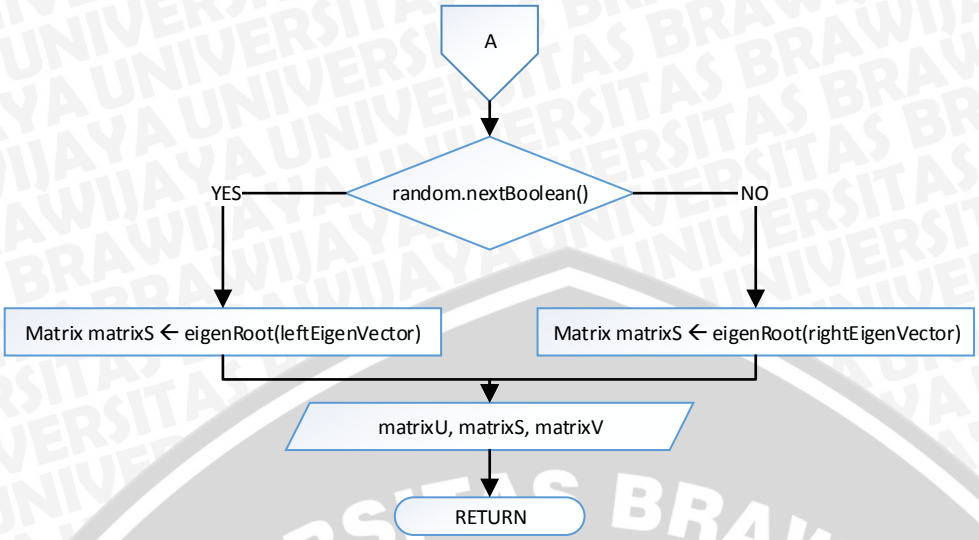


Gambar 3.33 Perancangan latent semantic indexing

Matrix tf_idf menjadi masukan pada tahap ini. Kemudian *matrix* tersebut diolah lebih lanjut pada sub proses *LSI* sehingga dihasilkan *matrixS, matrixU,* dan *matrixV*. Ketiga *matrix* tersebut merupakan hasil akhir sub proses *indexing*.

Perancangan *latent semantic indexing* yang digunakan di penelitian disajikan dalam *flowchart* pada Gambar 3.34 (Ariyanti, 2010).


START
Matrix A
Matrix transpose ← tranpose(A),
Matrix leftSingular ← times(A,transpose),
Matrix rightSingular ← times(transpose,A)
List leftEigenValue ← getEigenValue(leftSingular),
List rightEigenValue ← getEigenValue(rightSingular)
List leftEigenVector ← getEigenVector(leftEigenValue),
List rightEigenVector ← getEigenValue(rightEigenValue)
leftEigenVector ← normalize(leftEigenVector),
rightEigenVector ← normalize(rightEigenVector)
Matrix matrixU ← toMatrix(leftEigenVector),
Matrix matrix V ← toMatrix(rightEigenVector)
matrixS ← null,
Random random
A



Gambar 3.34 Perancangan perhitungan latent semantic indexing

Sub proses latent semantic indexing membutuhkan *matrix* tf_idf. Bobot yang didapatkan dari *matrix* tf_idf diolah lebih lanjut menggunakan *latent semantic indexing* (Paulson, 2010).

Pada langkah kedua dalam flowchart pada Gambar 3.34, *matrix* tf_idf diberikan nama yang lain yaitu *matrix* A untuk mempersingkat nama variabel.

Pada langkah ketiga, dilakukan perkalian *matrix* A dengan transposenya. Hasil dari perkalian tersebut adalah *matrix* singular kiri, dan *matrix* singular kanan.

Pada langkah keempat dilakukan perhitungan untuk mendapatkan nilai eigen dari *matrix* singular kiri dan singular kanan. Pada langkah ini, terjadi perubahan struktur data dari *matrix* menjadi list untuk mempermudah perhitungan.

Pada langkah kelima dilakukan perhitungan untuk mendapatkan vektor eigen dari list eigenValueKiri dan eigenValueKanan. Hasil akhir pada langkah ini adalah list eigenVectorKiri dan eigenVectorKanan.

Pada langkah keenam *matrix*U merupakan hasil konversi dari list ke *matrix* pada list eigenVectorKiri. MatrixV merupakan hasil konversi dari list ke *matrix* pada list eigenVectorKanan.

Pada langkah ketujuh, *matrix*S diinisialisasi dengan null. Hal ini dilakukan karena nilai singular dalam *matrix*S adalah akar pangkat dua dari nilai-nilai eigen *matrix* singularKiri atau *matrix* singular kanan (Ariyanti, 2010). Pada langkah ini juga diinisialisasi bilangan *random.*

Pada langkah kedelapan terdapat kondisi, yaitu jika *random* bernilai *true,* maka inisialisasi *matrixS* diambil dari nilai singular *matrixU.* Namun jika *random* bernilai *false,* maka inisialisasi *matrixS* diambil dari nilai singular *matrixV*.

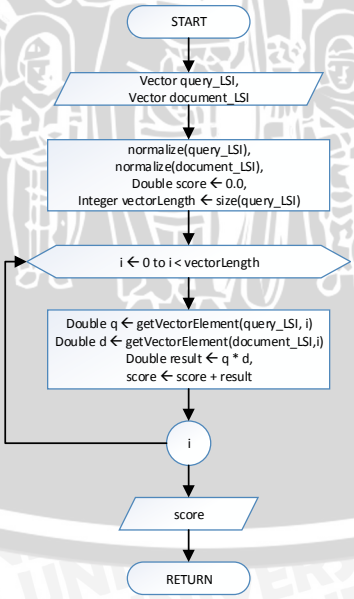
MatrixS, matrixU, dan *matrixV* adalah hasil dari *latent semantic indexing*. Selanjutnya dibutuhkan sebuah vektor kueri untuk menghitung nilai kesamaan. Ketiga *matrix* tersebut digunakan semua dalam perhitungan nilai kesamaan antara vektor kueri terhadap vektor dokumen.

Perancangan implementasi *latent semantic indexing* disajikan dalam Gambar 3.6, dengan metode *getGlobalSVD*. Metode tersebut bertipe *array* $matrix$. Didalam *array* tersebut, terdapat $matrixU$, $matrixS$, dan $matrixV$ sebagai hasil akhir dari *latent semantic indexing*.

Setelah perhitungan *latent semantic indexing* selesai dilakukan, maka langkah selanjutnya adalah melakukan perhitungan dokumen kueri terhadap koleksi dokumen dalam vektor yang direpresentasikan oleh *latent semantic indexing*.

Kromosom dianggap sebagai kueri. Vektor kueri kromosom pada tahap ini masih menggunakan vektor yang didapatkan dari perhitungan *term frequency—inverted document frequency*. Vektor kromosom tersebut perlu dirubah kedalam vektor kueri yang direpresentasikan oleh *latent semantic indexing*, agar dapat dibandingkan dengan koleksi dokumen yang terdapat pada $matrix\ V$. Perubahan vektor kueri dilakukan dengan menggunakan Persamaan 2.3.

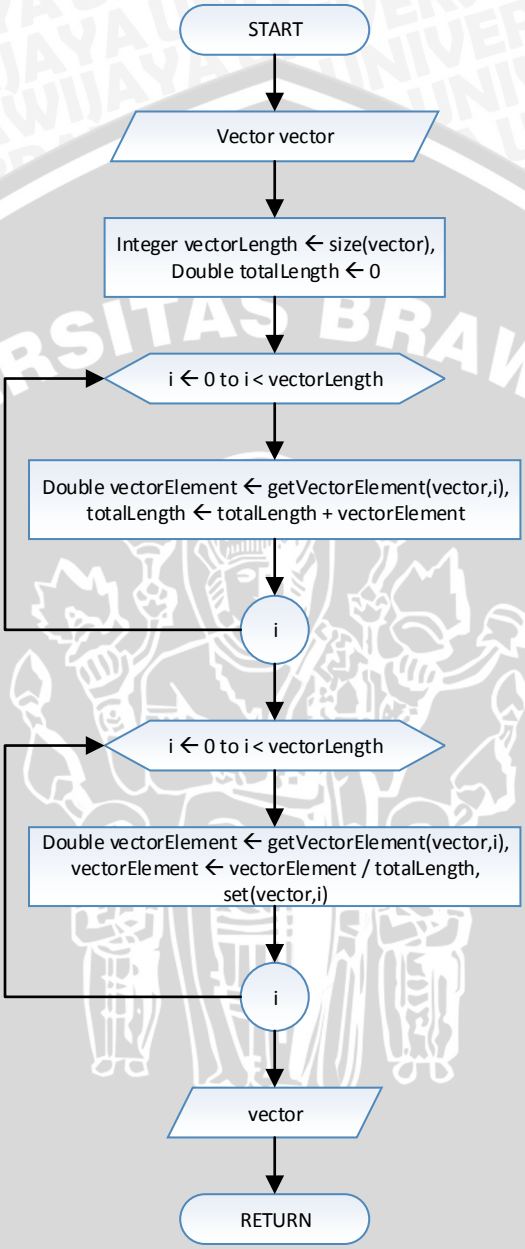
Ketika vektor kueri dalam representasi *latent semantic indexing,* maka vektor kueri dan koleksi vektor dokumen sudah siap untuk dilakukan perbandingan. Perbandingan vektor kueri dan koleksi vektor dokumen menggunakan metode *cosine similarity*. Perancangan perhitungan *cosine similarity* antara kedua vektor tersebut disajikan pada Gambar 3.35



Gambar 3.35 Perancangan perhitungan cosine similarity

Berdasarkan Gambar 3.35, perhitungan *cosine similartiy* membutuhkan vektor kueri dan vektor dokumen menggunakan metode *latent semantic indexing*.

Kedua vektor tersebut dinormalkan. Perancangan perhitungan normalisasi vektor disajikan pada *flowchart* Gambar 3.36



Gambar 3.36 Perancangan perhitungan normalisasi vektor

Berdasarkan Gambar 3.36, normalisasi vektor membutuhkan masukan berupa vektor. Kemudian dilakukan sebuah iterasi terhadap seluruh elemen pada vektor tersebut.

Pada iterasi tersebut nilai dari seluruh elemen vektor tersebut dijumlahkan kedalam sebuah variabel. Setelah iterasi tersebut berakhir, dilakukan iterasi lagi.

Iterasi tersebut dilakukan dengan tujuan untuk membagi nilai seluruh elemen pada vektor tersebut dengan panjang vektor.

Hasil akhir dari proses normalisasi vektor adalah vektor yang sudah dinormalkan.

Setelah vektor kueri dan vektor dokumen tersebut dinormalkan, maka dilakukan iterasi. Pada iterasi tersebut dilakukan :

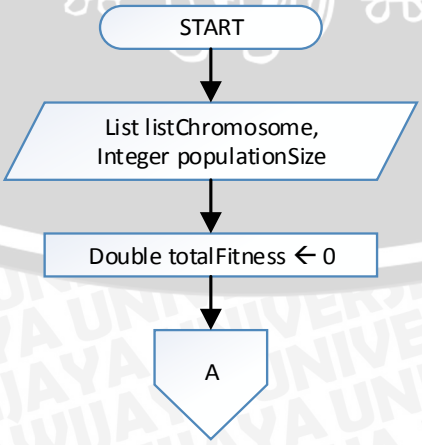
1. Mengambil nilai elemen dari vektor kueri dan menyimpan nilai tersebut didalam satu variabel.
2. Mengambil nilai elemen dari vektor dokumen dan menyimpan nilai tersebut didalam satu variabel.
3. Melakukan perkalian variabel elemen vektor kueri dan variabel vektor dokumen.
4. Melakukan penjumlahan nilai total vektor hasil perkalian.

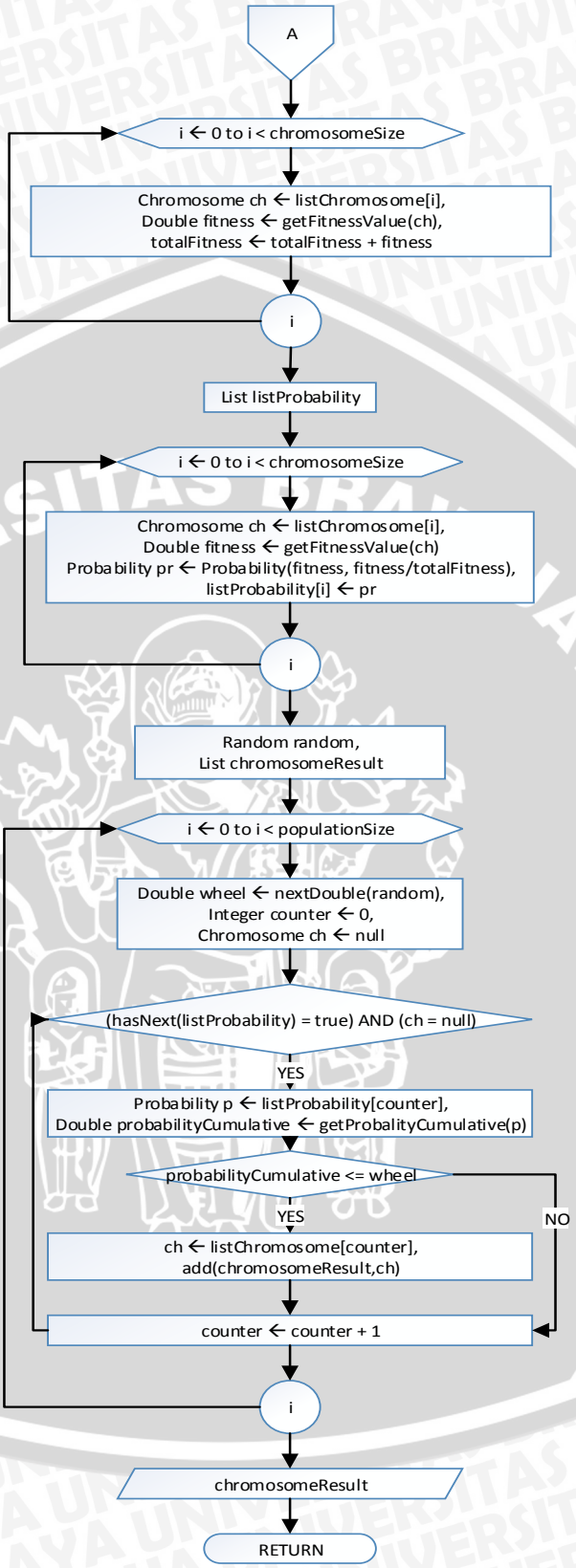
Setelah iterasi tersebut berakhir, maka hasil dari perhitungan *cosine similarity* adalah variabel *score* yang menyimpan hasil dari seluruh perkalian elemen vektor kueri dan vektor dokumen.

Variabel *score* merupakan *fitness* satu kromosom. Iterasi dilakukan sehingga semua kromosom memiliki nilai *fitness*. Hasil akhir dari proses evaluasi kromosom adalah *list* kromosom yang sudah memiliki nilai *fitness* pada setiap kromosom.

3.10.5 Perancangan seleksi

Siklus evolusi selanjutnya adalah melakukan seleksi. Metode seleksi yang digunakan dalam penelitian ini adalah metode *roulette wheel.* Metode seleksi tersebut tidak menjamin kromosom dengan nilai *fitness* yang besar selalu lolos, tetapi setiap kromosom diberikan peluang untuk lolos kepada generasi atau iterasi selanjutnya (Mahmudy, 2013). Perancangan metode seleksi menggunakan *roulette wheel* disajikan pada *flowchart* Gambar 3.37



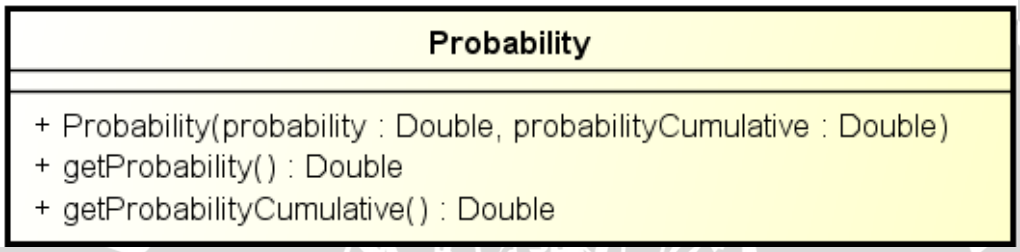


Gambar 3.37 Perancangan seleksi

Masukan pada *flowchart* tersebut adalah *list* kromosom yang berisi seluruh kromosom dari proses sebelumnya, dan parameter algoritma genetik *population size.* Parameter ini digunakan untuk menentukan berapa banyak kromosom yang akan diloloskan pada siklus evolusi selanjutnya.

Seluruh *fitness* dari kromosom dijumlahkan. Penjumlahan ini digunakan untuk menentukan total daerah peluang yang akan diterima oleh setiap kromosom. Hal ini diimplementasikan menggunakan sebuah iterasi terhadap *list* kromosom.

Pada langkah selanjutnya, sebuah struktur data *list* diinisialisasi. *List* tersebut digunakan untuk menampung objek dengan tipe data *Probability*. Tipe data tersebut digunakan untuk menyimpan *probability,* dan *probability cumulative* pada setiap kromosom. *Probability* diimplementasikan didalam sebuah klas pada Gambar 3.38



Gambar 3.38 Perancangan klas probability

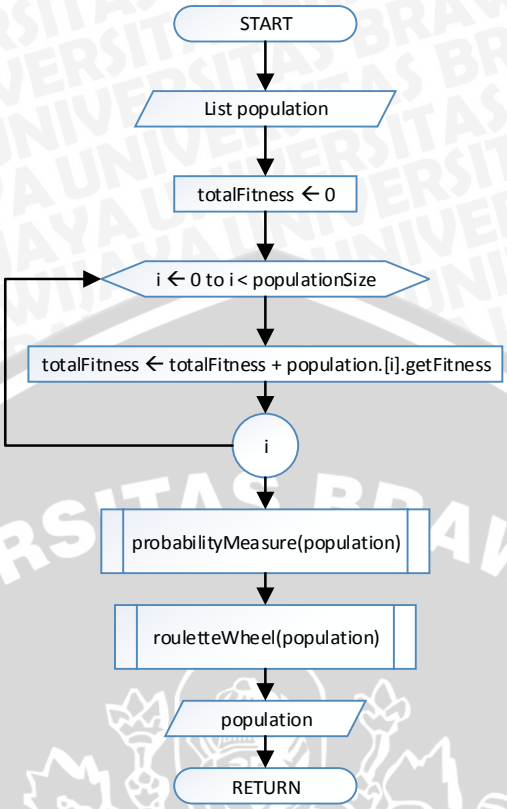
Pada langkah selanjutnya, setiap kromosom telah memiliki *probability,* dan *probability cumulative* yang telah disimpan didalam struktur data *list* yang berisi tipe data *probability*. Sebuah struktur data *list* yang digunakan untuk menampung hasil akhir dari proses seleksi diinisialisasi. Kemudian dibutuhkan sebuah iterasi yang sesuai dengan parameter algoritma genetik *population size*.

Pada iterasi tersebut, proses seleksi dilakukan dengan menggunakan bilangan acak. Batas bawah bilangan acak tersebut sama dengan nol, dan batas atas bilangan acak sama dengan satu. Hasil dari bilangan acak tersebut ditampung didalam sebuah variabel. Bilangan acak tersebut bertindak sebagai wheel. Ketika variabel tersebut sudah berisi nilai tertentu, maka dilakukan iterasi terhadap seluruh elemen struktur data *list* dengan tipe *probability. Probability* yang memiliki *probabibilty cumulative* yang lebih besar atau sama dengan bilangan acak tersebut dimasukkan kedalam *list* hasil akhir.

Hasil akhir proses seleksi adalah sebuah struktur data *list. List* tersebut merupakan kandidat kromosom yang lolos untuk proses evolusi selanjutnya.

Perancangan seleksi menggunakan metode *roulette wheel* disajikan pada Gambar 3.39. *List* populasi menjadi masukan dalam perancangan metode ini. *List* tersebut merupakan hasil dari proses seleksi.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



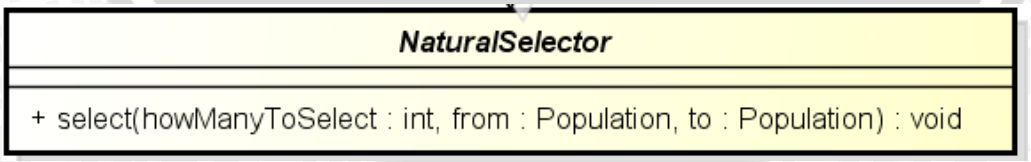
Gambar 3.39 Perancangan seleksi

Langkah selanjutnya adalah mendapatkan total nilai *fitness* dari seluruh kromosom didalam populasi. Hal tersebut dilakukan dengan sebuah iterasi. Dalam iterasi tersebut semua nilai *fitness* dari seluruh kromosom yang ada dijumlahkan.

Langkah berikutnya adalah membagi peluang untuk semua kromosom. Pembagian peluang ini berdasarkan nilai *fitness* setiap kromosom.

Perhitungan peluang untuk setiap kromosom terjadi didalam sub proses *probabilityMeasure*. Kemudian, *roulette wheel* diputar dengan menggunakan bilangan acak. Setiap kromosom yang terpilih dalam setiap putaran dipilih untuk menuju iterasi algoritma genetik selanjutnya.

Proses seleksi diimplementasikan kepada sebuah klas abstrak yang bernama *natural selector.* Perancangan klas abstrak tersebut disajikan pada Gambar 3.40



Gambar 3.40 Perancangan klas abstrak natural selector

Klas abstrak tersebut bertanggung jawab untuk melakukan proses seleksi. Seleksi dilakukan didalam metode *select*. Tugas metode tersebut adalah memilih

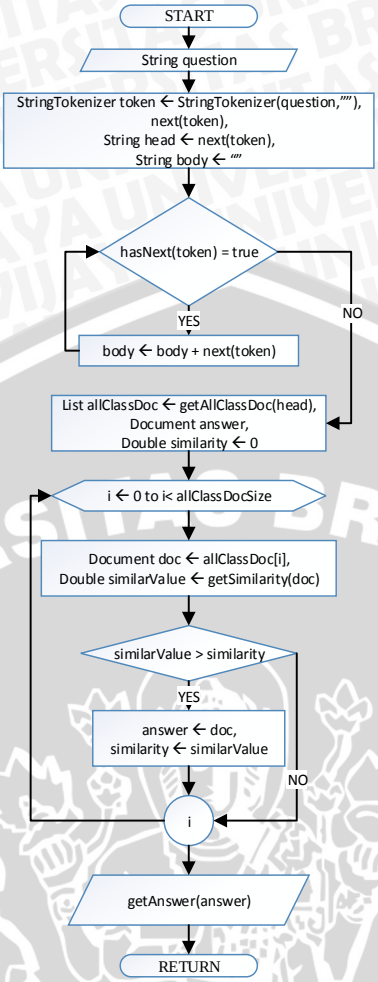
beberapa kromosom dari generasi sebelumnya untuk dibawa kepada generasi selanjutnya.

Parameter *howManyToSelect* mengatur berapa banyak kromosom yang akan diloloskan kepada proses evolusi selanjutnya. Parameter *from* dengan tipe data populasi berisi kromosom dari proses evaluasi. Parameter *to* dengan tipe data populasi berisi kromosom yang dibawa kepada proses evolusi selanjutnya.

3.11 Perancangan document pool

Document pool merupakan tempat yang berisi *term* dalam setiap kelas. Pada tahap ini, setiap *term* tersebut telah melalui proses klasifikasi. Setiap *term* tersebut berada didalam sistem basis data.

Pada setiap *term* terhadap dokumen dilakukan *document preprocessing*. Kemudian *term* tersebut siap untuk melakukan proses *question answering*. Perancangan proses *question answer* disajikan pada Gambar 3.41



Gambar 3.41 Perancangan question answer

Berdasarkan Gambar 3.41, setelah dilakukan proses klasifikasi, maka dapat dilakukan proses *question answering*.

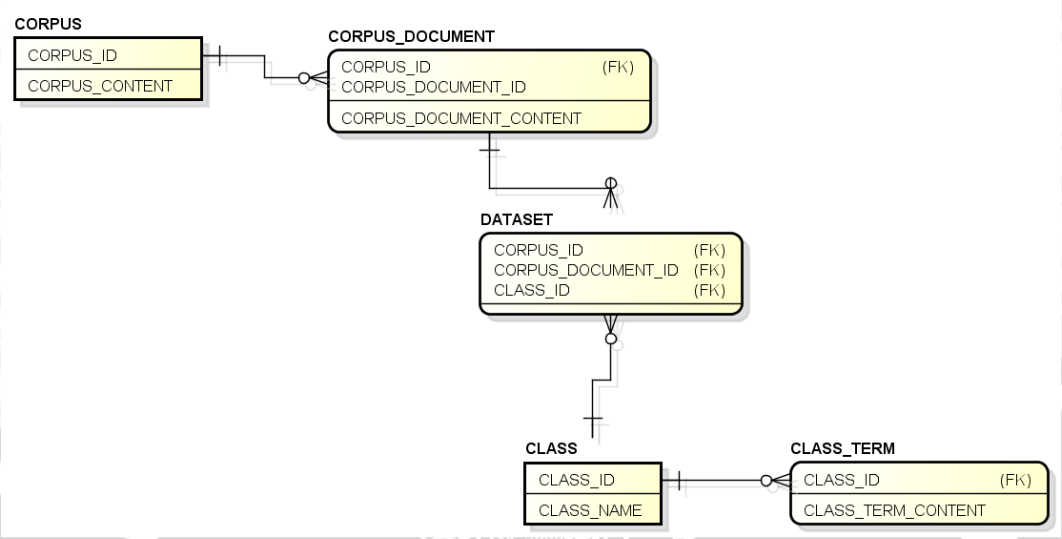
Proses *question answering* merupakan sebuah aktifitas tanya jawab antara pengguna dan sistem dengan menggunakan bahasa alami. Pertanyaan pengguna tersebut harus diawali dengan kata tanya sebagai kata awal.

Kemudian, kata awal tersebut menjadi kata kunci untuk mencari kelas yang bersangkutan. Ketika kelas tersebut ditemukan, maka dilakukan proses similaritas dengan menggunakan *latent semantic indexing* dan *cosine similarity* antara dokumen pada kelas tersebut terhadap pertanyaan pengguna.

Dokumen dengan nilai similaritas yang paling besar dikembalikan sebagai jawaban pengguna.

3.12 Perancangan sistem basis data

Pemodelan sistem basis data yang paling banyak digunakan adalah menggunakan *entity relationship diagram (erd)*. *Entity relationship diagram* digunakan dalam model basis data relasional (Shalahuddin & S, 2013).



Gambar 3.42 Entity relationship diagram

Gambar 3.42, merupakan perancangan sistem basis data didalam *entity relationship diagram* yang digunakan didalam penelitian ini. Terdapat beberapa entitas didalam perancangan sistem basis data tersebut.

Entitas *corpus* bertugas untuk menyimpan data *corpus* kedalam sistem basis data. Entitas tersebut memiliki atribut berupa *CORPUS_ID* yang berfungsi sebagai identitas bagi setiap dokumen *corpus,* dan juga berfungsi sebagai *primary key*. Atribut *CORPUS_CONTENT* berfungsi sebagai penyimpanan data *corpus* dengan tipe data *binary large object (blob)*.

Binary large object digunakan untuk menggantikan parameter pengaturan variabel *varchar*. *Varchar* membutuhkan sebuah argumen untuk parameter yang menyatakan berapa banyak karakter yang akan disimpan. Ukuran *corpus* yang masuk adalah tidak tentu, jika ukuran ini lebih kecil dari batas *varchar,* maka data *corpus* dapat disimpan, namun jika tidak maka data *corpus* tidak disimpan. Dengan menggunakan *binary large object,* maka data akan simpan sesuai ukuran data tersebut.

Entitas *CORPUS_DOCUMENT* berfungsi untuk menyimpan setiap dokumen dari *corpus* tertentu. Satu dokumen sama dengan satu kalimat. Atribut *CORPUS_ID* merupakan *foreign key* dari entitas *CORPUS_DOCUMENT*. Atribut ini digunakan untuk menjaga integritas antar entitas, artinya kalimat dokumen disimpan sesuai dengan *corpus*. Atribut *CORPUS_DOCUMENT_ID* digunakan untuk memberikan identitas kepada setiap dokumen yang berasal dari *corpus* tertentu.

CORPUS_DOCUMENT_ID memiliki tipe data *integer*. Atribut *CORPUS_DOCUMENT_CONTENT* digunakan untuk menyimpan dokumen dari *corpus*. Tipe data dari atribut *CORPUS_DOCUMENT_CONTENT* adalah *varchar*. Digunakan tipe data *varchar* karena yang disimpan adalah satu kalimat.

Entitas *dataset* digunakan untuk menyimpan fakta awal yang digunakan sebagai dasar proses klasifikasi. *CORPUS_ID* merupakan atribut yang digunakan sebagai identitas dokumen *corpus* didalam entitas ini. *CORPUS_DOCUMENT_ID* merupakan identitas kalimat didalam entitas *CORPUS_DOCUMENT*. *CLASS_ID* merupakan atribut identias kelas tertentu didalam entitas *CLASS*. *CORPUS_ID, CORPUS_DOCUMENT,* dan *CLASS_ID* merupakan *foreign key* yang digunakan untuk menjaga integritas data antara satu entitas dengan entitas yang lain.

Entitas *CLASS* berfungsi untuk menyimpan data kelas pada penelitian ini. Atribut *CLASS _ID* digunakan untuk menyimpan identitas kelas tersebut, dan digunakan juga sebagai *primary key*. *CLASS_NAME* merupakan nama dari kelas tersebut.

Entitas *CLASS_TERM* berfungsi untuk menyimpan *term* hasil dari proses klasifikasi terhadap setiap kelas. Atribut *CLASS_ID* merupakan identitas dari entitas *CLASS*. Atribut *CLASS_TERM_CONTENT* merupakan *term*.

3.13 Perancangan pengujian algoritma

Pengujian algoritma digunakan untuk mengetahui seberapa baik hasil proses klasifikasi yang telah dilakukan. Metode pengujian algoritma yang digunakan adalah $f-measure$.

F−measure merupakan salah satu metode pengujian algoritma pada *information retrieval*. Metode tersebut mengambil beberapa dokumen yang dijadikan sebagai dasar pengujian yang disebut sebagai *ground truth,* untuk dibandingkan hasilnya dengan algoritma tertentu (Manning, et al., 2009). Pada penelitian ini, *f−measure* dibandingkan dengan algoritma genetik.

Langkah yang dilakukan dalam pengujian algoritma adalah :

1. Mengumpulkan dokumen yang dianggap relevan.
2. Merepresentasikan satu *term* sebagai satu dokumen. *Term* ini disebut dengan *ground truth term*.
3. Mengurutkan *term* tersebut secara *ascending*.
4. Mengambil satu kromosom terbaik pada akhir evolusi algoritma genetik.
5. Menterjemahkan kromosom tersebut kedalam koleksi *term*.
6. Menghitung jumlah *term* yang cocok kromosom yang telah diterjemahkan tersebut terhadap *ground truth term*.

Satu kromosom terbaik tersebut diambil pada setiap kelas, sehingga setiap kelas memiliki perhitungan *f-measure* sendiri. Implementasi perhitungan *f-measure* disajikan pada Tabel 3.2

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 3.2 Perancangan pengujian algoritma genetik

Kelas	Precision	Recall	F-Measure

Persamaan $F - Measure$ disajikan pada Persamaan 3.4

$$F = \frac{2PR}{P+R} = \frac{2}{\frac{1}{R}+\frac{1}{P}} \tag{3.4}$$

Dimana P menyatakan $precision$, dan R menyatakan $Recall$. $Precision$ adalah kemampuan algoritma untuk mendapatkan beberapa dokumen yang sebagian besar relevan. $Recall$ adalah kemampuan algoritma untuk menemukan semua dokumen yang relevan (Manning, et al., 2009).

Perhitungan $precision$ dan $recall$ menggunakan Tabel 3.3

Tabel 3.3 Perhitungan precision dan recall

$Document/Action$	$Retrieved$	$Not\ Retrieved$
$Relevant$	$Relevant\ Retrieved$	$Relevant\ Rejected$
$Not\ Relevant$	$Irrelevant\ Retrieved$	$Irrelevant\ Rejected$

Persamaan $precision$ dan $recall$ diberikan oleh Persamaan 3.5 dan 3.6 secara berturut turut.

$$Precision = \frac{Relevant\ Retrieved}{Retrieved} \tag{3.5}$$

$$Recall = \frac{Relevant\ Retrieved}{Relevant} \tag{3.6}$$

3.14 Perancangan pengujian question answering system

Pengujian *question answering system* dilakukan terhadap pengguna. Pada pengujian tersebut, pengguna memasukkan kueri berupa pertanyaan menggunakan bahasa alami, kemudian sistem menjawab pertanyaan berdasarkan kueri pengguna.

Pengguna diasumsikan telah mengetahui domain permasalahan dari pertanyaan pada kueri sebelumnya, sehingga pengguna dapat memberikan respon kepada sistem apakah jawaban sistem sudah akurat atau tidak akurat.

Perancangan penilaian pengujian *question answering system* disajikan pada Tabel 3.4

Tabel 3.4 Perancangan pengujian question answering system

No	Pertanyaan	Jawaban	Respon (Benar/Salah)

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Berdasarkan Tabel 3.4, pertanyaan merupakan pertanyaan pengguna. Jawaban merupakan jawaban sistem. Respon berasal dari pengguna apakah jawaban yang dikembalikan oleh sistem akurat atau tidak menurut pengguna yang telah diasumsikan mengetahui domain masalah dari pertanyaan yang diajukan tersebut.

Pengguna yang dimaksud pada pengujian sistem adalah pengguna penguji. Pengguna penguji merupakan seorang pengguna yang menguasai domain permasalahan tertentu, atau dengan kata lain pengguna penguji mengetahui jawaban yang ditanyakan pada sistem.

3.15 Perancangan antarmuka

Perancangan antarmuka bertujuan untuk memberikan kemudahan kepada pengguna untuk melakukan operasi tertentu didalam aplikasi. Perancangan antarmuka merupakan pemodelan dari aplikasi yang akan diakses oleh pengguna.

Perancangan antarmuka disajikan dalam representasi *graphical user interface*. Perintah untuk mengakses operasi tertentu menggunakan *graphical user interface* oleh pengguna. Perintah tersebut diteruskan kepada *application programming interface*. *Application programming interface* merupakan antarmuka yang menjembatani permintaan akses tertentu kepada klas yang mengimplementasikan operasi yang sebenarnya.

Didalam implementasi algoritma genetik pada *question answering system,* perancangan antarmuka memodelkan :

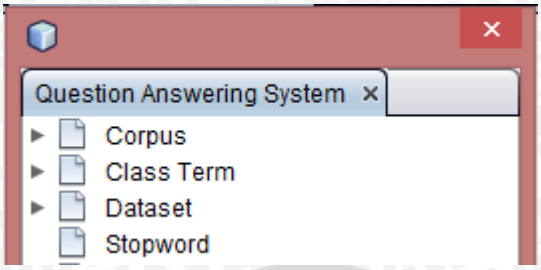
1. Antarmuka sistem basis data.
2. Antarmuka algoritma genetik.
3. Antarmuka pengujian algoritma genetik dan *question answering system*.
4. Antarmuka pengajuan pertanyaan.
5. Antarmuka perolehan jawaban.

Sebagian besar operasi dilakukan dengan *right click* pada *node* sebuah antarmuka tertentu.

3.15.1 Antarmuka sistem basis data

Antarmuka sistem basis data menyajikan pilihan operasi yang didukung didalam sistem basis data pada *question answering system*.

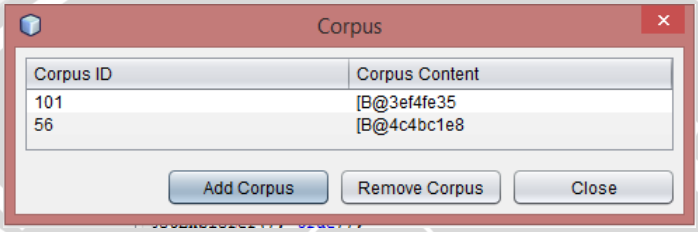
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 3.43 Antarmuka sistem basis data

Operasi yang didukung adalah :

1. Operasi *corpus*



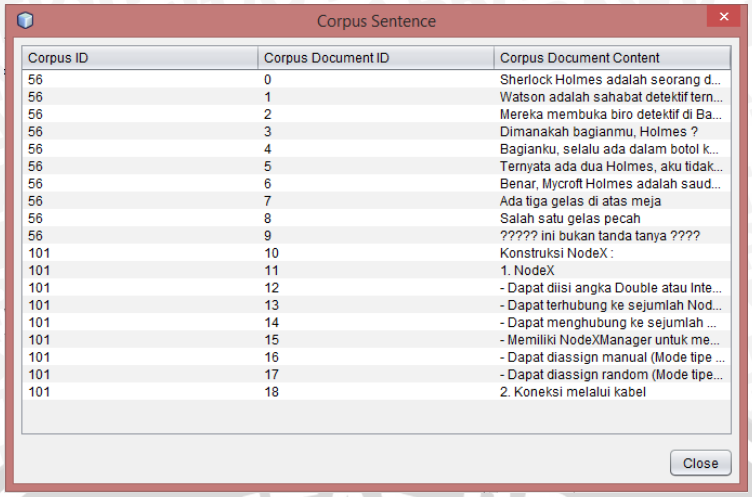
Gambar 3.44 Antarmuka corpus

Pada antarmuka *corpus,* ditampilkan dokumen *corpus* yang sudah ada didalam sistem basis data. *Corpus id* merupakan identitas dokumen *corpus* tersebut, dan *corpus content* merupakan representasi dari *corpus file* dengan tipe data *binary large object* pada sistem basis data.

Add corpus merupakan *menu* untuk menambahkan *corpus* dalam bentuk *file* kedalam sistem basis data. *Remove corpus* merupakan *menu* untuk menghapus *corpus file* tersebut dari sistem basis data, berdasarkan *corpus id*.

2. Operasi *corpus sentence*

Operasi corpus sentence memetakan setiap kalimat dari dokumen corpus. Pemetaan kalimat ini berlangsung ketika file dokumen corpus telah berhasil masuk kedalam sistem basis data.



Gambar 3.45 Antarmuka corpus sentence

Berdasarkan Gambar 3.45, pengguna dapat melihat *corpus id, corpus document id,* dan *corpus document content*.

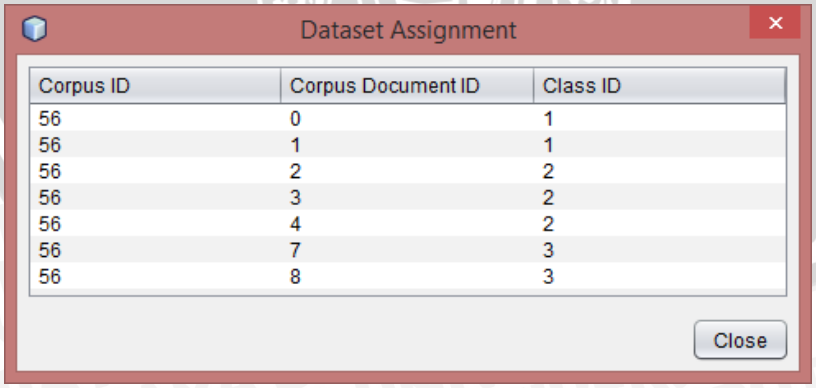
Corpus id merupakan identitas untuk dokumen *corpus*. *Corpus document id* merupakan identitas untuk *hashmap* pada sistem untuk dokumen *dataset*. *Corpus document content* merupakan kalimat dari dokumen *corpus* tersebut.

Karena operasi *document corpus sentence* dilakukan oleh sistem, maka yang dapat pengguna lakukan pada tahap ini adalah melihat hasil *document corpus sentence* kemudian menutup jendela *document corpus sentence*.

3. Operasi dataset

Operasi pada *dataset* digunakan untuk memilih dokumen yang terdaftar pada *corpus,* dan *document corpus sentence* untuk dipilih sebagai *dataset* kelas tertentu sebagai acuan awal algoritma genetika.

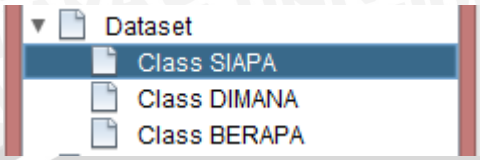
Node dataset memiliki operasi yang menampilkan daftar kalimat yang terdaftar sebagai *dataset* kelas tertentu.



Gambar 3.46 Dataset assignment

Berdasarkan Gambar 3.46, *corpus id* merupakan identitas dokumen *corpus. Corpus document id* merupakan identitas kalimat pada *hashmap*. *Class id* merupakan identitas kelas yang menjadi tujuan dari kalimat tertentu.

Node dataset juga memiliki operasi untuk menampilkan setiap kelas yang ada pada *question answering system*.

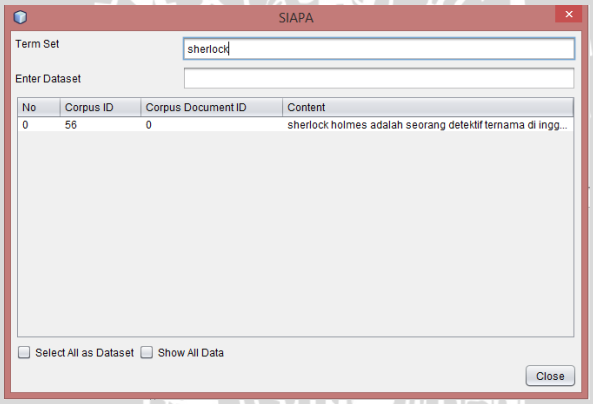


Gambar 3.47 Kelas pada dataset

Kelas yang terdaftar disajikan pada Gambar 3.47. Setiap *node* kelas tersebut bertanggung jawab terhadap *dataset* pada kelas tersebut. Operasi yang dapat dilakukan pada *node* kelas ini adalah *add dataset,* dan *show dataset*.

Add dataset merupakan operasi untuk menambahkan *dataset* yang baru terhadap koleksi *dataset*. Sistem akan mengalami pembaharuan data.

Operasi algoritma genetik yang dilakukan pada langkah selanjutnya berdasarkan perubahan ini.

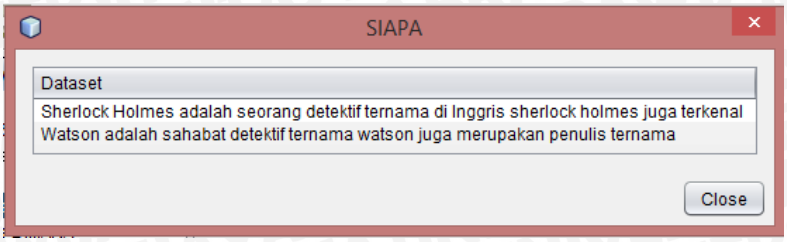


Gambar 3.48 Add dataset kelas SIAPA

Gambar 3.48, mengGambarkan pencarian *dataset* pada kelas "*SIAPA*". *Term set* merupakan *term* yang ingin dicari oleh pengguna pada koleksi dokumen *corpus*. Jika tombol *enter* ditekan maka sistem akan menampilkan semua kalimat yang mengandung *term* tersebut.

Corpus document id yang muncul pada Tabel tersebut diisikan pada *enter dataset*. Jika tombol enter ditekan, maka *corpus document id* yang belum terdapat pada *hashmap* akan ditambahkan kedalam koleksi *document hashmap* pada kelas ini.

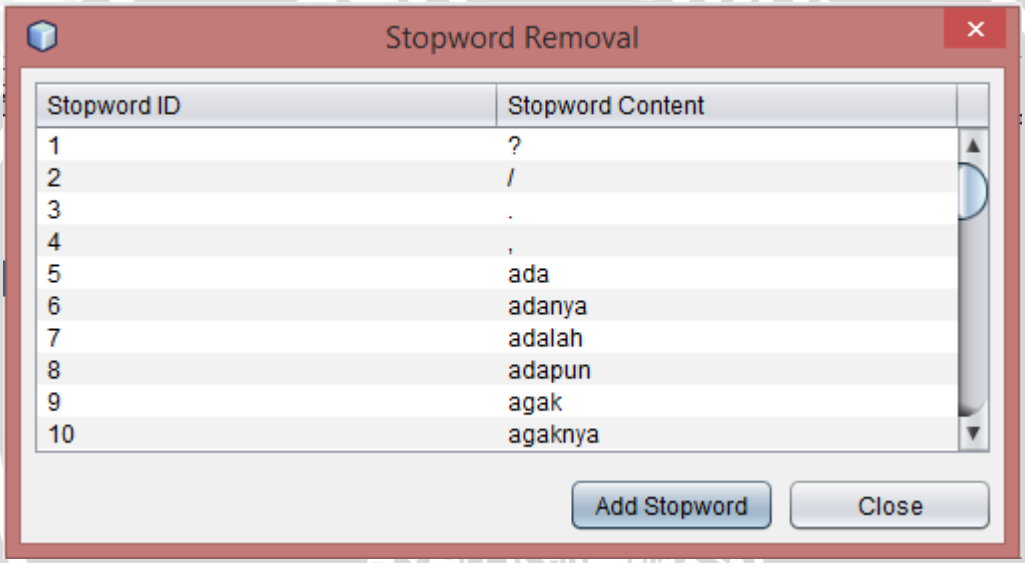
Kalimat dalam *dataset* pada setiap kelas dapat ditampilkan dengan operasi *show dataset*.



Gambar 3.49 Show dataset kelas SIAPA

4. Stopword

Operasi *stopword* mendaftar *term* dan karakter yang tidak diinginkan pada dokumen. *Stopword* digunakan pada *document preprocessing,* oleh karena itu konfigurasi *stopword* dilakukan pada proses sebelum klasifikasi menggunakan algoritma genetik dilakukan.



Gambar 3.50 Stopword removal

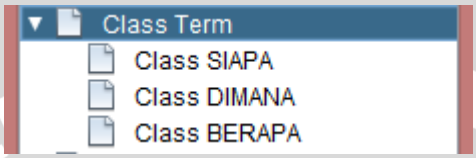
Operasi yang dapat dilakukan pada *stopword removal* adalah *add stopword* dan *remove stopword*. *Add stopword* akan menambahkan *stopword* dari *file* dengan format .txt. *Stopword* yang belum ada pada sistem basis data akan ditambahkan pada proses ini. *Remove stopword* akan menghapus *stopword* pada *hashmap* dalam keadaan *running time* dan *stopword* pada sistem basis data. Operasi ini dilakukan dengan *double click* pada baris Tabel.

Perbedaan cara melakukan antara *add stopword* dan *remove stopword* adalah kompleksitas operasi tersebut. Pada *add stopword* diasumsikan dilakukan sebelum pelatihan algoritma genetik, dan sistem hanya mengakses sistem basis data untuk mendapatkan nilai *stopword*. Sedangkan pada *remove stopword,* sistem memeriksa kepada acuan referensi *hashmap* pada sistem global, dan *hashmap* pada setiap kelas. Dilakukan proses yang bertujuan untuk menghilangkan pasangan *key-value* pada *hashmap,* kemudian dilakukan

penghapusan pada sistem basis data. Perbedaan yang lain adalah *id stopword* pada saat penambahan merupakan *auto generated* pada pengaturan sistem basis data. Sedangkan penghapusan *stopword* mengunakan *id* dari *stopword* tersebut. Jika penghapusan *stopword* menggunakan tombol, maka pengguna harus mengisikan *id stopword* tersebut, namun hal tersebut hilang ketika dilakukan proses secara *double click,* karena sistem yang akan mengisi *id stopword* tersebut.

5. Operasi *class term*

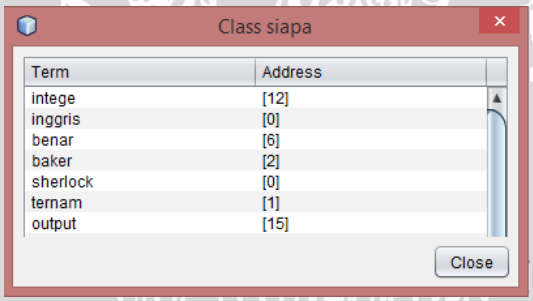
Operasi pada class term menangani proses setelah proses klasifikasi algoritma genetik dilakukan. Class term memiliki node utama dan node kelas yang bersangkutan.



Gambar 3.51 Class term

Node utama pada *class term* memiliku operasi untuk menampilkan *class term* yang telah disimpan pada sistem basis data untuk setiap kelas. Sistem basis data menyimpan objek *hashmap* dari *class term*.

Node untuk setiap kelas pada *class term* memperlihatkan *term* hasil klasifikasi algoritma genetik untuk kelas tersebut, dan alamat dokumen dimana *term* tersebut ada.

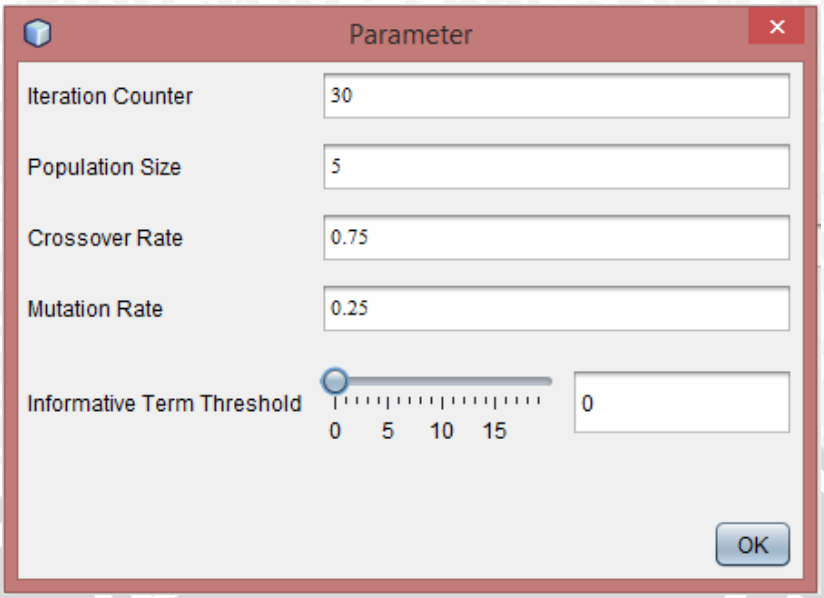


Gambar 3.52 Class term kelas siapa

3.15.2 Antarmuka algoritma genetik

Antarmuka algoritma genetik bertugas untuk mengatur parameter algoritma genetik, menampilkan proses evolusi algoritma genetik, dan menampilkan hasil klasifikasi algoritma genetik

1. Pengaturan Parameter Algoritma Genetik



Gambar 3.53 Parameter algoritma genetik

Berdasarkan Gambar 3.53, digambarkan pengaturan parameter algoritma genetik. Pengaturan algoritma genetik pada tahap ini, akan dilakukan pada proses evolusi algoritma genetik selanjutnya.

2. Menampilkan Proses Evolusi Algoritma Genetik

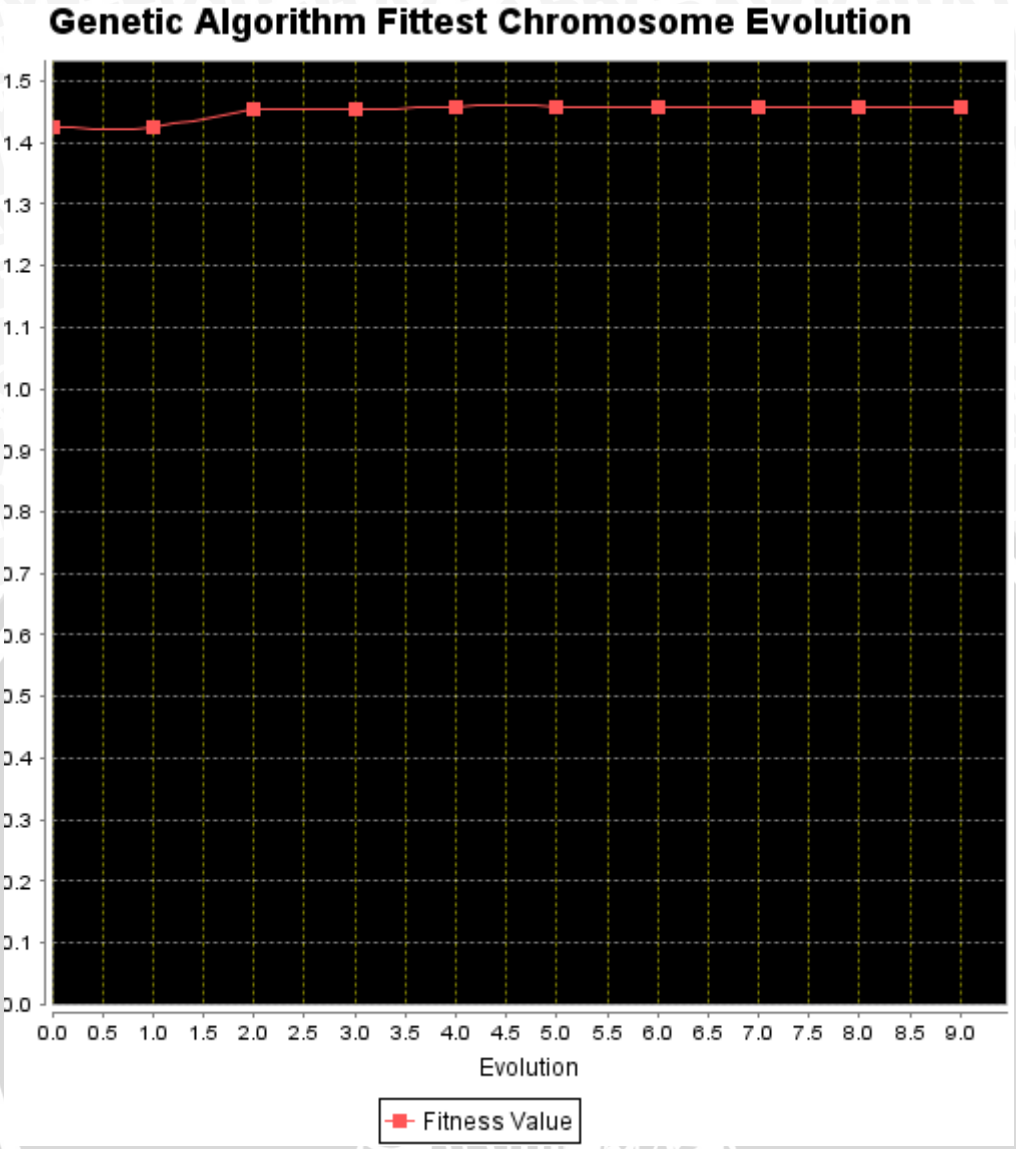
Evolution	Chromosome Number	Alelle
0	0	[0, 33, 3, 5, 38, 39, 40, 10, 42, 13, 15, 47, 48, 50, 19, 20, 21, 22,
0	1	[13, 11, 27, 32, 7, 21, 56, 44, 10, 24, 17, 4, 19, 23, 33, 42, 29, 5
1	0	[3, 25, 5, 39, 33, 0, 10, 47, 55, 50, 40, 15, 27, 48, 42, 22, 19, 28,
1	1	[0, 33, 3, 5, 38, 39, 40, 10, 42, 13, 15, 47, 48, 50, 19, 20, 21, 22,
2	0	[55, 42, 19, 13, 0, 48, 39, 47]
2	1	[13, 39, 47, 33, 28, 27, 19, 40, 56, 50, 5, 22, 38, 54, 25, 3, 42, 1
3	0	[42, 39, 19, 47, 48, 13, 55, 0]
3	1	[13, 55, 48, 47, 0, 39, 19, 42]
4	0	[55, 19, 42, 0, 48, 47, 39]
4	1	[42, 39, 19, 47, 48, 13, 55, 0]
5	0	[0, 47, 42, 55, 48, 19, 39]
5	1	[19, 0, 48, 39, 42, 55, 47]
6	0	[0, 47, 42, 55, 48, 19, 39]
6	1	[19, 0, 48, 39, 42, 55, 47]
7	0	[0, 47, 55, 48, 19, 42, 39]
7	1	[39, 55, 48, 47, 0, 42, 19]
8	0	[19, 48, 55, 42, 0, 39, 47]
8	1	[19, 55, 39, 48, 47, 0, 42]
9	0	[48, 39, 19, 0, 55, 47, 42]
9	1	[19, 48, 55, 47, 0, 42, 39]

Gambar 3.54 Keterangan kromosom algoritma genetik

Berdasarkan Gambar 3.54, diberikan keterangan kromosom algoritma genetik. Informasi yang ditampilkan adalah iterasi evolusi, alel pada kromosom, nomor kromosom, dan nilai *fitness* kromosom.

Kromosom yang diambil adalah kromosom dengan nilai *fitness* terbesar pada setiap proses evolusi. Selain keterangan setiap kromosom, ditampilkan juga grafik *fitness* kromosom terbaik pada setiap evolusi.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

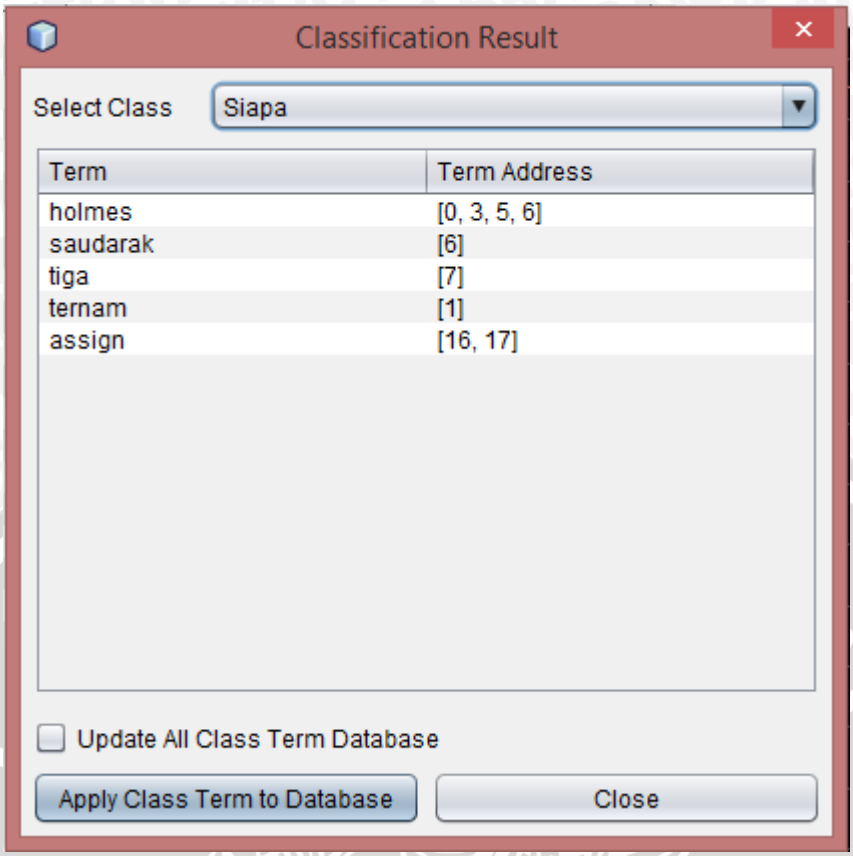


Gambar 3.55 Grafik kromosom terbaik pada setiap evolusi

Berdasarkan Gambar 3.55, sumbu x merupakan iterasi evolusi yang telah dicapai, dan sumbu y merupakan nilai *fitness* yang telah dicapai oleh kromosom. Gambar 3.55 juga menyajikan proses evolusi yang lebih baik daripada penyajian proses evolusi dengan tabel saja.

3. Menampilkan Hasil Klasifikasi Algoritma Genetik

Menu menampilkan hasil klasifikasi algoritma genetik aktif apabila telah dilakukan proses evolusi algoritma genetik. Menu ini akan kembali tidak aktif jika hasil algoritma genetik yang diperoleh selesai ditulis kedalam sistem basis data.



Gambar 3.56 Hasil klasifikasi algoritma genetik kelas siapa

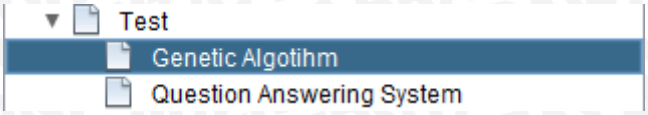
Gambar 3.56 menyajikan hasil klasifikasi menggunakan algoritma genetik untuk kelas *"SIAPA"*. *Term* tersebut disimpan didalam *hashmap* pada setiap kelas. Pertanyaan yang ditanyakan kepada sistem, akan dicari jawaban awalnya menggunakan *term* ini. Pengguna harus menulis *term* ini kedalam sistem basis data untuk memperbarui data hasil klasifikasi pada sistem basis data.

Update all class term database akan menulis hasil klasifikasi algoritma genetik untuk semua kelas kedalam sistem basis data. Pengguna tidak perlu memilih kelas tertentu, mengobservasi hasil klasifikasi tersebut, kemudian menulis perubahannya kedalam sistem basis data, namun hal tersebut akan dilakukan oleh sistem.

3.15.3 Antarmuka pengujian

Antarmuka pengujian yang dirancang meliputi antarmuka pengujian algoritma genetik, dan antarmuka pengujian *question answering system*. Perancangan antarmuka tersebut bertujuan untuk mempermudah penyajian data terhadap pengujian algoritma genetik dan *question answering system*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

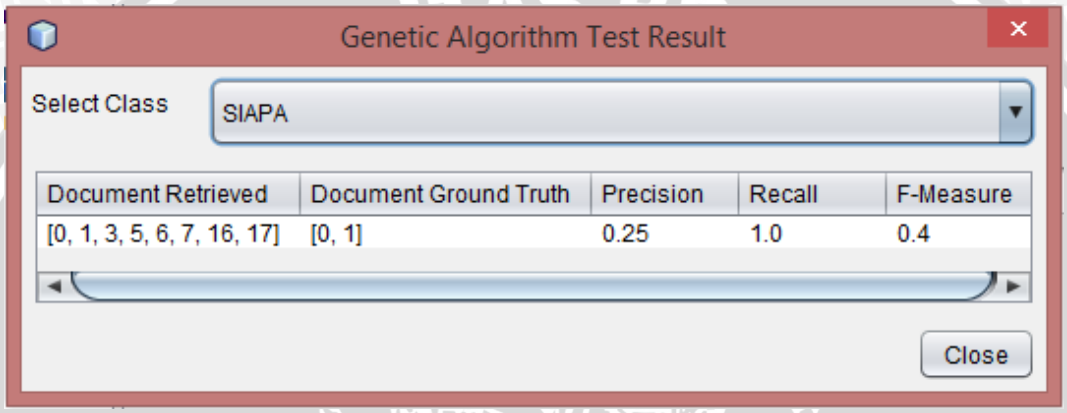


Gambar 3.57 Antarmuka pengujian

Antarmuka pengujian algoritma genetik dan *question answering system* disediakan dibawah *node test*. Akses pada operasi pengujian disediakan melalui klik kanan pada node yang sesuai.

1. Pengujian Algoritma Genetik

Pengujian algoritma genetik dilakukan dengan klik kanan di *node genetic algorithm*. Sebuah *dialog* akan muncul untuk menampilkan hasil pengujian algoritma genetik.



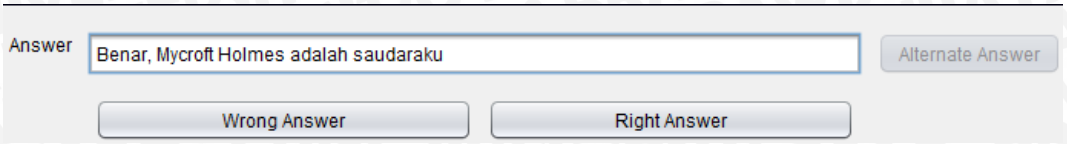
Gambar 3.58 Pengujian algoritma genetik

Pengujian algoritma genetik dilakukan setelah pengguna memilih kelas yang sesuai. Pengujian algoritma genetik menggunakan formula *f-measure*. Formula tersebut membandingkan dokumen yang dihasilkan dari hasil klasifikasi menggunakan algoritma genetik, terhadap dokumen *ground truth,* yaitu dokumen yang telah ditetapkan sebagai acuan dokumen yang relevan.

Berdasarkan Gambar 3.58, disajikan *document retrieved* yaitu dokumen hasil klasifikasi algoritma genetik. *Document ground truth* merupakan dokumen yang dijadikan acuan. Nilai *precision, recall,* dan *f-measure*.

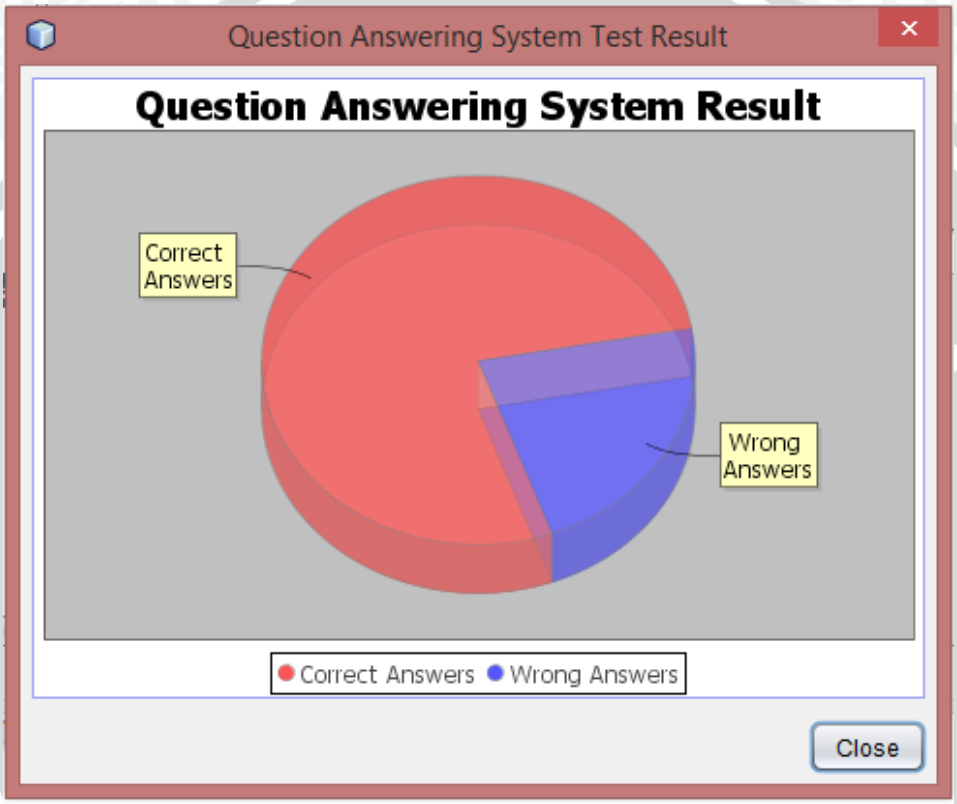
2. Pengujian *Question Answering System*

Pengujian *question answering system* berdasarkan respon dari pengguna. Disediakan dua tombol yang masing-masing mengacu pada tingkat kebenaran jawaban yang diberikan oleh *question answering system*. Diasumsikan pengguna yang menguji sudah mengetahui jawaban yang benar.



Gambar 3.59 Pengujian question answering system

Berdasarkan Gambar 3.59, tombol *wrong answer* merupakan respon yang disediakan untuk jawaban yang salah, dan tombol *right answer* merupakan respon yang disediakan untuk jawaban yang benar. Setiap respon yang yang diberikan oleh pengguna sebagai penguji disimpan didalam sistem basis data.

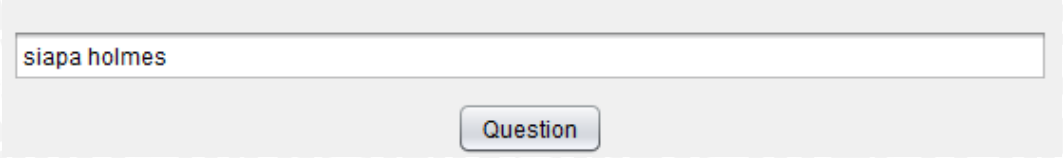


Gambar 3.60 Hasil pengujian question answering system

Berdasarkan Gambar 3.60 disajikan presentase respon pengguna sebagai penguji yang dihitung dari catatan sistem basis data.

3.15.4 Antarmuka pengajuan pertanyaan

Antarmuka pengajuan pertanyaan merupakan tempat bagi pengguna untuk mengajukan pertanyaan kepada sistem. Kalimat pertanyaan pengguna akan dianalisa lebih lanjut oleh sistem sehingga mendapatkan jawaban. Jika jawaban tidak dapat ditemukan oleh sistem, maka akan muncul dialog pada antarmuka pengajuan pertanyaan yang menyatakan bahwa jawaban tidak dapat ditemukan.



Gambar 3.61 Pengajuan pertanyaan

3.15.5 Antarmuka perolehan jawaban

Jawaban yang berhasil ditemukan oleh sistem ditampilkan pada antarmuka perolehan jawaban. Pada antarmuka perolehan jawaban, diberikan repon untuk menampilkan alternatif jawaban, apabila terdapat lebih dari satu kandidat jawaban, dan diberikan respon pengguna untuk menilai apakah jawaban yang diberikan oleh *question answering system* sudah memuaskan atau tidak memuaskan. Antarmuka perolehan jawaban disajikan pada gambar 3.59

3.16 Manualisasi

Manualisasi merupakan contoh dari perhitungan *question answering system*. Data yang dikerjakan pada perhitungan manualisasi merupakan data fiktif. Kromosom algoritma genetik yang digunakan pada perhitungan manualisasi merupakan contoh secara acak.

Iterasi yang digunakan pada manualisasi adalah terbatas. Pada manualisasi tidak dapat disimpulkan performa sistem secara keseluruhan karena adanya batasan data, kromosom, dan jumlah iterasi pada klasifikasi dokumen. Kelas yang digunakan pada contoh perhitungan ini adalah kelas *"SIAPA"*.

Hanya diberikan tiga kromosom pada perhitungan manualisasi. Kromosom tersebut kemudian dicari nilai *fitnessnya*, kemudian dilakukan dilakukan proses evolusi algoritma genetik satu siklus.

Misalkan diberikan koleksi dokumen sebagai berikut :

$d_0 = \text{"}Sherlock\ Holmes \text{ adalah seorang detektif ternama di Inggris"}$

$d_1 = \text{"dr.Watson adalah sahabat Shelock Holmes"}$

$d_2 = \text{"Mereka membuka biro detektif di } Baker\ Street\text{"}$

$d_3 = \text{"Dimanakah bagianmu, Holmes ? "}$

$d_4 = \text{"Bagianku ?, selalu ada dalam botol kecil ini"}$

$d_5 = \text{"Ternyata ada dua } Holmes\text{, aku tidak percaya"}$

$d_6 = \text{"Benar, } Mycroft\ Holmes \text{ adalah saudaraku"}$

Diberikan *dataset* untuk kelas *"SIAPA"* sebagai berikut :

Tabel 3.5 Manualisasi perancangan dataset kelas siapa

Dokumen	Isi Dokumen
d_0	$Sherlock\ Holmes$ adalah seorang detektif ternama di Inggris
d_1	Watson adalah sahabat Sherlock Holmes

Langkah selanjutnya adalah mendaftar semua *term* yang ada pada koleksi dokumen. Pada proses pendaftaran tersebut, tanda baca, tanda tanya, dan penulisan gelar tidak dimasukkan kedalam daftar, hal ini dilakukan untuk mengimplementasikan *stopword removal*. *Term* tersebut kemudian diurutkan secara ascending. Daftar *term* pada dokumen *corpus* disajikan oleh Tabel 3.6 dan Tabel 3.7

Tabel 3.6 Daftar term

No	Term
0	ada
1	adalah
2	aku
3	bagianku
4	bagianmu
5	baker
6	benar
7	biro
8	botol
9	dalam
10	detektif
11	di
12	dimanakah
13	dua
14	holmes
15	inggris
16	ini
17	kecil
18	membuka

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 3.7 Daftar term (lanjutan)

No	Term
20	mycroft
21	percaya
22	sahabat
23	saudaraku
24	selalu
25	seorang
26	sherlock
27	street
28	ternama
29	ternyata
30	tidak
31	watson

Langkah selanjutnya adalah melakukan proses klasifikasi dokumen. Algoritma yang digunakan pada proses klasifikasi dokumen adalah algoritma genetik.

Misalkan diberikan kromosom sebagai berikut :

0	1	2

Kromosom tersebut disebut dengan $kromosom_1$. Panjang $kromosom_1$ sama dengan 3. Panjang setiap kromosom tidak tetap, sehingga panjang kromosom satu dengan panjang kromosom yang lain dapat berubah. *Locus* atau *index* pada kromosom dimulai dari 0.

Term yang terdapat pada $kromosom_1$ adalah *term* yang ada pada Tabel 3.6 dengan *index* $0,1,2$. *Term* tersebut disajikan pada Tabel 3.8

Tabel 3.8 Term pada kromosom

Term Index	Term
0	ada
1	adalah
2	aku

Langkah selanjutnya adalah melakukan perhitungan *fitness* pada kromosom tersebut. Koleksi *term* pada kelas *"SIAPA"* disajikan pada Tabel 3.9

Tabel 3.9 Koleksi term pada kelas siapa

No	Term
0	adalah
1	detektif
2	di
3	holmes
4	inggris
5	sahabat
6	seorang
7	sherlock
8	ternama
9	watson

Setelah didapatkan data seperti pada Tabel 3.9, maka dilakukan perhitungan *term frequency–inverted index frequency* yang melibatkan koleksi *term* pada kromosom dan koleksi *term* pada *dataset* kelas *"SIAPA"*. Perhitungan *term frequency inverted document frequency* Tabel 3.10

Tabel 3.10 Perhitungan term frequency inverted document frequency pada dataset

No	Term	Weighted Term Frequency		Idf	Tf-Idf	
		d_0	d_1		d_0	d_1
0	adalah	1	1	0.00	0.00	0.00
1	detektif	1	0	0.30	0.30	0.00
2	di	1	0	0.30	0.30	0.00
3	holmes	1	1	0.00	0.00	0.00
4	inggris	1	0	0.30	0.30	0.00
5	sahabat	1	0	0.30	0.30	0.00
6	seorang	1	0	0.30	0.30	0.00
7	sherlock	1	1	0.00	0.00	0.00
8	ternama	1	0	0.30	0.30	0.00
9	watson	1	0	0.30	0.30	0.00

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 3.11 Term frequency inverted document frequency pada query

No	Term	TF-IDF
		Query Vector
0	adalah	0.00
1	detektif	0.00
2	di	0.00
3	holmes	0.00
4	inggris	0.00
5	sahabat	0.00
6	seorang	0.00
7	sherlock	0.00
8	ternama	0.00
9	watson	0.00

Berdasarkan Tabel 3.11, elemen vektor kueri bernilai nol semua. Hal tersebut bukan kesalahan perhitungan. Term yang ada pada kueri namun tidak ada pada koleksi term pada dokumen akan bernilai nol. Term yang terdapat pada kedua dokumen akan memiliki nilai inverted document frequency sama dengan nol, term pada kueri yang bernilai sama dengan term ini juga akan bernilai sama dengan nol.

Langkah selanjutnya adalah mengubah vektor pada koleksi dokumen dan kromosom menjadi vektor dalam representasi *latent semantic indexing*. Sehingga diberikan $matrix\ A$ yang merupakan kolom d_1 dan d_2 pada Tabel 3.10

$$A = \begin{bmatrix} 0 & 0 \\ 0.3 & 0 \\ 0.3 & 0 \\ 0 & 0 \\ 0.3 & 0 \\ 0.3 & 0 \\ 0.3 & 0 \\ 0 & 0 \\ 0.3 & 0 \\ 0.3 & 0 \end{bmatrix}$$

$Matrix\ A$ didekomposisi menjadi $matrix\ U, S$ dan V^T.

$$U = \begin{bmatrix} 0.000 & 0.378 \\ 0.378 & 0.857 \\ 0.378 & -0.143 \\ 0.000 & 0.000 \\ 0.378 & -0.143 \\ 0.378 & -0.143 \\ 0.378 & -0.143 \\ 0.00 & 0.000 \\ 0.378 & -0.143 \\ 0.378 & -0.143 \end{bmatrix}$$

$$S = \begin{bmatrix} 0.794 & 0.00 \\ 0.00 & 0.00 \end{bmatrix}$$

$$V^T = \begin{bmatrix} 1.000 & 0.000 \\ 0.000 & 1.000 \end{bmatrix}$$

$matrix\ S$ merupakan $matrix$ singular. $Matrix\ singular$ adalah $matrix$ yang nilai determinannya sama dengan 0. Pada salah satu operasi didalam metode *latent semantic indexing* dilakukan operasi *invers* $matrix\ S$. Operasi *invers* tidak dapat dilakukan jika $matrix$ yang akan dilakukan *invers* sama dengan $matrix\ singular$ (Weisstein, 2016).

Pada tahap ini, metode *latent semantic indexing* tidak dapat dilakukan, sehingga dilakukan *cosine similarity* menggunakan metode *term frequency-inverted document frequency*.

Vektor vektor dokumen tersebut harus dinormalkan dengan menggunakan persamaan 3.7. Berdasarkan Persamaan 3.7, pembilang merupakan vektor, penyebut merupakan panjang vektor.

$$\vec{d} = \frac{\vec{d}}{|\vec{d}|} \tag{3.7}$$

Tabel 3.12 Perhitungan panjang vektor

Dokumen ke	Panjang Vektor
0	0.793725
1	0

Perhitungan nilai *fitness* merupakan penjumlahan nilai *cosine similarity* pada terhadap seluruh vektor dokumen. Vektor dokumen yang baru berdasarkan perhitungan panjang vektor diberikan oleh Tabel 3.13 dan Tabel 3.14

Tabel 3.13 Vektor dokumen ternormalisasi

$dokumen_0$	$dokumen_1$
0	0
0.377	0

Tabel 3.14 Vektor dokumen ternormalisasi (lanjutan)

$dokumen_0$	$dokumen_1$
0.377	0
0	0
0.377	0
0.377	0
0.377	0
0	0
0.377	0
0.377	0

Menggunakan cara yang sama sehingga diperoleh vektor kueri yang disajikan pada Tabel 3.15

Tabel 3.15 Vektor kueri ternormalisasi

$dokumen_{kueri}$
0
0
0
0
0
0
0
0
0
0

Ketika vektor dokumen dan vektor kueri telah ternormalisasi, maka perhitungan *cosine similarity* menggunakan Persamaan 3.8. Tanda perkalian pada persamaan 3.8 merupakan perkalian vektor.

$$\cos(\vec{q}, \vec{d}) = \vec{q} \star \vec{d} \tag{3.8}$$

Perhitungan *cosine similarity* dilakukan sebagai berikut

$\cos(q, d_0) =$ 0

$\cos(q, d_1) =$ 0

---------- +

0

Berdasarkan perhitungan tersebut, maka nilai *fitness* $kromosom_0$ sama dengan 0.Misalkan diberikan $kromosom_2$ sebagai berikut.

2	1	0

Kemudian diberikan $kromosom_3$ sebagai berikut :

1	2	0

$kromosom_2$ dan $kromosom_3$ memiliki *term* yang sama dengan $kromosom_1$, sehingga nilai *fitness* ketiga kromosom tersebut adalah sama. Hal tersebut terjadi karena fungsi *fitness* dinilai berdasarkan vektor *term* terhadap dokumen tertentu. Kromosom dengan *term* yang sama akan memiliki nilai *fitness* yang sama.

Langkah selanjutnya adalah reproduksi. Reproduksi dilakukan dengan *crossover* dan *mutation*. Pada manualisasi ini parameter algoritma genetik diberikan sebagai berikut.

Tabel 3.16 Manualisasi parameter algoritma genetik

No	Parameter	Nilai
1	Jumlah Populasi	3
2	Jumlah Iterasi	1
3	*Crossover Rate*	0.75
4	*Mutation Rate*	0.25

Jumlah kromosom yang baru menggunakan *crossover* sama dengan $jumlahKromosom = 0.75 * 3 = 2.25$, digunakan pembulatan kebawah sehingga $jumlahKromosom = 2$.

Pada *crossover* dipilih dua kromosom. Misalkan kromosom yang terpilih adalah $kromosom_1$ dan $kromosom_2$.

Point of similarity untuk $anak_1$ sama dengan 0, sehingga $anak_1$ memiliki susunan alel sebagai berikut

1	0	2

Point of similarity untuk $anak_2$ sama dengan 1, sehigga $anak_2$ memiliki susunan alel sebagai berikut

2	1	0

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Nilai *fitness* untuk $anak_1$ dan $anak_2$ sama dengan 0, karena *term* yang terkandung sama dengan $kromosom_1$, $kromosom_2$, dan $kromosom_3$. Selanjutnya $anak_1$ dan $anak_2$ disebut dengan $kromosom_4$ dan $kromosom_5$.

Langkah selanjutnya adalah melakukan *mutation*. Metode *mutation* yang digunakan adalah *reciprocal exchange mutation*.

Jumlah anak yang didapatkan dari proses *mutation* sama dengan $jumlahAnak = 3 * 0.25 = 0.75$. Dilakukan pembulatan kebawah sehingga jumlah anak yang didapatkan dari proses *mutation* sama dengan 0. Oleh karena itu, proses *mutation* tidak dilakukan didalam manualisasi ini.

Langkah selanjutnya adalah proses seleksi kromosom. Metode seleksi kromosom menggunakan metode *roulette wheel*. Pada metode ini nilai *fitness* setiap kromosom dijumlahkan. Semua kromosom memiliki nilai *fitness* sama dengan 0, sehingga nilai total *fitness* sama dengan 0.

Total *fitness* dijadikan aturan untuk menentukan luas wilayah probabilitas pada *roulette wheel*. Karena total *fitness* bernilai sama dengan 0, maka luas wilayah probabilitas karena terjadi pembagian dengan penyebut sama dengan 0, dengan demikian pemilihan kromosom untuk proses evolusi selanjutnya dilakukan dengan pengambilan kromosom secara acak sesuai dengan jumlah populasi.

Berdasarkan pemilihan kromosom secara acak, maka kromosom yang lolos kepada proses evolusi selanjutnya disajikan pada Tabel 3.18

Tabel 3.17 Manualisasi kromosom proses evolusi selanjutnya

No	Alel	Keterangan
1	{0,1,2}	Lolos
2	{2,1,0}	Lolos
3	{1,2,0}	Lolos
4	{1,0,2}	Tidak Lolos
5	{0,1,2}	Tidak Lolos

Berdasarkan Tabel 3.18 kromosom yang lolos kepada proses evolusi selanjutnya adalah $kromosom_1$, $kromosom_2$, dan $kromosom_3$.

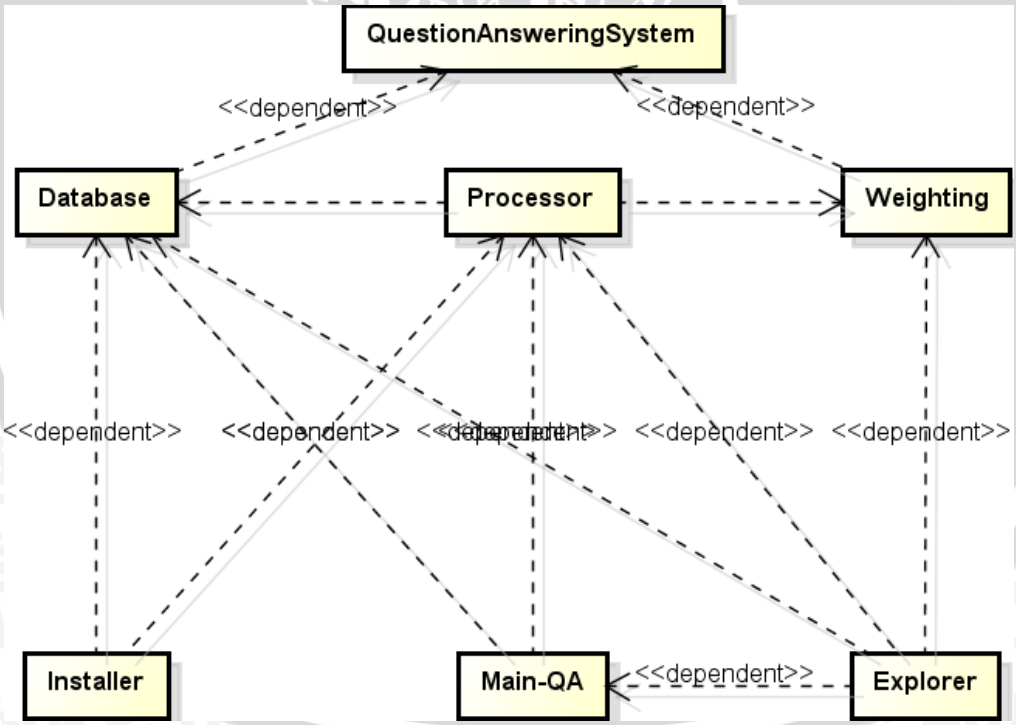
BAB 4 IMPLEMENTASI

Bab IV menjelaskan tentang struktur implementasi algoritma genetik pada *question answering system*, implementasi algoritma, implementasi pengujian algoritma dan *question answering system*, spesifikasi perangkat keras, dan spesifkasi perangkat lunak.

4.1 Struktur implementasi algoritma genetik pada question answering system

Implementasi algoritma genetik pada *question answering system* menggunakan *netbeans platform. Netbeans platform* merupakan *generic framework* untuk *java swing*. *Netbeans platform* menyediakan konfigurasi pengaturan sebuah aplikasi secara umum, seperti pengaturan jendela, pengaturan *file,* dan pengaturan aksi yang ditugaskan kepada elemen tertentu (Bock, 2011).

Netbeans platform menggunakan modul-modul untuk membangun sebuah aplikasi. Modul tersebut memiliki peran yang berbeda. Beberapa modul memiliki *depedency* terhadap modul yang lain. Struktur modul yang digunakan dalam implementasi algoritma genetik pada *question answering system* adalah :

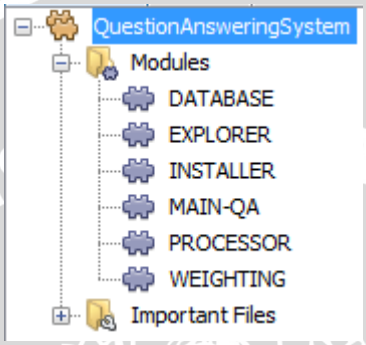


Gambar 4.1 Struktur modul implementasi

4.1.1 Modul question answering system

Berdasarkan Gambar 4.1, modul *question answering system* adalah modul utama. Modul utama bertugas untuk melakukan inisialisasi, mengkomunikasikan informasi yang diperlukan selama sistem berjalan, dan bertindak sesuai kejadian atau *event* yang dipicu oleh pengguna.

Modul yang lain bergantung kepada modul utama, sehingga modul utama harus mengetahui seluruh modul lain yang ada, dan tugas masing-masing modul yang lain tersebut.



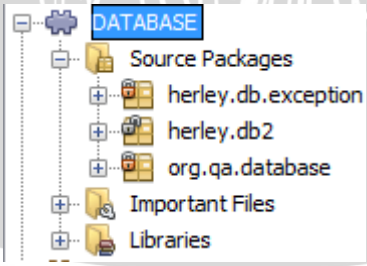
Gambar 4.2 Struktur modul question answering system

Konfigurasi sistem yang meliputi splash screen, dialog icon,module loader, application starter, dan application installer berada pada konfigurasi modul utama.

4.1.2 Modul database

Modul *database* menyediakan fungsi yang bertanggung jawab terhadap sistem basis data. Modul *database* memiliki *depedency* terhadap modul utama *question answering system*.

Modul *database* diimplementasikan kedalam beberapa *package.* Setiap *package* berisi *interface* sebagai jalur komunikasi antar modul, dan klas sebagai implementor *interface* tersebut.



Gambar 4.3 Struktur modul database

Package herley.db.exception bertugas untuk melaporkan eksepsi yang terjadi didalam operasi yang berhubungan dengan sistem basis data. *Package* ini memiliki sebuah klas yaitu *DBException.java*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Klas *DBException.java* merupakan turunan dari kelas *Exception.java* yang merupakan eksepsi standar *java,* sehingga ketika konstruktor kelas *DBException.java* dijalankan maka eksepsi akan dilemparkan. Jika pelemparan eksepsi berada didalam blok *try-catch,* maka *log error* akan ditampilkan dan program tetap berjalan jika tidak dipaksa untuk berakhir.

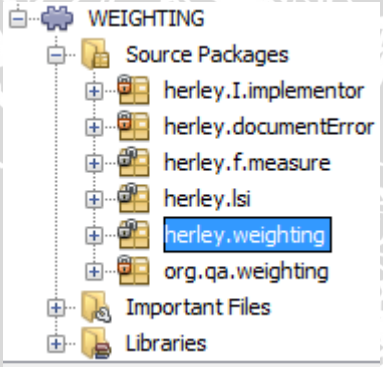
Package herley.db2 berisi *interface* dan kelas yang ditugaskan untuk menangani operasi pada sistem basis data. Operasi yang ditangani adalah operasi pada *corpus, dataset, stopword,* dan *document class term*.

Package org.qa.database berisi identitas dari modul *database*. Tidak ada kode program didalam *package* ini. Identitas didalam *package* ini, digunakan oleh modul utama sebagai sarana inisialisasi dan komunikasi antar modul.

Hanya terdapat satu *instance* dari modul *database* yang digunakan didalam sistem.

4.1.3 Modul weighting

Modul *weighting* bertugas untulk melakukan operasi perhitungan *term frequency-inverted index frequency, latent semantic indexing,* dan *f-measure*. Struktur modul *weighting* disajikan pada Gambar 4.4



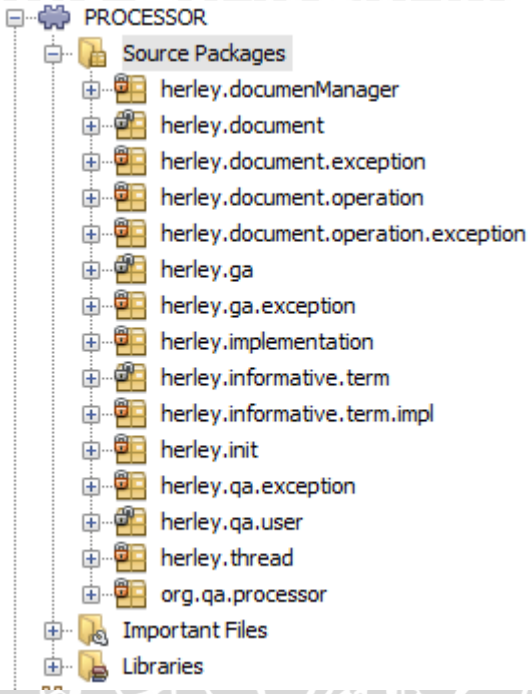
Gambar 4.4 Struktur modul weighting

Package herley.weighting berisi klas yang bertugas untuk melakukan operasi term frequency-inverted index frequency. Package herley.lst berisi klas yang digunakan pada operasi latent semantic indexing. Package herley.f.measure berisi klas untuk melakukan operasi f-measure.

Perhitungan yang digunakan didalam sistem melibatkan lebih dari satu thread, sehingga jika ada klas dari modul lain yang menggunakan modul weighting maka akan dibuatkan instance yang baru untuk menghindari konflik didalam operasi yang melibatkan lebih dari satu thread.

4.1.4 Modul processor

Modul *processor* merupakan modul yang bertugas untuk melakukan implementasi algoritma genetik pada *question answering system*. Struktur modul *processor* disajikan pada Gambar 4.5



Gambar 4.5 Struktur modul processor

Tidak semua *package* diketahui atau dapat diakses oleh modul yang lain. Hal ini dilakukan untuk mendukung enkapsulasi. Enkapsulasi dilakukan untuk menjaga konsistensi operasi sistem pada saat *runtime* karena sistem berjalan pada lebih dari satu *thread*.

Sistem dijalankan pada lebih dari satu *thread* pada perhitungan klasifikasi dokumen menggunakan algoritma genetik. Hal ini dilakukan untuk menghindarkan *graphical user interface* dari *screen freeze*. *Screen freeze* dapat terjadi karena *event dispathced thread* pada *graphical user interface* diblok oleh *thread* utama sedemikian sehingga bagian *thread* utama yang menangani perhitungan klasifikasi menggunakan algoritma genetik telah selesai melaksanakan tugasnya.

Proses klasifikasi menggunakan algoritma genetik merupakan proses utama pada sistem. Proses klasifikasi tersebut dilakukan kepada semua kelas. Proses klasifikasi tersebut juga dilakukan apabila hasil yang dicapai oleh klasifkasi algoritma genetik yang sebelumnya dianggap kurang memuaskan.

Jika proses perhitungan klasifikasi tersebut dilakukan pada *thread* yang lain dari *thread* yang menangani *graphical user interface* atau disebut juga sebagai

event dispatched thread, maka *event dispathced thread* tidak akan diblok oleh *thread* utama sehingga *screen freeze* tidak terjadi. Waktu yang digunakan untuk klasifikasi menjadi relatif lebih cepat, karena proses klasifikasi dilakukan secara bersamaan untuk setiap kelas, sedangkan jika tidak menggunakan *multi thread* maka klasifikasi akan dilakukan secara berurutan.

Dengan menggunakan lebih satu *thread,* maka terdapat pengaturan beberapa sumberdaya yang dapat diakses oleh lebih dari satu *thread* dan terdapat sumberdaya yang hanya boleh diakes oleh satu *thread*. Sumberdaya disini didefinisikan sebagai variabel global yang beriteraksi dengan *instance* klas yang lain. Misalkan modul *database*, satu *instance* modul *database* dapat digunakan oleh berbagai modul pada sistem. Namun, untuk variabel pada sebuah klas tertentu yang memegang peranan sebagai *trigger* ekseskusi klasifikasi menggunakan algoritma genetik hanya dapat diakses dalam satu modul tersebut, dan tidak dapat diakses oleh modul yang lain.

Package yang dapat digunakan oleh *modul* yang lain adalah *herley.document, herley.ga, herley.informative.term,* dan *herley.qa.user*. Akses terhadap *package* tersebut berada didalam satu *thread* dengan *event dispathced thread*. *Thread* pada klas didalam *package tersebut* tidak dipisahkan kedalam *thread* yang lain karena dipandang operasi pada klas didalam *thread* tersebut bukan operasi utama.

Package herley.document berisi informasi tentang setiap kelas yang digunakan dalam proses klasifikasi. Informasi tersebut meliputi *term list, term frequency-inverted document frequency,* dan *document class term*. *Document class term* merupakan *instance* dari objek *hashmap* yang berisi hasil klasifikasi menggunakan algoritma genetik. *Key* pada *hashmap* tersebut adalah *term,* dan *value* adalah alamat dari *term* tersebut.

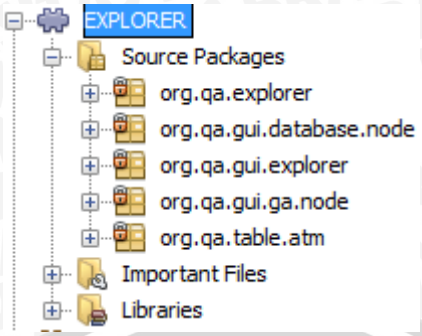
4.1.5 Modul explorer

Modul *explorer* merupakan implementasi dari *node* pada *graphical user interface*. Setiap *node* dapat memiliki satu atau lebih *node* yang lain. Setiap *node* dapat memiliki aksi yang memicu kejadian yang lain.

Modul *explorer* memiliki *depedency* terhadap modul *database, weighting,* dan *processor* karena operasi pada modul *explorer* membutuhkan ketiga modul tersebut. Operasi yang dilakukan oleh modul *explorer* adalah operasi manajemen *corpus, dataset, stopword,* konfigurasi algoritma genetk, pengujian algoritma genetik, dan pengujian *question answering system*.

Untuk melakuka operasi yang diinginkan oleh pengguna, maka pengguna harus berinteraksi secara langsung dengan *node* yang berada pada manajemen modul *explorer*. Struktur modul *explorer* disajikan pada Gambar 4.6

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA



Gambar 4.6 Struktur modul explorer

Berdasarkan Gambar 4.6, modul *explorer* memiliki modul yang semuanya tidak dapat digunakan oleh modul yang lain. Hal ini disebabkan modul *explorer* merupakan modul yang memuat kontrol sistem terhadap pengguna. Semua *package* didalam modul *explorer* bertugas untuk menyajikan data kepada pengguna, atau mengizinkan pengguna untuk melakukan operasi tertentu didalam sistem.

Package org.qa.explorer berisi identitas modul agar dapat dikenali oleh modul yang lain sehigga dapat menggunakan operasi yang didukung oleh modul yang lain.

Package org.qa.database.node berisi klas yang bertugas untuk membentuk *node* untuk disajikan kepada pengguna. Aksi yang didukung oleh *node* tersebut juga dikerjakan pada klas didalam *package* ini.

Package qa.gui.explorer berisi klas yang merupakan *panel* utama tempat semua *node* berada. Semua *node* berada didalam satu manajemen pada *panel* utama ini, sehingga *panel* utama ini disebut *root panel* (Bock, 2011). *Panel* pendukung untuk operasi algoritma genetik juga berada pada *panel* ini.

Package org.gui.ga.node berisi klas yang merupakan *panel* untuk menampilkan parameter algoritma genetik, hasil klasifikasi oleh algoritma genetik, dan *panel* untuk pengujian algoritma genetik, serta *panel* untuk pengujian *question answering system*. Aksi yang berkaitan dengan operasi tersebut juga didefinisikan didalam *package* ini.

Package org.qa.tabel.atm merupakan *package* yang bertanggung jawab untuk menangani manipulasi data pada tabel. Data yang akan ditampilkan pada tabel tertentu, diolah terlebih dahulu oleh *package* ini. Pengolahan tersebut meliputi pemeriksaan tipe data pada Tabel, pemeriksaan ketersediaan data yang akan ditampilkan pada tabel tersebut, dan proses perhitungan baris kolom yang sesuai terhadap data yang masuk.

Package tersebut digunakan oleh *package* didalam satu modul ini, yang memerlukan untuk menampilkan data dalam representasi tabel. Aksi yang dapat dilakukan pada sebuah tabel, misalkan *double click* untuk menghapus data *corpus*

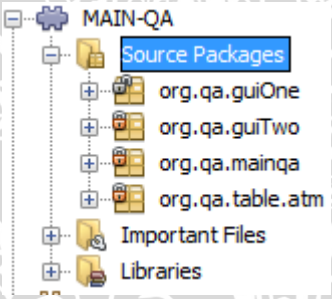
didefinisikan didalam *package* tersebut. *Package* tersebut menjamin bahwa data yang sedang ditampilkan merupakan data yang sedang digunakan.

Package tersebut bekerjasama dengan sistem basis data untuk memastikan integritas data. Apabila data tersebut berada didalam manajemen sistem basis data, maka pembaharuan tampilan Tabel terjadi apabila telah dilakukan pembaharuan kepada sistem basis data. Namun, apabila data tersebut berada didalam manajemen sistem, data yang ditampilkan adalah data yang benar benar digunakan oleh sistem pada saat itu. Konfigurasi *listener* pada tabel menjamin bahwa pembaharuan data yang digunakan oleh sistem dilakukan dalam penyajian data tersebut pada representasi data dengan menggunakan tabel.

4.1.6 Modul main-qa

Modul *main qa* merupakan modul yang menangani *question answering*. Pertanyaan pengguna dan jawaban yang diberikan oleh sistem berada pada modul ini.

Modul *main-qa* memiliki *depedency* terhadap modul *database* dan modul *processor*. Struktur modul *main-qa* disajikan oleh Gambar 4.7



Gambar 4.7 Struktur modul main-qa

Berdasarkan Gambar 4.7, semua *package* didalam modul *main-qa* bersifat *private* kecuali *package org.qa.guiOne*. *Package org.qa.guiOne* tidak *private* karena *window system* yang didefinisikan didalam *package org.qa.guiOne* dibutuhkan oleh operasi modul *processor*. Modul *processor* membutuhkan *window system* pada *package org.qa.guiOne* karena modul *processor* harus menampilkan jawaban apakah pertanyaan yang diajukan oleh pengguna tersedia jawaban. Jika tidak tersedia jawaban, maka ditampilkan sebuah dialog yang mengindikasikan bahwa jawaban atas pertanyaan pengguna tidak tersedia, namun jika jawaban yang diminta tersedia, maka jendela jawaban pada sistem akan diinisialisasi sebagai *instance* yang baru, kemudian jawaban tersebut akan dikembalikan oleh *processor* kepada jendela jawaban.

Jika jumlah jawaban yang tersedia sama dengan satu, maka jawaban tersebut akan langsung ditampilkan, namun jika jawaban yang tersedia berjumlah lebih dari satu, maka jawaban alternatif tersebut ditampilkan didalam sebuah Tabel dimana pengguna dapat memilih jawaban tersebut sebagai jawaban terbaik, dan

diharapkan pengguna diharapkan memberikan respon yang baik juga terhadap *question answering system*.

Package org.qa.mainqa berisi deskripsi modul ini agar dapat digunakan dan menggunakan modul yang lain. Modul utama mengenali modul ini dari deskripsi tersebut.

Package org.qa.tabel.atm bertugas untuk manajemen data yang disajikan kepada pengguna dalam representasi tabel. Aksi yang dapat dilakukan pada tabel tersebut juga didefinisikan pada *package* tersebut.

4.1.7 Modul installer

Modul *installer* merupakan modul yang dieksekusi pertama kali ketika semua modul telah diidentifikasi oleh modul utama (Bock, 2011). Modul *installer* juga berfungsi untuk membuat *application installer* pada sistem operasi *windows* dengan syarat telah terdapat *java runtime environment*.

Struktur modul *installer* disajikan pada Gambar 4.8

Gambar 4.8 Struktur modul installer

Modul *installer* hanya terdiri dari satu *package,* yaitu *package org.qa.installerl.* Didalam *package* tersebut terdapat klas yang melakukan inisialisasi kepada sistem basis data. Inisialisasi tersebut meliputi *host, port, database name, username,* dan *password*. Jika koneksi yang dilakukan kepada sistem basis data berhasil, maka *instance* pada modul *database* dapat digunakan oleh semua modul tanpa perlu melakukan inisialisasi lagi.

Dengan menggunakan koneksi kepada sistem basis data, modul *installer* dapat melakukan *document preprocessing* yang meliputi *stopword removal,* dan *stemming*. Pada saat operasi ini terjadi, semua *term* yang didapatkan dari dokumen *corpus* dan *dataset* telah didaftarkan kepada *hashmap* agar dapat digunakan dalam perhitungan *fitness* oleh algoritma genetik.

4.2 Implementasi algoritma

Implementasi algoritma yang dijelaskan adalah algoritma *term frequency-inverted document frequency* dan algoritma genetik. Kedua algoritma tersebut digunakan didalam perhitungan *fitness* kromosom pada proses klasifikasi menggunakan algoritma genetik.

4.2.1 Term Frequency-Inverted Document Frequency

pada modul *weighting* pada *package herley.weighting,* pada klas *W.java*. Klas *W.java* merupakan realisasi dari *interface Weighting.java*. *Interface* tersebut digunakan oleh modul lain yang ingin mengakses operasi *tf-idf*.

Perhitungan *term frequency-inverted document frequency* dilakukan pada dengan dua tahap. Tahap pertama untuk menghitung *inverted index,* dan tahap kedua merepresentasikan *inverted index* dalam struktur data *matrix*.

Pada perhitungan tahap pertama, dilakukan perhitungan dengan hasil akhir *term frequency-document frequency*. Tahap ini mencatat setiap *term* yang bersifat unik dari koleksi dokumen *corpus*. Kemudian mencatat kejadian *term* pada setiap dokumen, dan kemuculan *term* tersebut kepada seluruh koleksi dokumen.

Kode sumber untuk operasi tahap pertama, disajikan pada Kode Sumber 4.1

```
1  @Override
2  public Map getInvertedIndex() {
3  try {
4    if (this.documents == null) {
5     throw
6  new DocumentException("Koleksi dokumen corpus belum diset");
7     } else {
8       if (this.query == null) {
9           throw new DocumentException("Query belum diset");
10        }
11 /**
12 * Pengosongan hashmap, instance yang lain menggunakan dalam
13 * keadaan kosong.
14 */
15         if (!this.hashmap.isEmpty()) {
16             this.hashmap.clear();
17         }
18 /**
19 * Main Document Loop -Do Not Modify-
20 */
21 for (int i = 0; i < this.documents.size(); i++) {
22   String dokumen = this.documents.get(i);
23     final StringTokenizer token
24             = new StringTokenizer(dokumen.toLowerCase());
25     while (token.hasMoreTokens()) {
26        String term = token.nextToken();
```

```
27
28  /**
29  * Point 1.
30  */
31    if (this.hashmap.containsKey(term)) {
32      List valueList = this.hashmap.get(term);
33
34        while (valueList == null) {
35          valueList = this.hashmap.get(term);
36        }
37
38       int dfCount = (int) valueList.get(0);
39       boolean modified = false;
40
41  /**
42  * Point 2
43  */
44  for (int j = 1; j < valueList.size(); j++) {
45   TermFrequency termFrequency
46     = (TermFrequency) valueList.get(j);
47   int documentID = termFrequency.getDocumentID();
48   if (documentID == i) {
49     int tf = termFrequency.getTermFrequency();
50     tf++;
51     termFrequency.setTermFrequency(tf);
52     modified = true;
53     break;
54     }
55  }
56
57     if (modified == false) {
58      dfCount++;
59      valueList.set(0, dfCount);
60      int countTF = 1;
61  int tfID = i;
62  final TF termFrequency = new TF(tfID, countTF);
63  valueList.add(termFrequency);
64  }
65
```

```
66   } else {
67     int countDF = 1;
68     int countTF = 1;
69     int tfID = i;
70  TF tf = new TF(tfID, countTF);
71  final List valueList = new ArrayList();
72  valueList.add(countDF);
73  valueList.add(tf);
74  this.hashmap.put(term, valueList);
75      }
76    }
77
78  }
79
80  Map<String, List> map = new TreeMap(this.hashmap);
81  /**
82  * Return Point.
83  */
84   this.theMap = map;
85   return map;
86  }
87   } catch (Exception e) {
88     e.printStackTrace();
89   }
90   return null;
91  }
```

Kode Sumber 4.1 Perhitungan term frequency inverted document frequency

Kode sumber pada Kode Sumber 4.1 dapat dilakukan jika ada koleksi dokumen, dan sebuah kueri. Pada perhitungan *fitness* menggunakan algoritma genetik, koleksi dokumen merupakan *dataset,* dan kueri merupakan kromosom pada algoritma genetik tersebut.

Kode program untuk operasi pada tahap kedua yang bertujuan untuk merubah *inverted index* dalam struktur data *matrix* disajikan pada Kode Sumber 4.2

```
1  @Override
2  public Matrix getTermFrequencyInvertedIndexFrequency() {
3  try {
4    if ((this.hashmap == null) || (this.hashmap.isEmpty())) {
5        this.getInvertedIndex();
```

```
6  } else {
7
8  /**
9  * Jika Belum kosong, maka bersihkan dulu.
10 */
11 if (!this.weightedTermFrequency.isEmpty()) {
12  this.weightedTermFrequency.clear();
13 }
14
15 final Matrix matrix = DenseMatrix.Factory
16   .zeros(this.hashmap.size(), this.documents.size());
17 int rowMatrix = 0;
18
19 final TreeMap<String, List> map = new TreeMap();
20  map.putAll(this.hashmap);
21
22 Iterator<Map.Entry<String, List>> iterSatu = map
23  .entrySet().iterator();
24 while (iterSatu.hasNext()) {
25 Map.Entry<String, List> ne = iterSatu.next();
26 String key = ne.getKey();
27 final List<Double> wtfList = new ArrayList();
28
29 List valueList = ne.getValue();
30 int df = (int) valueList.get(0);
31 /**
32 * POINT 1.
33 */
34 for (int i = 0; i < this.documents.size(); i++) {
35
36 boolean found = false;
37 for (int j = 1; j < valueList.size(); j++) {
38  TermFrequency tf = (TermFrequency) valueList.get(j);
39  int documentID = tf.getDocumentID();
40
41  if (documentID == i) {
42   double wtf = weightedTerm(tf.getTermFrequency());
43   wtfList.add(wtf);
44   double idf = invertedDocumentFrequency(df);
```

```
45    double tfidf = wtf * idf;
46    matrix.setAsDouble(tfidf, rowMatrix, i);
47    found = true;
48    break;
49    }
50   }
51
52  /**
53  * POINT 2.
54  */
55  if (found == false) {
56   matrix.setAsDouble(0, rowMatrix, i);
57   }
58  }
59  rowMatrix++;
60
61  /**
62  * Weighted-TF Point.
63  */
64  this.weightedTermFrequency.put(key, wtfList);
65   }
66
67  /**
68  * RETURN POINT.
69  */
70   this.theTFIDF = matrix;
71   return matrix;
72   }
73   } catch (Exception e) {
74   e.printStackTrace();
75   }
76   return null;
77  }
```

Kode Sumber 4.2 Representasi matrix term frequency inverted document frequency

Kode program yang disajikan pada Kode Sumber 4.2 mengubah *inverted index* yang sebelumnya berada pada representasi *hashmap* menjadi representasi *matrix*. Perubahan struktur data tersebut dilakukan karena pengolahan data pada *latent semantic indexing* melibatkan struktur data *matrix*.

Dilakukan operasi perkalian *term frequency* dan *inverted document frequency* pada tahap kedua. Baris pada *matrix* merupakan nilai *term frequency-inverted index frequency* terhadap *term* tertentu, kolom pada *matrix* merupakan nilai *term frequency-inverted index frequency* terhadap dokumen tertentu.

4.2.2 Algoritma genetik

Implementasi algoritma genetik berlangsung pada modul *processor*, pada *package herley.ga*. Implementasi algoritma genetik pada *package* ini merupakan realisasi dari *interface GeneticAlgorithm.java*. Digunakan sebuah *interface* sebagai sarana komunikasi antar modul. Untuk setiap kelas yang melakukan proses klasifikasi menggunakan algoritma genetik, dibuatkan sebuah *instance* yang berbeda dari kelas satu terhadap kelas yang lain. Hal tersebut dilakukan karena proses pelatihan algoritma genetik terjadi pada *thread* yang berbeda untuk setiap klas, sehingga dapat dilakukan klasifikasi menggunakan algoritma genetik secara bersama.

Kode sumber implementasi algoritma genetik pada klas *GA.java* disajikan pada Kode Sumber 4.3

```
1   @Override
2   public void run() {
3   /**
4   * Configuration Reset.
5   */
6   final Configuration configuration = new DefaultConfiguration("Senju
    Hashirama", "The First Hokage");
7   Configuration.reset("Senju Hashirama");
8
9   /**
10  * Selection.
11  */
12  final Roulette roulette
13          = new Roulette(configuration);
14  configuration.addNaturalSelector(roulette, false);
15
16  /**
17  * Crossover.
18  */
19  final Crossover crossover
20          = new Crossover(this.crossoverRate);
21  configuration.addGeneticOperator(crossover);
22
```

```
23  /**
24  * Mutation.
25  */
26  final Mutation mutation
27          = new Mutation(this.mutationRate);
28  configuration.addGeneticOperator(mutation);
29
30  /**
31  * Chromosome Representation.
32  */
33  final TermChromosome termo
34          = new TermChromosome(configuration);
35
36  /**
37  * Genotype Configuration.
38  */
39  final Gene[] genes = new Gene[1];
40  genes[0] = termo;
41  final Chromosome chromosome
42          = new Chromosome(configuration, genes);
43
44  /**
45  * Set perhitungan document preprocessing pada DocumentClass.
46  */
47  this.documentClass.doTermFrequencyInvertedIndexCalculation();
48
49  /**
50  * Set population size.
51  */
52  configuration.setPopulationSize(this.populationSize);
53
54  /**
55  * Set sample chromosome.
56  */
57  configuration.setSampleChromosome(chromosome);
58
59  /**
60  * Set fitness function.
61  */
```

```
62  configuration.setFitnessFunction(new TermFitness());
63
    final IChromosome [] arrayIChromosome = new
64  IChromosome[this.populationSize];
65  for(int i=0; i<arrayIChromosome.length; i++){
66      final Gene[] geneArray = new Gene[1];
67      TermChromosome termChromosome = new TermChromosome(configuration);
68      geneArray[0] = termChromosome;
69      while(termChromosome.getChromosome().isEmpty()){
70          /** Random Generator Term Chromosome - Code Omitted **/
71
            final RandomGeneratorTermChromosome adam = new
72  RandomGeneratorTermChromosome();
73          termChromosome = new TermChromosome(configuration);
74          termChromosome.setToRandomValue(adam);
75          geneArray[0] = termChromosome;
76      }
77      final Chromosome chrom = new Chromosome(configuration,geneArray);
78      arrayIChromosome[i] = chrom;
79  }
80
    final Genotype genotype = new
81  Genotype(configuration,arrayIChromosome);
82
83  /** Swing Worker - Code Omitted **/
84
    final SwingWorkerExecutor apocal = new SwingWorkerExecutor(genotype,
85  this.iterationCount, this.documentClass);
86  apocal.execute();
87
88  final TheGenotype tg = Lookup.getDefault().lookup(TheGenotype.class);
89  tg.setGenotype(genotype);
90  }
```

Kode Sumber 4.3 Implementasi algoritma genetik

Kode sumber 4.3 melakukan tugas sebagai berikut :

1. Memeriksa apakah parameter algoritma genetik sudah diatur dan apakah kelas yang akan diproses sudah melakukan pendaftaran.
2. Membuat satu konfigurasi algoritma genetik pusat untuk satu proses algoritma genetik, dari tahap pengenalan kromosom hingga seluruh proses evolusi algoritma genetik telah selesai dilakukan.

3. Melakukan inisialisasi klas yang bertugas untuk melakukan *crossover, mutation, selection,* dan perhitungan *fitness* pada algoritma genetik. Kemudian klas tersebut didaftarkan kepada konfigurasi pusat.
4. Melakukan inisialisasi terhadap klas yang bertugas untuk menangani *thread*. Klas tersebut merupakan turunan dari klas *SwingWorker.java* yang mampu memisahkan *event dispatched thread* dari *thread* utama. Pada klas ini, berlangsung proses evolusi algoritma genetik sesuai dengan pengaturan konfigurasi algoritma genetik pusat. Penampilan proses perhitungan algoritma genetik pada Tabel dan grafik juga diatur didalam klas ini.
5. Melakukan inisialisasi terhadap klas yang bertugas untuk mengatur tampilan data pada Tabel. Pengaturan meliputi perhitungan panjang kolom Tabel, dan pengaturan panjang kolom Tabel minimal sama dengan jumlah panjang data yang ditampilkan didalam Tabel.
6. Melakukan penyimpanan hasil klasifikasi menggunakan algoritma genetik didalam sebuah struktur data *hashmap* dengan *key* sama dengan *term,* dan *value* sama dengan alamat *term* tersebut pada dokumen.
7. Melakukan penyimpanan *hashmap* kedalam sistem basis data. Sistem basis data tidak menyimpan tipe data primitif dari elemen pada *hashmap,* melainkan menerima *hashmap* itu sendiri.

Setiap terdapat perintah untuk melakukan klasifikasi menggunakan algoritma genetik, maka metode *run* yang disajikan pada Tabel akan dijalankan kembali.

4.3 Implementasi pengujian

Implementasi pengujian menyajikan implementasi kode program pengujian yang meliputi pengujian algoritma genetik, dan pengujian *question answer*.

4.3.1 Pengujian algoritma genetik

Pengujian algoritma genetik menggunakan formula *f-measure*. Dalam perhitungan *f-measure* dibutuhkan :

1. *Ground Truth Document.*
2. Perhitungan *Recall.*
3. Perhitungan *Precision.*

Ground truth document merupakan koleksi dokumen yang dinyatakan benar terhadap kelas tertentu. Representasi satu dokumen sama dengan satu kalimat, sehingga kalimat dianggap sebagai *ground truth document*. *Ground truth document* ditentukan oleh pengguna penguji dengan asumsi bahwa pengguna penguji memiliki kecakapan pada bidang yang menjadi pokok bahasan pada *question answering system*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Implementasi *ground truth document* adalah koleksi dokumen pada *dataset*. *Dataset* disimpan didalam sistem basis data. Atribut *CORPUS_DOCUMENT_ID* pada Tabel *DATASET* merupakan *key* untuk *hashmap*. *CORPUS_DOCUMENT_ID* bernilai tidak sama untuk setiap kalimat yang ditugaskan untuk menjadi *dataset* bagi kelas tertentu.

Sebuah kromosom memiliki gen yang memiliki alel dengan identitas nomor *term* pada *hashmap*. *Term* tersebut berada pada suatu dokumen. Setiap dokumen memiliki identitas, sehingga identitas dokumen yang didapatkan dari satu kromosom algoritma genetik dicocokkan dengan identitas *ground truth document*.

Perhitungan pengujian algoritma genetik berlangsung pada modul *explorer,* pada *package org.qa.gui.explorer*. Kode sumber pengujian algoritma genetik menggunakan formula *f-measure* disajikan pada Kode Sumber 4.4

```
1   try {
2   /**
3   * Retrieved Document.
4   */
5   String query = "SELECT CLASS_TERM_CONTENT FROM QA.CLASS_TERM WHERE
    CLASS_ID =  ?";
6   PreparedStatement ps =
    super.getDatabaseConnection().prepareStatement(query);
7    ps.setInt(1, classID);
8    ResultSet rs = ps.executeQuery();
9    rs.next();
10  final byte[] byteArray = rs.getBytes("CLASS_TERM_CONTENT");
11  final FSTConfiguration fst =
    FSTConfiguration.createDefaultConfiguration();
12  final Object ao = fst.asObject(byteArray);
13  final HashMap<String, List> documentRetrieved = (HashMap) ao;
14  final List<Integer> retrievedDocument = new ArrayList() {
15
16  @Override
17  public boolean contains(Object o) {
18   final Integer inter = (Integer) o;
19   final Object[] ta = this.toArray();
20   int counter = 0;
21   final Integer hasil = null;
22   while ((counter < ta.length) && (hasil == null)) {
23    if (ta[counter] == inter) {
```

```
24      return true;
25    }
26      counter++;
27    }
28      return false;
29   }
30 };
31
32 Iterator<Map.Entry<String, List>> iterSatu = documentRetrieved.entrySet().iterator();
33 while (iterSatu.hasNext()) {
34   Map.Entry<String, List> ne = iterSatu.next();
35   final List<Integer> list = ne.getValue();
36   if (list != null) {
37    for (Integer inter : list) {
38     if (!retrievedDocument.contains(inter)) {
39       retrievedDocument.add(inter);
40         }
41        }
42       }
43     }
44
45 /**
46 * Ground Truth Document.
47 */
48 query = "SELECT CORPUS_DOCUMENT_ID FROM QA.DATASET WHERE CLASS_ID = ?";
49 ps  = super.getDatabaseConnection().prepareStatement(query);
50 ps.setInt(1, classID);
51 rs = ps.executeQuery();
52
53 final List<Integer> groundTruthDocument = new ArrayList();
54 while(rs.next()){
55  groundTruthDocument.add(rs.getInt("CORPUS_DOCUMENT_ID"));
56 }
57
58 Collections.sort(retrievedDocument, new Comparator<Integer>() {
```

```
59
60  @Override
61  public int compare(Integer t, Integer t1) {
62    return t.compareTo(t1);
63   }
64 });
65
66 Collections.sort(groundTruthDocument, new Comparator<Integer>() {
67
68 @Override
69 public int compare(Integer t, Integer t1) {
70  return t.compareTo(t1);
71  }
72 });
73
74 final List kolom = new ArrayList();
75 kolom.add(retrievedDocument.toString());
76 kolom.add(groundTruthDocument.toString());
77 final FMeasure fm = Lookup.getDefault().lookup(FMeasure.class);
78 fm.setDocumentList(retrievedDocument, groundTruthDocument);
79 kolom.add(fm.getPrecision());
80 kolom.add(fm.getRecall());
81 kolom.add(fm.getFMeasure());
82 super.pusat.add(kolom);
83
84 /**
85 * Update JTabel Dataset.
86 */
87 super.fireTabelStructureChanged();
88 super.fireTabelDataChanged();
89  } catch (Exception e) {
90  e.printStackTrace();
91 }
```

Kode Sumber 4.4 Implementasi pengujian algoritma genetik

Perhitungan pengujian algoritma genetik dilakukan terlebih dahulu, kemudian pembaharuan pada Tabel dilakukan.

Perhitungan pengujian algoritma genetik dimulai ketika pengguna memilih salah satu kelas pada sistem. Terdapat *listener* pada pemilihan salah satu kelas tersebut.

Jika kelas tersebut sudah terpilih, maka langkah selanjutnya adalah memanggil hasil klasifikasi algoritma genetik yang untuk kelas tersebut yang disimpan didalam sistem basis data. Kemudian, memanggil *dataset* dari sistem basis data untuk dijadikan sebagai *dataset* terhadap kelas tersebut.

Alamat dokumen pada kelas tersebut dibandingkan dengan *dataset* yang sesuai dengan kelas tersebut. Dilakukan perhitungan *precision* dan *recall,* kemudian dilakukan perhitungan *fmeasure*.

4.3.2 Pengujian question answering system

Pengujian *question answering system* dilakukan untuk mengetahui kepuasan pengguna terhadap hasil *question answering system*. Pengujian *question answering system* dilakukan dengan memberikan akses respon kepada pengguna penguji berupa pilihan jawaban yang dikembalikan oleh *question answering system* benar atau salah.

Implementasi pengujian *question answering system* berlangsung didalam modul *explorer, package org.qa.gui.explorer,* klas *QATestJPanel.java*. Klas tersebut merupakan turunan dari klas *JPanel.java*. Kode program implementasi berada didalam metode yang dieksekusi pertama kali oleh konstruktor klas *QATestJPanel.java*

Kode sumber pengujian *question answering system* disajikan pada Kode Sumber 4.5

```
1   try {
2   String query = "SELECT COUNT(*) AS HITUNG FROM QA.RESPONSE";
3    PreparedStatement
4
5       ps = Lookup.getDefault().lookup(Database.class
6
7   ).getConnection().prepareStatement(query);
8   ResultSet rs = ps.executeQuery();
9
10  rs.next ();
11  final int hitung = rs.getInt("HITUNG");
12
    query  = "SELECT COUNT(RESPONSE) AS HITUNG FROM QA.RESPONSE
13  WHERE RESPONSE = 1";
```

```
ps  =
Lookup.getDefault().lookup(Database.class).getConnection().p
repareStatement(query);
rs  = ps.executeQuery();

rs.next ();
final int jumlahYa = rs.getInt("HITUNG");
final int jumlahTidak = hitung - jumlahYa;

final DefaultPieDataset dataset = new DefaultPieDataset();

dataset.setValue (

 "Correct Answers", jumlahYa);
 dataset.setValue (
 "Wrong Answers", jumlahTidak);

final JFreeChart chart = ChartFactory.createPieChart3D(
"Question Answering System Result", // chart title
 dataset, // data
           true, // include legend
           true,
           False
   );
   final PiePlot3D plot = (PiePlot3D) chart.getPlot();

   plot.setStartAngle (

   290);
   plot.setDirection (Rotation.CLOCKWISE);

   plot.setForegroundAlpha (

   0.5f);
   plot.setNoDataMessage (
   "No data to display");

```

```
49
50 //   this.chartPanel  = new ChartPanel(chart);
51 chartPanel = new ChartPanel(chart);
52 this.chartPanel.setPreferredSize (
53 this.getPreferredSize());
54
55
56 this.chartPanel.setVisible (
57    true);
   chartPanel.setBorder(javax.swing.BorderFactory.createMatteBo
58 rder(1, 1, 1, 1, new java.awt.Color(153, 153, 255)));
59 } catch (Exception e) {
60 e.printStackTrace();}
```

Kode Sumber 4.5 Question answering system

Sebelum perhitungan dilakukan, pada sistem basis data harus terdapat respon pengguna penguji. Jika belum terdapat data tersebut, maka perhitungan tidak dilakukan.

Koneksi sistem basis data didapatkan terlebih dahulu. Koneksi sistem basis data tidak dalam keadaan *null,* karena sistem sudah *login* kepada sistem basis data ketika sistem diinisalisasi pada modul *installer*.

Kueri dikirimkan kepada sistem basis data untuk menghitung jumlah satu respon tertentu. Jumlah respon yang lain dapat diketahui berdasarkan jumlah keseluruhan respon pada sistem basis data. Respon yang satu dibandingkan terhadap respon yang lain dengan menggunakan presentase. Hasil presentase tersebut disajikan pada representasi grafik dengan dua kategori, yaitu kategori *right answer* dan kategori *wrong answer*.

4.4 Spesifikasi perangkat lunak

Implementasi Algoritma Genetik pada *Question Answering System* diimplementasikan pada perangkat lunak sebagai berikut :

1. Sistem Operasi *Windows 8.1 64-bit.*
2. *DB2 v.10.5* sebagai *Database Management System.*
3. Bahasa Pemograman *Java.*
4. *Netbeans IDE 8.0.2* sebagai *code editor*.
5. *Netbeans Platform 8.0.2* sebagai *application plaftorm*.
6. *Java Development Kit 8.0*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

4.5 Spesifikasi perangkat keras

Implementasi Algoritma Genetik pada *Question Answering System* diimplementasikan pada perangkat keras sebagai berikut :

1. *Processor Intel(R)Core(TM)i7-4710HQ @2.50 GHz~3.40GHz*
2. *Random Acces Memory 3.88 Gb*
3. *Hard Disk Drive 1 TB*
4. *Monitor Resolution 1366* x *76*

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

BAB 5 PENGUJIAN DAN ANALISIS

Bab V menjelaskan tentang pengujian dan analisis implementasi algoritma genetik pada *question answering system*. Pengujian dan analisis yang dilakukan terhadap implementasi algoritma genetik pada *question answering system* meliputi pengujian parameter algoritma genetik, pengujian algoritma genetik, dan pengujian *question answering system*.

5.1 Pengujian parameter algoritma genetik

Parameter algoritma genetik yang digunakan melalui beberapa pengujian untuk mengetahui hasil klasifikasi yang bagus dengan waktu yang terbatas. Parameter algoritma genetik yang diuji adalah jumlah iterasi, jumlah populasi, *crossover rate, mutation rate, informative term threshold,* dan *chromosome length*.

5.1.1 Penegujian jumlah iterasi

Pengujian jumlah iterasi dilakukan untuk mengetahui waktu yang dibutuhkan untuk melakukan pelatihan algoritma genetik. Pada pengujian jumlah iterasi, diujikan jumlah iterasi sama dengan $1, 5, 10$, dan 15. Parameter algoritma genetik yang lain disajikan pada Tabel 5.1

Tabel 5.1 Parameter algoritma genetik untuk pengujian iterasi

Parameter	Nilai
Jumlah Populasi	10
Crossover Rate	0.75
Mutation Rate	0.25
Dimension Rate	2
Informative Term Threshold	10
Chromosome Length	10

Hasil pengujian jumlah iterasi untuk lama waktu evolusi menggunakan parameter algoritma genetik berdasarkan Tabel 5.1, disajikan pada Tabel 5.2. Jumlah iterasi yang terlalu sedikit menyebabkan algoritma genetik tidak ada kesempatan untuk berevolusi, namun jumlah iterasi yang terlalu banyak mengakibatkan waktu yang dibutuhkan untuk pelatihan algoritma genetik menjadi terlalu lama, sehingga tidak dapat diterima.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.2 Hasil pengujian jumlah iterasi

Jumlah Iterasi	Waktu Proses Evolusi (Dalam Menit)
1	15
5	60
10	150
15	290
20	430

Berdasarkan Tabel 5.2, jika jumlah iterasi ditambah maka waktu proses evolusi untuk algoritma genetik juga bertambah. Faktor lain yang berpengaruh terhadap lamanya waktu proses evolusi pada algoritma genetik adalah jumlah kalimat, dan jumlah *term*.

Jumlah kalimat sama dengan 387 kalimat, dan jumlah *term* sama dengan 1620 *term*. Berdasarkan jumlah kalimat dan jumlah *term,* maka jumlah iterasi yang dipilih untuk pengujian algoritma genetik dan pengujian *question answering system* sama dengan 10. Hal ini dilakukan karena jika algoritma genetik menggunakan iterasi yang telalu sedikit untuk cakupan data yang besar, maka algoritma genetik belum mampu untuk menemukan solusi yang baik, namun jika jumlah iterasi yang dipilih terlalu besar, maka waktu evolusi algoritma genetik tidak dapat diterima karena toleransi waktu evolusi algoritma genetik yang digunakan pada penelitian ini kurang dari 200 menit.

5.1.2 Pengujian jumlah populasi

Pengujian jumlah populasi dilakukan untuk mengetahui berapa jumlah *term* hasil klasifkasi algoritma genetik. Jumlah populasi yang terlalu sedikit, akan berakibat algoritma genetik tidak dapat menemukan solusi optimum, namun jumlah populasi yang terlalu banyak akan menyebabkan algoritma genetik memiliki banyak solusi untuk ditawarkan. Solusi tersebut adalah *term,* sehingga *term* hasil klasifikasi tidak akurat (Gotshall & Rylander, 2002).

Parameter pengujian yang dilakukan diberikan pada Tabel 5.3 dan Tabel 5.4

Tabel 5.3 Parameter algoritma genetik untuk jumlah populasi

Parameter	Nilai
Jumlah Iterasi	5
Crossover Rate	0.75
Mutation Rate	0.25
Dimension Rate	2

Informative Term Threshold	10

Tabel 5.4 Parameter algoritma genetik untuk jumlah populasi (lanjutan)

Parameter	Nilai
Chromosome Length	10

Hasil pengujian jumlah populasi disajikan pada Tabel 5.5

Tabel 5.5 Hasil pengujian jumlah populasi

Jumlah Populasi	Jumlah Term
5	50
10	100

Term yang didapatkan berdasarkan Tabel 5.4, dimasukkan kedalam sebuah *hashmap* untuk mencari *term* yang unik. Berdasarkan parameter algoritma genetik pada Tabel 5.3, pada setiap evolusi algoritma genetik akan mengarahkan kromosom hasil akhir pada setiap evolusi untuk mendekati 10 *term* pertama sesuai dengan parameter *informative term threshold,* sehingga jumlah minimal *term* unik pada *hashmap* sama dengan 10. *Hashmap* tersebut dijadikan sebagai acuan utama dalam *question answering*.

Semakin banyak *term* unik yang mendekati *term* yang didefinisikan pada *dataset*, maka *term* hasil klasifikasi tersebut semakin baik. Namun jika persis sama dengan *dataset* atau tidak mendekati sama sekali, maka klasifikasi tersebut tidak baik. Jumlah *term* sama dengan 100 lebih baik melakukan *cover* terhadap 387 dokumen dan 1620 *term* yang unik daripada jumlah *term* sama dengan 50.

5.1.3 Pengujian crossover rate dan mutation rate

Crossover rate dan *mutation rate* telah diuji pada penelitian yang dilakukan oleh (Mitchell, 1998) dan contoh persoalan algoritma genetik yang diuji oleh (Mahmudy, 2013). *Crossover* dan *mutation rate* yang digunakan disajikan pada Tabel 5.6

Tabel 5.6 Hasil pengujian crossover rate dan mutation rate

Crossver Rate	Mutation Rate
0.75	0.25

5.1.4 Pengujian dimension rate

Dimension rate merupakan ukuran dimensi pada *matrix* yang dibangun menggunakan *latent semanctic indexing*. *Dimension rate* merupakan salah satu

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

fitur *latent semantic indexing*. Digunakan *dimension rate* untuk mengurangi waktu evolusi pada perhitungan *fitness* algoritma genetik.

Pengujian *dimension rate* yang dilakukan pada berbagai penelitian pada (Grossman & Frieder, 2004) menyebutkan bahwa *dimension rate* disekitar 100 memberikan performa yang baik bagi *information retrieval*. Semakin besar dimensi *matrix* pada *latent semantic indexing*, maka perhitungan yang dilakukan pada fungsi *fitness* algoritma genetik akan semakin kompleks karena ukuran dimensi *matrix* yang besar.

Oleh karena itu, berdasarkan solusi dari contoh permasalah yang diberikan oleh (Grossman & Frieder, 2004), maka *dimension rate* sama dengan 2. Namun, karena *dimension rate* bukan sebuah tetapan yang mutlak, diberikan akses *dimension rate* pada pengguna untuk melakukan eksperimen sendiri sehingga mencapai hasil klasifikasi yang optimal. Sebab koleksi dokumen dan algoritma yang berbeda membuat konfigurasi *dimension rate* juga berbeda (Grossman & Frieder, 2004).

5.1.5 Pengujian informative term threshold

Pengujian *informative term threshold* dilakukan untuk mengetahui perubahan nilai *fitness* sebuah kromosom apabila mengandung beberapa *term* pada urutan teratas berdasarkan pada perhitungan *term frequency-inverted document frequency*.

Fitness pada sebuah kromosom akan bertambah jika mengandung beberapa *term* tersebut. Semakin banyak *term* yang terdapat pada kromosom, dan *term* tersebut sama dengan beberapa *term* teratas, maka *fitness* kromosom tersebut juga semakin bertambah. Dengan demikian, maka evolusi algoritma genetik akan mendekati *term* teratas tersebut.

Hasil pengujian *informative term threshold* disajikan pada Tabel 5.7

Tabel 5.7 Hasil pengujian informative term threshold

Informative Term Threshold	Panjang Kromosom	Jumlah Term	Fitness	
			f_1	$f_2 = f_1 + \Delta$
1	10	1	11.79070668	11.79918793
5	10	5	11.42155495	11.46533183
10	10	9	11.98412423	12.05922359
15	10	4	11.62313798	11.30912942

Kolom pada Tabel 5.7 dijelaskan sebagai berikut :

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Informative term threshold merupakan sebuah konstanta yang dikonfigurasi oleh pengguna. Konstanta tersebut digunakan selama proses evolusi algritma genetik berlangsung. Pergantian nilai konstanta hanya dapat dilakukan sebelum proses evolusi algoritma genetik dilakukan.

Panjang kromosom merupakan panjang setiap kromosom dalam populasi. Diberikan panjang kromosom yang sama untuk mengetahui perubahan nilai parameter yang lain.

Jumlah *term* merupakan jumlah *term* yang sama didalam kromosom tersebut dengan *term* teratas menggunakan parameter *informative term threshold*.

Fitness sebelum perubahan (f_1) adalah nilai *fitness* kromosom tersebut sebelum dijumlahkan dengan nilai kesamaan seberapa banyak *term* dari kromosom tersebut yang sama dengan *term* teratas menggunakan parameter *informative term threshold*.

Fitness setelah perubahan (f_2) adalah nilai *fitness* kromosom tersebut setelah dijumlahkan dengan nilai kesamaan seberapa banyak *term* dari kromosom tersebut yang sama dengan *term* teratas menggunakan parameter *informative term threshold*.

Kromosom yang diujikan pada Tabel 5.7 adalah satu kromosom terbaik pada evlusi terakhir algoritma genetik didalam satu populasi. Parameter evolusi algoritma genetik yang lain disajikan pada Tabel 5.8

Tabel 5.8 Parameter algoritma genetik untuk informative term threshold

Parameter	Nilai
Jumlah Iterasi	10
Jumlah Populasi	10
Crossover Rate	0.75
Mutation Rate	0.25
Dimension Rate	2
Chromosome Length	10

Berdasarkan Tabel 5.7, nilai *fitness* kromosom bertambah jika sebagian besar pada kromosom tersebut mengandung *term* yang sama dengan *term* teratas pada *informative term threshold*. Jika kromosom terlalu panjang, namun hanya sebagian kecil alel saja yang mengandung *term* yang sama tersebut, maka nilai *fitness* kromosom tersebut lebih kecil dari sebuah kromosom yang pendek namun sebagian besar alelnya mengandung *term* teratas menggunakan *term frequency-inverted index frequency*.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Dengan demikian, algoritma genetik akan berusaha untuk menghasilkan kromosom yang sebagian besar alelnya sama dengan beberapa *term* teratas pada *informative term threshold* menggunakan *term frequency-inverted index frequency*.

5.2 Pengujian algoritma genetik

Pengujian algoritma genetik menggunakan formula *f-measure*. Kromosom dilatihkan terlebih dahulu dengan parameter yang telah ditentukan. Setelah pelatihan algoritma genetik telah selesai dilakukan, maka pengujian algoritma genetik dapat dilakukan.

Data yang digunakan pada pengujian merupakan data berita dari http://tekno.kompas.com/apps. Data tersebut dimasukkan kedalam sistem basis data, kemudian dilakukan pendaftaran kalimat, dan pendaftaran *term*.

Jumlah data berita yang didapatkan sama dengan 30. Jumlah data berita tersebut sama dengan jumlah dokumen *corpus*. Berdasarkan jumlah dokumen *corpus* tersebut, didapatkan jumlah kalimat sama dengan 387. Berdasarkan jumlah kalimat tersebut didapatkan jumlah *term* sama dengan 1620.

Parameter pengujian algoritma genetik disajikan pada Tabel 5.9 :

Tabel 5.9 Parameter algoritma genetik untuk pengujian

Parameter	Nilai
Iterasi	10
Jumlah Populasi	10
Crossover Rate	0.75
Mutation Rate	0.25
Dimension Rate	2
Informative Term Threshold	10
Chromosome Length	10

Parameter algoritma genetik yang digunakan pada pengujian algoritma genetik berdasarkan pengujian parameter algoritma genetik yang telah dilakukan.

Pengujian algoritma genetik dilakukan dengan tujuan untuk menilai sebaik mana algoritma genetik dengan parameter tertentu mendaftarkan *term* yang dianggap relevan terhadap kelas tertentu. Pengujian algoritma genetik menggunakan formula *f-measure*.

Hasil pengujian algoritma genetik disajikan pada Tabel 5.10

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.10 Hasil pengujian algoritma genetik

Kelas	Precision	Recall	F-Measure
Apa	0.25	0.01334	0.02532
Berapa	0.08333	0.02272	0.03571
Dimana	0.02479	0.11538	0.04081
Kapan	0.05882	0.02941	0.03921
Siapa	0.07692	0.04348	0.05555
Siapa	0.07692	0.04348	0.05555

Pengujian menggunakan *f-measure* menggunakan dokumen *ground truth* sebagai acuan dasar. Dokumen *ground truth* yang digunakan adalah *dataset*.

Dokumen *ground truth* tersebut dibandingkan dengan hasil satu kromosom terbaik pada akhir evolusi algoritma genetik untuk semua kelas.

Alel pada kromosom merupakan *term* dari dokumen tertentu. Dokumen tertentu tersebut dapat dilacak keberadaannya, sehingga informasi dokumen berapa saja yang berada pada kromosom tersebut dapat diketahui.

5.3 Pengujian question answering system

Pengujian *question answering system* dilakukan dengan melibatkan pengguna penguji. Pengguna penguji adalah seorang pengguna yang memiliki pengetahuan atas pertanyaan yang ditanyakan kepada sistem. Dalam hal ini, pengguna penguji sudah mengetahui jawaban yang benar berdasarkan dokumen *corpus* pada sistem.

Pengujian dilakukan berdasarkan kebenaran jawaban sistem berdasarkan pengetahuan pengguna penguji. Jawaban sistem adalah jawaban yang memiliki nilai tertinggi dan calon jawaban yang ditampilkan oleh sistem.

Jawaban yang dikembalikan sistem bernilai benar jika jawaban tersebut sama dengan pengetahuan pengguna penguji berdasarkan dokumen *corpus*. Jawaban yang dikembalikan sistem bernilai salah jika jawaban tersebut tidak sama dengan pengetahuan pengguna penguji berdasarkan dokumen *corpus*. Jika sistem tidak dapat mengembalikan jawaban, maka jawaban sistem tersebut tidak mendapatkan nilai, karena tidak ada jawaban untuk dinilai.

Hasil pengujian *question answering system* menggunakan algoritma genetik berdasarkan parameter pada Tabel 5.3, dokumen *corpus,* koleksi *dataset* disajikan pada Tabel 5.3

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.11 Jawaban sistem berdasarkan pertanyaan pengguna

No	Pertanyaan	Jawaban	Respon Pengguna Penguji
1	Siapa mantan pegawai badan keamanan nasional amerika serikat ?	KOMPAS.com - Aplikasi chatting bertajuk "Signal for Android" resmi diluncurkan awal November 2015 ini. Mantan pegawai Badan Keamanan Nasional Amerika Serikat (NSA) Edward Snowden menganjurkan masyarakat untuk menggunakan aplikasi tersebut.	Benar
2	Siapa yang melakukan update pada 30 oktober lalu ?	Baidu yang melakukan update pada 30 Oktober lalu menjelaskan, "Kode yang diidentifikasi Trend Micro berpotensi berbahaya kini menjadi kode mati setelah kami perbarui, tidak ada efeknya lagi	Benar.
3	Siapa yang melakukan enkripsi percakapan ?	Kami tak bisa mendengar percakapanmu atau membaca pesanmu, tak ada yang bisa. Apapun di Signal for Android dienkripsi terus menerus, dan dengan susah payah digodok untuk menjamin keamanan berkomunikasi," begitu tertulis pada situs resmi Open Whisper Systems (OWS) yang menggodok aplikasi tersebut	Benar.
4	Siapa yang dapat membeli koin ?	Pemain juga dapat membeli koin sebanyak yang diinginkan dengan harga tertentu. Koin tersebut berfungsi untuk menggunakan alat bantu. Sebab, setiap penjajalan alat bantu, pemain harus mengeluarkan lima koin.	Benar.
5	Siapa petugas kecamatan ?	"Setidaknya 18.169 kepala keluarga di delapan kecamatan yang butuh bantuan air bersih. Daerah Wonogiri bagian selatan yang paling banyak terkena dampak kekeringan,"katanya saat dikonfirmasi, Rabu (4/11/2015).	Salah.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.12 Jawaban sistem berdasarkan pertanyaan pengguna (lanjutan)

No	Pertanyaan	Jawaban	Respon Pengguna Penguji
6	siapa yang dapat memilih kategori pembuatan proposal ide dan pembuatan aplikasi lewat coding ?	Para peserta di ajang ini dapat memilih kategori pembuatan proposal ide dan pembuatan aplikasi lewat coding competition. Hadiah yang diperebutkan peserta berupa piala, sertifikat, dan gadget terbaru.	Benar.
7	Dimana kunjungan tim terbaik setelah mendapatkan uang tunai ?	Tahap ini untuk menentukan siapakah tiga tim terbaik yang bakal mendapatkan hadiah uang tunai dan kunjungan belajar ke pelaku industri telekomunikasi besar di luar negeri. Selain itu ada juga pemenang favorit pilihan publik yang bisa Anda pilih melalui tautan ini.	Benar.
8	Dimana rudiantara saat memberikan pendapat pentingnya usaha rintisan startup yang berkualitas ?	JAKARTA, KOMPAS.com - Menteri Komunikasi dan Informatika Rudiantara menyoroti pentingnya kompetisi aplikasi dalam upaya menumbuhkan ekonomi digital di Indonesia. Terutama dengan merangsang munculnya usaha rintisan digital atau startup yang berkualitas.	Benar.
9	Dimana Ari Grant ?	Masalah ini diakui langsung oleh Engineer Manager Facebook Ari Grant.	Benar.
10	Berapa ruang penyimpanan yang didukung oleh Joox ?	Joox juga dapat menjadi Music Player dengan cara mengimpor koleksi musik mp3 yang ada di dalam smartphone. Musik dalam Joox dapat disimpan atau dipindahkan ke SD card; hal ini sangat penting bagi yang ingin terus menikmati musik namun memiliki memori dan ruang penyimpan terbatas.	Salah.

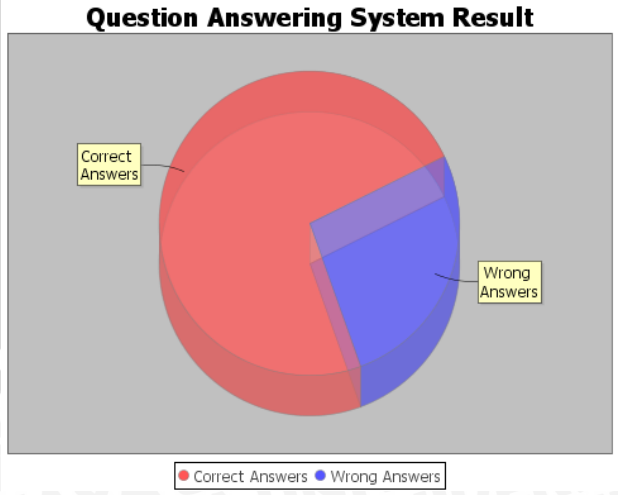
repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Tabel 5.13 Jawaban sistem berdasarkan pertanyaan pengguna (lanjutan)

No	Pertanyaan	Jawaban	Respon Pengguna Penguji
13	Kapan karyawan twitter mengungkapkan kekecewaannya ?	Bahkan karyawan yang bekerja di Twitter pun mengungkapkan kekecewaannya terhadap keputusan twitter tersebut.	Salah.
14	Kapan tanggapan baidu dilontarkan ?	Tanggapan Baidu	Salah.
15	Kapan gemastik 8 diselenggarakan ?	Aplikasi ini berhasil meraih juara 3 pada Kategori Pengembangan Aplikasi Games dalam Pagelaran Mahasiswa Nasional Bidang Teknologi Informasi dan Komunikasi (Gemastik) 8 Tahun 2015.	Benar.

Berdasarkan Tabel 5.11, 5.12 dan 5.13 didapatkan 15 pertanyaan dari pengguna penguji. 4 jawaban dari sistem dianggap salah, dan 11 jawaban dari sistem dianggap benar. Dengan demikian akurasi yang diberikan oleh sistem berdasarkan konfigurasi parameter algoritma genetik, hasil klasifikasi, pertanyaan pengguna penguji, dan jawaban dari sistem mencapai 73.32 %.

Akurasi sistem yang disajikan dapat direpresentasikan kedalam diagram yang disajikan oleh Gambar 5.1



Gambar 5.1 Akurasi implementasi algoritma genetik pada question answering system

BAB 6 PENUTUP

6.1 Kesimpulan

Berdasarkan setiap proses penelitian Implementasi Algoritma Genetik pada *Question Answering System* didapatkan kesimpulan sebagai berikut :

1. Algoritma genetik dapat diimplementasikan pada *question answering system*. Hal ini berdasarkan pengujian *question answer*. Sistem mampu menjawab benar 11 pertanyaan dari 15 pertanyaan yang diajukan.
2. Semakin baik kualitas *dataset* maka semakin baik hasil klasfikasi. Kalimat didalam *dataset* menjadi acuan awal algoritma genetik, sehingga algoritma genetik berusaha untuk menghasilkan kalimat yang serupa dengan *dataset*.

6.2 Saran

Terdapat beberapa saran untuk memperbaiki kekurangan dan kelemahan Implementasi Algoritma Genetik pada *Question Answering System*. Saran tersebut adalah sebagai berikut :

1. Proses evolusi algoritma genetik memerlukan waktu yang relatif lama untuk melakukan klasifikasi seiring dengan bertambahnya data, sehingga diperlukan sebuah metode untuk merepresentasikan data tersebut supaya waktu yang diperlukan untuk proses evolusi menjadi lebih singkat.
2. Diperlukan sebuah *web crawler* yang terintegrasi terhadap sistem basis data, agar data pada *question answering* dapat terus diperbaharui sesuai dengan *crawler* tersebut

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

DAFTAR PUSTAKA

Aggraeny, M. D., 2007. Implementasi Question Answering System dengan metode Rule-Based pada terjemahan Al Qurán Surat Al Baqarah. *Institut Pertanian Bogor.*

Anon., 1995. *Q1.1: What's a Genetic Algorithm (GA).* [Online] Available at: http://www.cs.cmu.edu/Groups/AI/html/faqs/ai/genetic/part2/faq-doc-2.html

Ariyanti, G., 2010. Dekomposisi Nilai Singular dan Aplikasinya. pp. 4-7.

Barskar, R., Ahmed, G. F. & Barskar, N., 2011. An Approach for Extracting Exact Answers to Question Answering (QA) System for English Sentences. *Elsevier,* pp. 1-7.

Bentley, P. J. & Wakefield, J. P., 1996. Hierarchical Crossover in Genetic Algorithms. pp. 1-8.

Biswas, P., Sharan, A. & Malik, N., 2014. A Framework for Restricted Domain Question Answering System. *IEEE,* p. 613.

Bock, H., 2011. *The Definitive Guide to Netbeans Platfom 7.* Berkely: Apress Berkely.

Brill, E., Dumais, S. & Banko, M., 2002. An Analysis of The AskMSR Question-Answering System. *Microsoft Research, One Microsoft Way,* p. 1.

Day, M. Y., Ong, C. S. & Lian-Hsu, W., 2007. Question Classification in English-Chinese Cross-Language Question Answering : An Integrated Genetic Algorithm and Machine Learning Approach. *IEEE.*

Figueroa, A. G. & Neumann, G., 2008. Genetic Algorithms for data-driven Web Question Answering. *MIT Evolutionary Computation.*

Gotshall, S. & Rylander, B., 2002. Optimal Population Size and The Genetic Algorithm.

Grossman, D. A. & Frieder, O., 2004. *Information Retrieval : Algorithms and Heuristics.* s.l.:Springer.

Gunawan & Lovina, G., 2006. Question Answering System dan Penerapannya pada Alkitab. *Jurnal Informatika,* p. 1.

Kadir, A., 2003. *Dasar Pemograman Java 2.* Innsbruck: Penerbit Andi.

Kim, Y. & Wreck, O. d., 2004. Variable Chromosome Length Genetic Algorithm for Structural Topology Design Optimization. *American Institute of Aeronautics and Astronautics,* p. 1.

Koepke, H., 2007. Latent Semantic Indexing. p. 16.

Mahmudy, W. F., 2013. *Algoritma Evolusi.* Malang: s.n.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Manning, C. D., Raghavan, P. & Schütze, H., 2009. *Introduction to Information Retrieval.* s.l.:Springer.

Mitchell, M., 1998. *An Introduction to Genetic Algorithms.* s.l.:MIT Press.

Paulson, C., 2010. *Latent Semantic Analysis (LSA) Tutorial - Part 2 - Modify the Counts with TFIDF.* [Online] Available at: http://www.puffinwarellc.com/index.php/news-and-articles/articles/33-latent-semantic-analysis-tutorial.html?showall=1

Porter, M., 2006. *The Porter Stemmer Algorithm.* [Online] Available at: http://tartarus.org/martin/PorterStemmer/

Shalahuddin, M. & S, R. A., 2013. *Rekayasa Perangkat Lunak Terstruktur dan Beroritentasi Objek.* Bandung: Penerbit Informatika.

Skinner, M., 2001. *Genetic Algorithms Overview.* [Online] Available at: http://geneticalgorithms.ai-depot.com/Tutorial/Overview.html

Suyanto, 2007. *Artificial Intelligence.* Bandung: Penerbit Informatika.

Todd, R., 2015. *Orthogonal Matrix.* [Online] Available at: http://mathworld.wolfram.com/OrthogonalMatrix.html

Turian, J., 2012. *What's the difference between Latent Semantic Indexing (LSI) and Latent Dirichlet Allocation (LDA)?.* [Online] Available at: http://www.quora.com/Whats-the-difference-between-Latent-Semantic-Indexing-LSI-and-Latent-Dirichlet-Allocation-LDA#

Weisstein, E., 2015. *Diagonal Matrix.* [Online] Available at: http://mathworld.wolfram.com/DiagonalMatrix.html

Weisstein, E. W., 2016. *Singular Matrix.* [Online] Available at: http://mathworld.wolfram.com/SingularMatrix.html [Accessed 18 January 2016].

LAMPIRAN A DATA

Corpus ID : 127

Sumber :

http://tekno.kompas.com/read/2015/11/03/20220027/100.juta.perangkat.android.terinfeksi.gara-gara.baidu

KOMPAS.com -Alat pembuat aplikasi (software development kit-SDK) Android miliki perusahaan layanan internet China Baidu disinyalir mengandung komponen yang membahayakan. SDK tersebut kini sudah ada di dalam ribuan aplikasi Android.

SDK bernama Moplus tersebut, menurut PC World dan dikutip KompasTekno, Selasa (3/11/2015), telah diintegrasikan dalam 14.000 aplikasi Android, dimana 4.000 aplikasi di antaranya dibuat oleh Baidu.

Aplikasi-aplikasi yang terinfeksi itu menurut firma keamanan internet Trend Micro telah dipakai oleh sekitar 100 juta pengguna perangkat Android di seluruh dunia.

Menurut penelitian Trend Micro, SDK Moplus bisa mengaktifkan HTTP server di perangkat yang terdapat aplikasi terinfeksi. Server tersebut tidak dibekali proses otentikasi dan bisa menerima request dari siapa saja di internet.

Dengan mengirim kode perintah tertentu ke perangkat Android yang sudah aktif HTTP server-nya, peretas bisa membuat perintah tertentu yang tertanam di SDK.

Cara tersebut bisa digunakan peretas untuk mengakses informasi sensitif, seperti data lokasi, kata pencarian, menambah kontak baru secara diam-diam, mengunggah file, melakukan panggilan telepon, menampilkan pesan, hingga menginstal aplikasi.

Menurut Trend Micro, lubang keamanan dalam SDK Moplus ini lebih berbahaya dibandingkan bug Stagefright yang juga sempat heboh sebelumnya.

Sebab, dijelaskan Trend Micro, Stagefright membutuhkan peretas untuk mengirim pesan multimedia ke pemilik smartphone, sementara dalam SDK Moplus ini tidak. Trend Micro telah memperingatkan Baidu dan Google akan celah keamanan ini.

Tanggapan Baidu

Baidu sendiri telah merilis versi SDK baru yang beberapa perintah di dalamnya telah dihapus, namun HTTP server tersebut masih terbuka dan beberapa fungsi bisa disalahgunakan.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Baidu yang melakukan update pada 30 Oktober lalu menjelaskan, "Kode yang diidentifikasi Trend Micro berpotensi berbahaya kini menjadi kode mati setelah kami perbarui, tidak ada efeknya lagi."

"Tidak ada backdoor lagi," imbuh perwakilan Baidu. Kode yang tidak aktif lagi itu juga akan dihapus di aplikasi versi berikutnya untuk menjernihkan masalah.

Kini, masalahnya tinggal secepat apa pembuat aplikasi pihak ketiga yang memakai SDK tersebut memperbaruinya dengan versi terbaru yang dirilis Baidu.

Corpus ID : 56

Sumber : Penelitian

Sherlock Holmes adalah seorang detektif ternama di Inggris sherlock holmes juga terkenal.Watson adalah sahabat detektif ternama watson juga merupakan penulis ternama. Mereka membuka biro detektif di Baker Street.

Dimanakah bagianmu, Holmes ?

Bagianku, selalu ada dalam botol kecil ini

Ternyata ada dua Holmes, aku tidak percaya

Benar, Mycroft Holmes adalah saudaraku

Ada tiga gelas di atas meja

Salah satu gelas pecah.

Corpus ID : 141

Sumber :

http://tekno.kompas.com/read/2015/10/25/08300067/Di.Indonesia.Joox.Punya.Tim.Khusus.Pilih.Musik

KOMPAS.com - Satu lagi layanan streaming musik masuk ke pasar aplikasi Indonesia. Dirilis pertama kali di Hong Kong dan Malaysia pada awal 2015, aplikasi Joox akhirnya hadir juga untuk perangkat Androiddan iOS di Indonesia.

Untuk menembus pasar Indonesia yang sudah mulai ramai dengan aplikasi streaming musik lain, sebagaimana dirangkum KompasTeknodari keterangan pers yang diterima, Sabtu (24/10/2015), Joox mengandalkan sebuah tim editor musik asli Indonesia.

Tim tersebut bertugas untuk mengurasikan musik dan memasukkannya ke playlist radio gratis, berisikan musik-musik yang sedang digemari atau nge-tren di Tanah Air.

Bagi yang belum tahu akan mendengarkan musik apa, Joox bakal hadir dengan fitur Editors' Pick, berisikan lagu-lagu yang dipilihkan langsung oleh tim editor.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Selain itu, ada juga fitur Top Charts, berisikan lagu-lagu yang paling banyak didengar. Daftar Top Charts ini diisi oleh jajaran artis, baik dalam negeri maupun luar negeri.

Ada juga fitur New Releases. Seperti namanya, ia akan berisikan berbagai lagu teranyar yang baru dilepas oleh penyanyi atau band.

Musik yang Legal

Melalui aplikasi ini, pengguna juga bisa mencari lagu yang diinginkan berdasarkan judul, penyanyi, hingga genre. Kelebihan dari aplikasi ini, semua musik yang ada dalam library sudah berlisensi. Oleh karena itu, pengguna bisa mendengarkan musik secara legal.

Joox juga dapat menjadi Music Player dengan cara mengimpor koleksi musik mp3 yang ada di dalam smartphone. Musik dalam Joox dapat disimpan atau dipindahkan ke SD card; hal ini sangat penting bagi yang ingin terus menikmati musik namun memiliki memori dan ruang penyimpan terbatas.

Bagi penggemar karaoke atau sing-along, fitur Lyric yang ada di setiap lagu akan jadi teman menyanyi yang menyenangkan. Tampilan koleksi/library juga dapat dikustomisasi dengan aneka warna-warni atau grafis yang bisa dipilih dari Theme Gallery.

Bisa berlangganan/

Kebanyakan fitur tersebut tersedia secara gratis. Meskipun begitu, ada beberapa lagu yang tidak bisa dinikmati. Untuk bisa mendengarkan beberapa musik tertentu, pengguna diharuskan untuk berlangganan atau menjadi VIP di layanan ini.

Ada beberapa keuntungan yang bisa didapatkan ketika berlangganan, seperti menikmati semua musik yang ada (menurut klaim Joox, ada tiga juta lagu yang ada di library), mendengarkan musik secara offline, kualitas suara yang lebih baik, dan tanpa iklan.

Biaya berlangganannya pun beragam, mulai dari Rp 25.000 untuk masa berlangganan 1 minggu hingga Rp 509.000 untuk biaya berlangganan 1 tahun.

Untuk mengunduh Joox di Android silahkan kunjungi tautan berikut. Untuk versi iOS, kunjungi tautan berikut ini. Joox juga bisa dinikmati melalui web melalui www.joox.com.

Corpus ID : 132

Sumber :

http://travel.kompas.com/read/2015/11/04/135614227/Bau.Belerang.sampai.ke.Plawangan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

MATARAM, KOMPAS.com - Bau belerang akibat dari erupsi Gunung Baru Jari, anak Gunung Rinjani di Pulau Lombok, Nusa Tenggara Barat, sudah tercium sampai Plawangan, atau lebih kurang 3 kilometer dari titik erupsi.

Hal ini diungkap Pengelola Taman Nasional Gunung Rinjani, Basitya saat dihubungi KompasTravel, Rabu (4/11/2015).

"Baunya belerangnya sudah sampai Plawangan mas," kata Basitya.

Sementara berdasarkan citra dari Satelit Cuaca Himawari Badan Meteorologi Klimatologi dan Geofisika, sebaran abu vulkanik mengarah ke barat, dan sudah mencapai Bali dan Banyuwangi, Jatim.

Gunung Baru Jari kembali menunjukkan peningkatan aktivitas Senin (3/11/2015) kemarin. Hingga saat ini pendakian ke Rinjani ditutup total sampai waktu yang belum ditentukan.

Corpus ID : 133

Sumber :

http://regional.kompas.com/read/2015/11/04/13540111/Hujan.Tak.Kunjung.Datang.Status.Darurat.Kekeringan.di.Wonogiri.Diperpanjang.

WONOGIRI, KOMPAS.com - Karena hujan tak kunjung datang, status darurat kekeringan di Kabupaten Wonogiri, Jawa Tengah, diperpanjang hingga akhir bulan November.

Sebelumnya, Badan Penanggulangan Bencana Daerah (BPBD) Wonogiri memperkirakan status darurat kekeringan hingga bulan Oktober. Hal tersebut dilakukan setelah ada data pemetaan dari Stasiun Klimatologi Semarang.

Kepala BPBD Wonogiri, Jawa Tengah, Bambang Haryanto, menjelaskan bahwa perpanjangan status darurat kekeringan di Kabupaten Wonogiri akan ditindak lanjuti dengan perpanjangan bantuan air bersih kepada masyarakat.

Berdasarkan data yang dimiliki BPBD, kurang lebih 8 kecamatan di Wonogiri mengalami krisis air bersih.

"Setidaknya 18.169 kepala keluarga di delapan kecamatan yang butuh bantuan air bersih. Daerah Wonogiri bagian selatan yang paling banyak terkena dampak kekeringan,"katanya saat dikonfirmasi, Rabu (4/11/2015).

Bambang menambahkan, kecamatan tersebut antara lain, Paranggupito, Giritontro, Giriwoyo, Pracimantoro, Eromoko dan Manyaran. Selain itu, warga di lokasi krisis air bersih, kelimpungan akibat kemarau yang tak kunjung usai ini.

Warga harus merogoh kocek lebih dalam untuk mendapatkan air bersih. Mirisnya, banyak warga yang sudah tidak mampu lagi untuk membeli air bersih dan mengandalkan bantuan air bersih dari donatur atau pemerintah.

"Dulu awal awal musim kemarau masih bisa beli, tetapi saat ini, warga tidak mampu lagi membeli, harganya sudah 250 ribu per tangki, yang isinya 6 ribu liter," kata Warsito, Camat Pracimantoro saat dihubungi.

Corpus ID : 142

Sumber :

http://tekno.kompas.com/read/2015/10/24/11110047/Facebook.Bikin.Google.Internal?utm_source=WP&utm_medium=box&utm_campaign=Kknwp

KOMPAS.com - Jejaring sosial Facebook akan meluncurkan fitur pencarian sejati di dalam aplikasinya. Sejati artinya, semua postingan yang diunggah di *Facebook* akan di-indeks secara langsung, layaknya pencarian *Google* namun kali ini hanya di dalam *Facebook*.

Fitur pencarian baru ini dikutip KompasTekno dari The Verge, Jumat 923/10/2015), akan berdasar kepada orang-orang yang dikenal pengguna di Facebook, dan apa yang mereka unggah atau katakan di Facebook.

Facebook Search yang baru akan diluncurkan pertama kali bagi wilayah-wilayah yang berbasis bahasa Inggris, dan tersedia di platform iOS, Android, dan desktop dalam beberapa minggu ke depan.

Sebelumnya, Facebook memang telah merilis pencarian Graph Search, namun pencarian ini masih terkesan setengah-setengah, Graph Search tidak mencari postingan publik yang tidak dibagi dengan pencarinya.

Fitur pencarian ini dikatakan oleh Facebook Search Team, Roussea Kazi, sebagai upaya untuk mengubah Facebook menjadi bagian dari percakapan global.

"Kami meng-indeks semua percakapan di dunia, kami tunjukkan pa yang sedang menjadi tren, peristiwa penting, dan apa yang sedang terjadi saat ini," ujarnya.

Facebook mengatakan dalam membuat indeks pencarian ini mereka akan melakukannya dengan pendekatan kuratorial, hasil pencarian akan ditampilkan berdasar urutan yang paling akurat, penting dan dianggap benar.

Postingan dari akun Facebook organisasi akan lebih sering ditaruh di deretan atas,diikuti kemudian oleh postingan dari teman, topik masalah, dan berdasar lokasi geografis.

Pencarian tersebut juga bisa dipersonalisasi, layaknya Google Search, berdasar perilaku dan minat pengguna Facebook. Namun berbeda dari Google, hasil pencarian Facebook ini diklaim akan lebih presisi.

Facebook mengatakan konten yang diunggah dan diatur sebagai konten pribadi (private), tidak akan muncul dalam pencarian. Walau demikian, belum tentu semua pengguna Facebook mengerti bagaimana cara mengaturnya.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Corpus ID : 143

Sumber :

http://tekno.kompas.com/read/2015/10/26/14100067/fitur.quick.reply.hadir.di.whatsapp.untuk.ios.9

KOMPAS.com - WhatsApp di iOS mendapatkan pembaruan. Kini pengguna bisa langsung membalas pesan yang diterima dari layar notifikasi, tanpa harus membuka aplikasi WhatsApp.

Fitur Quick Reply itu hanya berlaku bagi pengguna dengan sistem operasi terbaru iOS 9.1 yang mulai dirilis akhir pekan lalu.

Sementara bagi pengguna jam tangan Apple Watch yang menjalankan watchOS 2.0.1, reply pre-set bisa digunakan untuk menjawab pesan WhatsApp.

Dikutip KompasTekno dari Phone Arena, Senin (26/10/2015), Quick Reply pertama kalinya dikenalkan Apple lewat peranti iPhone dengan sistem operasi iOS 8.Fitur itu tadinya dirancang untuk aplikasi pesan Apple sendiri, yaitu iMessages.

Kini di iOS 9, Apple mengizinkan Quick Reply dipakai oleh aplikasi pesan instan buatan pihak ketiga. Selain WhatsApp, fitur Quick Reply juga telah diadopsi oleh Skype dan Facebook Messenger.

Sementara bagi pengguna WhatsApp di piranti iPhone 6s dan iPhone 6s Plus, update terbaru di aplikasi ini memungkinkan pengguna memanfaatkan fitur 3D Touch.

Apple menjelaskan, sentuhan ringan memungkinkan pengguna "mengintip" konten pesan di WhatsApp, sementara tekanan yang lebih keras akan memunculkan halaman WhatsApp.

Selain kemamuan Quick Reply dan 3D Touch, WhatsApp di iOS 9.1 juga disertai dengan 150 emoji baru.

Corpus ID : 144

Sumber :

http://tekno.kompas.com/read/2015/11/02/10070007/Google.Bikin.Aplikasi.Pencari.Teman.Nongkrong

KOMPAS.com - Tanpa pengumuman, Google merilis sebuah aplikasi baru bernama Who's Down. Pengguna bisa memakainya untuk menemukan kawan yang siap untuk diajak pergi, entah untuk nongkrong atau melakukan kegiatan tertentu.

Who's Down sudah bisa diunduh melalui Play Store dan ada juga versi iOS. Namun untuk sementara ini aplikasi tersebut hanya terbuka untuk mereka yang sudah memperoleh undangan, belum bisa dipakai secara bebas.

Dilansir KompasTekno dari 9to5Google, Senin (2/11/2015), cara memakai aplikasi ini tampaknya sederhana dan hanya untuk tujuan menunjukkan siapa yang sedang siap diajak jalan-jalan offline.

Hanya sebuah switch dengan fungsi semacam indikator on/off yang diatur untuk memberi tahu kawan-kawan bahwa Anda siap diajak mengerjakan sesuatu.

Setelah menyalakan switch tersebut, maka akun Anda akan terus "tersedia" selama tiga jam. Selain itu ada juga pilihan berbagi aktivitas untuk menunjukkan kegiatan yang diminati saat itu.

Pengguna juga bisa melihat daftar teman yang siap diajak melakukan berbagai kegiatan. Bila ingin mengulik lebih jauh, bisa juga membuka obrolan dengan akun yang tersedia.

Corpus ID : 145

Sumber :

http://regional.kompas.com/read/2015/11/04/13540111/Hujan.Tak.Kunjung.Datang.Status.Darurat.Kekeringan.di.Wonogiri.Diperpanjang.

WONOGIRI, KOMPAS.com - Karena hujan tak kunjung datang, status darurat kekeringan di Kabupaten Wonogiri, Jawa Tengah, diperpanjang hingga akhir bulan November.

Sebelumnya, Badan Penanggulangan Bencana Daerah (BPBD) Wonogiri memperkirakan status darurat kekeringan hingga bulan Oktober. Hal tersebut dilakukan setelah ada data pemetaan dari Stasiun Klimatologi Semarang.

Kepala BPBD Wonogiri, Jawa Tengah, Bambang Haryanto, menjelaskan bahwa perpanjangan status darurat kekeringan di Kabupaten Wonogiri akan ditindak lanjuti dengan perpanjangan bantuan air bersih kepada masyarakat.

Berdasarkan data yang dimiliki BPBD, kurang lebih 8 kecamatan di Wonogiri mengalami krisis air bersih.

"Setidaknya 18.169 kepala keluarga di delapan kecamatan yang butuh bantuan air bersih. Daerah Wonogiri bagian selatan yang paling banyak terkena dampak kekeringan,"katanya saat dikonfirmasi, Rabu (4/11/2015).

Bambang menambahkan, kecamatan tersebut antara lain, Paranggupito, Giritontro, Giriwoyo, Pracimantoro, Eromoko dan Manyaran. Selain itu, warga di lokasi krisis air bersih, kelimpungan akibat kemarau yang tak kunjung usai ini.

Warga harus merogoh kocek lebih dalam untuk mendapatkan air bersih. Mirisnya, banyak warga yang sudah tidak mampu lagi untuk membeli air bersih dan mengandalkan bantuan air bersih dari donatur atau pemerintah.

"Dulu awal awal musim kemarau masih bisa beli, tetapi saat ini, warga tidak mampu lagi membeli, harganya sudah 250 ribu per tangki, yang isinya 6 ribu liter," kata Warsito, Camat Pracimantoro saat dihubungi.

Corpus ID : 146

Sumber :

http://tekno.kompas.com/read/2015/10/25/14270087/Indosat.Kembali.Gelar.Hackathon.Anak.dan.Remaja

KOMPAS.com - Indosat kembali menyelenggarakan Kids & Teens Hackathon, Sabtu (24/10/2015). Dalam ajang tersebut, anak-anak dan remaja berusia 6 hingga 15 tahun berkompetisi membuat aplikasi dalam waktu tertentu.

"Teknologi kini menjadi bagian yang tidak terpisahkan dari anak-anak dan remaja. Oleh karena itu, Indosat mengajak mereka untuk tidak sekadar menjadi pengguna tetapi juga content creator," ujar Deva Rachman, Group Head Corporate Communication Indosat, dalam keterangan pers yang KompasTekno terima, Minggu (25/10/2015).

"Melalui kompetisi ini, kami berharap anak-anak dan remaja semakin menyukai dunia pemrograman yang mampu melatih mereka berpikir logis serta dapat menjadi media untuk mewujudkan kreasi dan imajinasi anak-anak dan remaja ke dalam suatu aplikasi," lanjutnya.

Kids & Teens Hackaton kali ini bekerja sama dengan Clevio Coder Camp, sebuah tempat belajar pemrograman komputer bagi anak-anak yang menggunakan metode belajar Self-Organizing Learning Environment, Games, dan Team Work.

Para peserta di ajang ini dapat memilih kategori pembuatan proposal ide dan pembuatan aplikasi lewat coding competition. Hadiah yang diperebutkan peserta berupa piala, sertifikat, dan gadget terbaru.

Kompetisi Kids & Teens Hackathon ini sendiri sudah dua kali diadakan oleh Indosat. Sebelumnya, acara ini diadakan bekerja sama dengan Cody's App Academy.

Acara itu meruapakan salah satu rangkaian dari Indosat Wireless Innovation Contest (IWIC) ke-9.

IWIC ke-9 menghadirkan beberapa kategori untuk diikuti yaitu kategori Kids (SD), Teens (SMP dan SMA), Mahasiswa dan Umum, Developer, serta kategori spesial untuk Perempuan dan Inbound Tourism untuk pembuatan Ide dan Aplikasi dalam bidang (1) kategori Communication, Lifestyle, Education; (2) kategori Multimedia & Games; (3) kategori Utilities (tools, security, apps for disable).

Peserta akan berkesempatan mendapatkan ratusan juta rupiah, saldo Indosat Dompetku, dan Beasiswa di Founder Institute.

Corpus ID : 147

Sumber :

http://tekno.kompas.com/read/2015/11/04/07430057/Ini.Dia.4.Aplikasi.Terpopuler.untuk.Memantau.Pasar.Saham.

KOMPAS.com - Smartphone tak lagi sekadar menjadi alat komunikasi atau sarana berinteraksi di media sosial. Pergerakan harga saham di pasar keuangan pun kini bisa mudah dipantau lewat peranti ini.

Beragam aplikasi pemantau harga saham berbasis Android sekarang bertebaran. Memantau perubahan harga saham dalam hitungan detik bukan lagi persoalan, sejalan dengan teknologi seluler dan gadget.

Berikut ini sejumlah aplilkasi tersebut:

1. IDX mobile

IDX mobile akan menyediakan informasi secara real time mengenai running trade, order book, stock watch hingga stock quotes. Setelah mendaftar, pengguna akan mendapat email verifikasi berupa kode untuk masuk ke aplikasi ini.

Pada awalnya pengguna akan disuguhi sembilan menu bar dengan Gambar ikon sesuai fungsinya. Ikon itu antara lain menu index, stock, news, virtual trading, portofolio, message, setting, notify dan menu ikon sign out.

Yang menarik dari IDX berbasis Android ini adalah Anda dapat memantau indeks harga saham dilengkapi dengan perubahan data secara real time. Pergerakan indeks tersebut terkoneksi langsung dengan Bursa Efek Indonesia (BEI).

2. Info Saham Bursa Efek

Dengan mudah melalui aplikasi 'Info Saham Bursa Efek' ini secara update akan menampilkan pergerakan pasar saham sesuai yang dibutuhkan. Penggunaannya pun secara otomatif terkoneksi dengan BEI dan tidak memerlukan proses registrasi.

Cukup download dan mulai menjalankan maka selanjutnya muncul tiga menu utama, yaitu global, regional dan news. Bila memilih menu global akan memperlihatkan pergerakan indeks harga saham seluruh dunia berbentuk Tabel.

Namun, jika memilih menu regional terdapat beberapa pilihan menu lagi mengenai indeks saham regional hingga pergerakan saham paling aktif ataupun terpopuler. Lalu, ada berita ter-update membahas mengenai info harga saham terkini.

Data pada aplikasi ini dikumpulkan dari berbagai sumber. Nilai dan kontennya pun bersifat dinamis dan berubah setiap waktu. Dengan begitu, Anda tak perlu khawatir ketinggalan informasi pasar saham setiap harinya.

Shutterstock

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Ilustrasi

3. Poems

Aplikasi pemantau pasar saham lainnya yaitu bernama Poems. Dengan menggunakan sistem online trading, aplikasi ini cukup populer berkembang dikalangan para investor dalam memonitor pergerakan harga saham secara real time.

Hanya saja bagi tidak bisa sembarang login ke aplikasi ini, karena dikhususkan untuk para nasabah 'Poems' yang telah terdaftar. Namun pada laman muka aplikasi ini sudah menampilkan nilai pergerakan kurs secara update.

4. MetaTrader 4

Berbeda dengan aplikasi saham lainnya. MetaTrader 4 akan membantu Anda dalam memantau pergerakan kurs setiap detiknya. Menariknya aplikasi ini akan menampilkan grafik pergerakan kurs secara real time.

Aplikasi ini mendukung trading Forex yang selalu update nilai jual dan beli setiap detiknya. Desain aplikasi ini cukup detail dan sederhana yang menampilkan menu di bagian kiri layar.

Kini, device Anda pun akan semakin berguna dalam genggaman, karena tak hanya aplikasi chatting saja melainkan aplikasi trading forex pun semakin mudah diakses. Hingga saat ini telah terkoneksi dengan 1.000 lebih server dari perusahaan pialang yang menggunakan platform perdagangan MetaTrader 4.

Selain itu, dukungan gadget yang mumpuni saat memantau pergerakan pasar saham cukup dibutuhkan. Salah satunya menggunakan Tabelt Samsung Galaxy Tab S2. Dengan ukuran layar dalam dua varian, yaitu 8 inci dan 9,7 inci semakin memanjakan mata ketika menjalankan aplikasi pasar saham.

Tabelt ini pun didukung kecepatan koneksi 4G LTE yang dilengkapi prosesor Octacore 3GB. Dengan begitu, tak perlu khawatir menjalankan fungsi multi-tasking saat memantau pasar saham sambil berselancar di media sosial.

Desain tipisnya yang hanya 5,6 milimeter menambah kesan elegan ketika ditenteng dan digunakan saat melihat trading forex dimana pun. Dengan demikian, koneksi antara aplikasi dengan gadget semakin memberikan kenyamanan saat berinvestasi.

Corpus ID : 148

Sumber :

http://tekno.kompas.com/read/2015/10/28/23091937/Ini.Dia.Pemenang.Lomba.Aplikasi.Telkomsel.NexDev

JAKARTA, KOMPAS.com - Setelah mengumpulkan 20 tim finalis untuk mempresentasikan karya masing-masing di Glass House, Pacific Place, para juri

kompetisi Telkomsel The Next Dev telah memilih tiga pemenang yang diumumkan dalam acara awarding night di Jakarta, Kamis (22/10/2015).

Juri menilai tim dan aplikasi pemenang berdasarkan beberapa kriteria, di antaranya kerjasama tim, produk/ fitur yang ditawarkan, model bisnis yang dipakai, target pasar, serta scalability dari aplikasi yang bersangkutan.

Sebelumnya, sejak mulai digelar Juni lalu, The NextDev berhasil menjaring lebih dari 400 aplikasi solusi perkotaan.

Enam sub-tema untuk aplikasi yang dilombakan adalah Pemerintahan (E-Government), Kesehatan (E-Health), Pariwisata (E-Tourism), Pendidikan (E-Education), Transportasi Publik (E-Transportation), serta Usaha Mikro, Kecil, dan Menengah (E-UKM).

Pemenang The NextDev mendapat hadiah berupa dukungan "5M", yakni Market Access, Marketing Publicity, Mentoring, Management Trip, dan Money untuk mendukung pengembangan karya.

Berikut ini profil singkat dari ketiga pemenang.

RumahSinau (E-Education)

Oik Yusuf/ Kompas.com

Direktur Utama Telkomsel Ririek Adriansyah (paling kiri), bersama tim pengembang RumahSinau

Merupakan platform belajar yang dapat mempertemukan pemilik ruang kosong dengan orang yang membutuhkannya untuk kegiatan belajar mengajar atau les

GandengTangan (E-Government)

Oik Yusuf/ Kompas.com

Direktur Utama Telkomsel Ririek Adriansyah (paling kiri), bersama tim pengembang GandengTangan

Dibuat sebagai wadah kolaborasi untuk mempertemukan pemilik usaha sosial dengan publok yang ingin membantu memberikan pinjaman.

Jejakku (E-Tourism)

Oik Yusuf/ Kompas.com

Direktur Utama Telkomsel Ririek Adriansyah (paling kiri), bersama tim pengembang Jejakku

Aplikasi trip organizer ini memudahkan traveler mendapatkan informasi lokasi wisata, harga tiket, dan transportasi yang dibutuhkan. Di dalamnya juga terdapat ajakan untuk menuntaskan objektif tertentu dan mengumpulkan poin.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Selain ketiga pemenang di atas, aplikasi Travis yang dirancang sebagai alat untuk memperoleh informasi mengenai berbagai obyek wisata di Indonesia turut terpilih menjadi juara favorit publik melalui hasil polling yang dikumpulkan dari media sosial dan situs internet The NextDev.

Mengenai The NextDev

The NextDev adalah tema untuk Digital Creative Indonesia Competition 2015 yang diselenggarkan operator seluler Telkomsel dalam rangka program tanggung jawab perusahaan mengembangkan ekosistem digital.

Dengan topik Smart City, aplikasi yang diikutsertakan dalam kompetisi harus dikembangkan dari permasalahan kota dan pemecahan solusinya melalui penerapan teknologi, baik berbasis website maupun mobile dengan plaform yang dikuasai, seperti iOS, Android, atau Windows Phone.

Keenam sub-tema untuk pengembangan aplikasi didesain untuk mendukung pemanfaatan teknologi broadband yang dapat digunakan untuk masyarakat luas.

Corpus ID : 149

Sumber :

http://tekno.kompas.com/read/2015/10/23/09055307/www.phonearena.com

KOMPAS.com — Banyak pengguna iOS yang mengeluhkan betapa cepat habisnya baterai perangkat mereka belakangan ini. Setelah diselidiki, ternyata "penjahat" dari masalah tersebut adalah aplikasi Facebook untuk iOS.

Masalah ini diakui langsung oleh Engineer Manager Facebook Ari Grant.

Melalui status Facebook-nya, sebagaimana KompasTekno rangkum dari PhoneArena, Jumat (23/10/2015), ia menginformasikan beberapa masalah yang membuat aplikasi Facebook versi iOS membuat baterai iPhone lebih cepat habis dari seharusnya.

Masalah pertama yang ditemukan diberi istilah "CPU Spin". Masalah ini ada di kode jaringan aplikasi tersebut.

Inti dari masalah satu ini, CPU merasa seakan ada sebuah tugas yang belum diselesaikan. Oleh karena itu, CPU akan terus bekerja sehingga membuat baterai lebih cepat habis.

"CPU Spin seperti seorang anak kecil di dalam mobil bertanya, 'Apakah kita sudah sampai? Apakah kita sudah sampai?', pertanyaan yang tidak membuat adanya peningkatan untuk mencapai tujuan," tulis Grant.

Facebook juga menemukan adanya masalah di manajemen sesi audio. Jika Anda menutup aplikasi Facebook setelah menonton video, sesi audio terkadang tetap berjalan secara diam-diam. Inilah yang membuat baterai terus disedot dayanya.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Facebook dikatakan sudah menyelesaikan masalah ini. Perusahaan jejaring sosial ini menjanjikan peningkatan performa baterai pada versi aplikasi terbarunya yang bakal tersedia hari ini.

"Kami meminta maaf terhadap ketidaknyamanan yang kami sebabkan. Kami akan terus berkomitmen untuk melanjutkan peningkatan penggunaan baterai di aplikasi dan Anda akan melihat peningkatan dalam versi yang dirilis hari ini," pungkas Grant.

Corpus ID : 150

Sumber :

http://tekno.kompas.com/read/2015/10/23/10372837/Instagram.Perkenalkan.Boomerang.Pembuat.Foto.Bergerak

KOMPAS.com - Instagran meluncurkan aplikasi terpisah yang diberi nama Boomerang. Aplikasi ini bisa dipakai untuk membuat Gambar bergerak (GIF).

GIF adalah format Gambar bergerak di internet yang terdiri atas serangkaian foto yang ditampilkan berurutan secara terus-menerus (loop), membuatnya seperti video mini dengan durasi pendek.

Dikutip KompasTekno dari blog resmi Instagram, Jumat (23/10/2015), Instagram bisa dipakai untuk memotret lima foto yang bisa dirangkai menjadi format GIF dan memutarnya.

"Boomerang mengambil serangkaian foto dan merangkainya menjadi video mini berkualitas tinggi yang bisa diputar maju-mundur," demikian tulis postingan resmi Instagram di blog-nya.

Sama dengan Instagram, pengguna bisa merekam foto dalam mode portrait (vertikal) atau lanskap (horisontal). Hasil Gambar GIF juga bisa di-share di akun Instagram dan disimpan di album kamera.

Google Play Store

Boomerang Instagram

Aplikasi ini merupakan hasil hackathon yang dilakukan Instagram pada Juli lalu. Boomerang menjadi aplikasi terpisah ketiga yang dirilis oleh Instagram. Sebelumnya jejaring sosial berbasis foto ini telah meluncurkan aplikasi foto time-lapse Hyperlapse dan aplikasi pembuat kolase foto, Layout.

Aplikasi lain yang memiliki fungsi serupa dengan Boomerang antara lain Google Photo dan Phhhoto yang telah melampaui 1 juta pengguna pada Maret lalu.

"Dengan Boomerang, kami tidak sabar untuk melihat hasil kreasi kalian," ujar Instagram.

Boomerang from Instagram bisa diunduh di Play Store lewat tautan berikut ini, dan di iOS melalui tautan berikut ini.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Corpus ID : 151

Sumber :

http://tekno.kompas.com/read/2015/10/28/13290027/Keseruan.Game.Duel.Otak.yang.Sedang.Digandrungi

JAKARTA, KOMPAS.com - Belakangan, mulai sering muncul Gambar pemindaian yang menunjukkan skor permainan "Duel Otak". Di Twitter, Gambar-Gambar serta komentar atas permaianan tersebut bisa dicek lewat tagar #duelotak.

Game adu kecerdasan yang baru dirilis di Indonesia pada pertengahan 2015 tersebut tampaknya tengah digandrungi. Apa serunya?

Secara garis besar, mekanisme bermain duel otak mirip kuis "Who Wants To Be a Millionaire" yang sempat populer di Indonesia dan dibawakan oleh presenter Tantowi Yahya.

Duel Otak merangsang kemampuan pikir pemain dengan pertanyaan-pertanyaan tentang pengetahuan umum pada kategori tertentu. Pemain bisa memilih kategori yang diinginkan.

Duel Otak

Duel Otak

Misalnya tentang musik, film, sejarah, agama, tumbuhan, hewan, kesehatan, buku, olahraga, teknologi hingga pengetahuan umum seputar dunia.

Permainan dibagi menjadi dua mode. Yakni mode "Klasik" dan "Taktis". Pada mode Klasik, pemain tak bisa memilih alat bantu untuk menjawab pertanyaan.

Sebaliknya, pada mode Taktis, pemain diperkenankan memilih alat bantu. Ada tiga alat pilihan yang tersedia, yakni "50:50", "Presentasi" dan "Dua Pilihan".

Nah yang paling membedakan Duel Otak dengan "Who Wants To Be a Millionaire", sekaligus yang paling menarik dari game ini adalah kemampuan menantang dan ditantang teman untuk adu kecerdasan.

Duel Otak

Duel Otak

Pemain bisa mengundang teman untuk janjian berduel di waktu-waktu tertentu. Perekrutan teman bisa melalui nama user yang didaftarkan pada game atau melalui Facebook. Pemain juga bisa memilih lawan acak yang akan dipilihkan oleh Duel Otak.

Setelah menerima tantangan atau menantang teman, pemain bisa mulai beradu. Ada total 18 pertanyaan yang diadukan. Tiap tiga pertanyaan, pemain secara bergantian memilih kategori pertanyaan.

Misalnya awalnya A memilih pertanyaan tentang musik. Akan ada tiga soal pilihan ganda yang diberikan seputar kategori tersebut.

Setelah tiga pertanyaan dijawab, kedua pemain dapat saling melihat skor satu sama lain. Selanjutnya, pemain B yang menentukan kategori. Begitu seterusnya hingga enam ronde.

Duel Otak

Duel Otak

Di akhir ronde, pemain akan melihat siapa yang menang dan siapa yang kalah. Pemenang akan diberikan piala sebagai penghargaan. Keberhasilan memenangkan inilah yang kerap dipindai pemain untuk dibagi ke media sosial sebagai kebanggaan tersendiri.

Walau Duel Otak berpotensi membuat pemain kecanduan, game ini menyediakan kemampuan agar pemain bisa rehat dari peraduan. Pemain diberi keleluasaan bermain pada waktu senggang.

Di tengah permainan, jika ada hal yang harus diselesaikan, pemain bisa meninggalkan permainan dan melanjutkannya nanti. Asalkan waktu berhenti tak lebih dari 23 jam. Jika melebihi jangka waktu itu, pemain akan otomatis gagal.

Seperti game-game pada umumnya, Duel Otak juga menyediakan fitur berbayar untuk kemampuan permainan yang lebih mumpuni. Pemain bisa meningkatkan kemampuan game dengan biaya tertentu agar dapat memodifikasi avatar.

Pemain juga dapat membeli koin sebanyak yang diinginkan dengan harga tertentu. Koin tersebut berfungsi untuk menggunakan alat bantu. Sebab, setiap penjajalan alat bantu, pemain harus mengeluarkan lima koin.

Walau namanya "Duel Otak", permainan ini tak dikembangkan oleh pengembang lokal. Permaianan ini hanya disesuaikan ke dalam Bahasa Indonesia.

Aslinya, permainan bertajuk "Quiz Clash" yang dibuat oleh FEO Media AB dari Swedia. Game tersebut sudah berekspansi ke 21 negara.

Untuk mengunduh game ini, bisa melalui Google Play Store, Apple App Store atau Windows Phone Store.

Corpus ID : 152

Sumber :

http://tekno.kompas.com/read/2015/10/21/08310057/Ketahuan.Maling.Data.Ratusan.Aplikasi.Dihapus.Apple

KOMPAS.com - Apple telah menghapus sejumlah aplikasi dari App Store-nya setelah riset yang dilakukan oleh firma keamanan menunjukkan bahwa ratusan aplikasi iOS ternyata bisa mengakses data pribadi penggunanya.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Dikutip KompasTekno dari Mashable, Selasa (20/10/2015), peneliti di firma keamanan SourceDNA mengatakan mereka menemukan sebanyak 256 aplikasi yang melanggar panduan keamanan yang ditetapkan oleh Apple.

Pelanggaran tersebut berupa penggunaan software yang bisa dipakai untuk mengumpulkan data-data sensitif yang bersifat pribadi, termasuk email dan nomor identitas perangkat mereka.

Berasal dari China

SourceDNA tidak merinci satu per satu nama-nama aplikasi tersebut, namun mengatakan kebanyakan adalah buatan pengembang dari China.

Sumber dari permasalahn tersebut menurut SourceDNA adalah software development kit (SDK) yang dimiliki oleh perusahaan iklan China, Youmi.

SDK iklan Youmi jika dipakai oleh developer bisa mengambil data seperti alamat email, nomor serial perangkat dan daftar aplikasi yang diunduh pengguna iOS.

"Data-data yang dikumpulkan oleh SDK tersebut kemudian diunggah ke server milik Youmi sendiri," demikian pernyataan SourceDNA.

Diunduh Sejuta Kali

Dalam pernyataannya, Apple mengaku telah menemukan sejumlah aplikasi yang menggunakan SDK yang dimaksud, yang memang telah melanggar aturan keamanan yang telah ditetapkan Apple.

"Aplikasi yang menggunakan SDK Youmi akan dihapus dari App Store, aplikasi baru yang di-submit ke App Store dengan SDK ini juga akan ditolak," demikian pernyataan Apple.

SourceDNA mengatakan 256 aplikasi tersebut telah diunduh sebanyak 1 juta orang. Walau angka tersebut tergolong kecil untuk ukuran App Store, mereka takut langkah yang sama bakal dipakai pihak-pihak lain.

Corpus ID : 153

Sumber :

http://tekno.kompas.com/read/2015/10/27/19490027/layanan.transjakarta.go-busway.sudah.ada.di.aplikasi.go-jek

KOMPAS.com — Ada layanan anyar yang muncul di aplikasi Go-Jek, yakni Go-Busway. Ikon untuk layanan ini bisa ditemukan di menu utama aplikasi, berdekatan dengan Go-Box dan Go-Mart yang juga terhitung baru.

Menurut pantauan KompasTekno, Selasa (27/10/2015), Go-Busway adalah layanan transportasi ojek untuk mengantar pengguna ke selter bus transjakarta yang diinginkan.

Cara memesan ojek melalui fitur Go-Busway ini tampak mirip dengan layanan lain dari Go-Jek yang sama-sama memanfaatkan jasa pengendara sepeda motor.

Perbedaannya hanya terletak pada awal pemesanan, yaitu pengguna diminta untuk langsung melakukan pencarian nama selter melalui kotak search. Prosesnya setelah itu sama seperti Go-Ride (pemesanan ojek biasa).

Pengguna bisa mengetikkan nama selter yang dituju dalam kotak pencarian. Fitur ini khusus ditujukan untuk mencari selter, bukan lokasi lain. Sebuah daftar lengkap berisi semua selter bus transjakarta dan lokasinya akan ditampilkan untuk membantu pengguna.

Oik Yusuf/ Kompas.com

Jadwal keberangkatan bus TransJakarta di dalam layanan Go-Busway

Alternatifnya, pengguna bisa langsung memilih selter lewat ikon berwarna biru yang muncul di peta.

Begitu menemukan selter tujuan, aplikasi akan menampilkan jadwal keberangkatan bus dari selter yang bersangkutan menuju selter berikut dan sebaliknya.

Go-Busway turut menampilkan ikon berwarna hijau muda yang berseliweran di jalanan dalam peta. Ikon hijau ini agaknya mewakili posisi dari masing-masing bus transjakarta yang dilengkapi GPS.

Dari pijat sampai bus transjakarta

Pada Juni lalu memang sudah dicanangkan bahwa bus-bus transjakarta akan dilengkapi GPS untuk menunjang operasional layanan Go-Busway.

Waktu ketersediaan layanan ini yang muncul pada bulan Oktober pun sesuai dengan jadwal yang ditetapkan sebelumnya.

Dari pantauan Kompas Tekno, layanan Go-Busway baru tersedia di aplikasi Go-Jek untuk platform Android. Sebelum bisa menggunakan layanan Go-Busway, pengguna mesti memperbarui aplikasi Go-Jek ke versi 1.0.62. Belum diketahui kapan layanan Go-Busway akan menyusul hadir di aplikasi Go-Jek versi iOS.

Pihak Go-Jek sendiri diketahui akan menggelar acara yang berhubungan dengan bus transjakarta di Balai Kota DKI Jakarta, Jalan Medan Merdeka Selatan, besok siang.

Go-Busway menambah panjang layanan yang dihadirkan oleh Go-Jek. Sebelumnya, perusahaan yang dibentuk oleh Nadiem Makarim ini telah meluncurkan Go-Mart, layanan pembelian barang; Go-Clean, layanan bersih-bersih rumah; dan Go-Massage, layanan pemanggil jasa pijat.

Corpus ID : 154

Sumber :

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

http://tekno.kompas.com/read/2015/11/01/08380037/mahasiswa.bandung.kembangkan.game.jomblo.indonesia.

BANDUNG, KOMPAS.com - Tim Hash Tag Telkom University Bandung mengembangkan aplikasi Jombie yang berarti "Jomblo Indonesia".

Aplikasi ini berhasil meraih juara 3 pada Kategori Pengembangan Aplikasi Games dalam Pagelaran Mahasiswa Nasional Bidang Teknologi Informasi dan Komunikasi (Gemastik) 8 Tahun 2015.

Menurut Bagus, salah satu anggota tim, Jombie merupakan permainan (game) berbasis Android yang mengenalkan tradisi, adat, dan budaya pernikahan di Indonesia. Game ini terdiri dari beberapa mini-game sebagai tantangannya.

“Siapa yang dapat menyelesaikannya, bisa berfoto menggunakan fitur Augmented Reality baju pernikahan sesuai dengan daerahnya,” ujar Bagus dalam keterangan tertulis yang diterima Kompas.com, Sabtu (31/10/2015).

Game Jombie sebenarnya merupakan satu dari sekian banyak karya akhir mahasiswa Prodi D3 Informatika FIT yang wajib untuk diikutsertakan pada Kompetisi Gemastik 8 Tahun 2015.

Karya ini pada dasarnya bukanlah karya yang dipersiapkan untuk mengikuti kompetisi. Tetapi kerja keras dari Tim HashTag yang terdiri dari Miftahul Bagus Pranoto, Ranita Windriani, dan Yuni Kumaeroh berhasil meraih Juara 3 dalam Kategori Pengembangan Aplikasi Games.

Bagus mengungkapkan bahwa mereka tidak menyangka bisa meraih tiga besar dalam kategori tersebut. “Saran bagi teman-teman, mencari ide tidak perlu jauh-jauh yang penting unik dan menarik,” ujarnya.

Kompetisi yang mengusung Kreativitas dan Inovasi Pemuda Menuju Masyarakat Indonesia Berdikari tersebut diselenggarakan di Auditorium Graha Sabha Pramana Universitas Gadjah Mada Yogyakarta dari Senin-Rabu, tanggal 26 – 28 Oktober 2015.

Sementara itu, untuk juara Umum Gemastik 8 2015 diraih oleh Insititut Teknologi Bandung dengan raihan 3 emas dan 3 perunggu. Kemudian diikuti oleh Universitas Gadjah Mada dengan raihan 2 emas, 2 perak, dan 2 perunggu, Universitas Indonesia dengan raihan 1 emas dan 3 perak, Institut Pertanian Bogor dengan raihan 1 emas dan 1 perak, serta Politeknik Caltex Riau dan Politeknik Negeri Jakarta yang sama-sama meraih 1 emas.

Direktur Kemahasiswaan Universitas Gadjah Mada, Dr. Drs. Senawi, MP menuturkan, Gemastik 8 Tahun 2015 diikuti oleh 4.087 tim yang terdiri dari 166 perguruan tinggi seluruh Indonesia. Setelah melakukan tahap seleksi, finalis lomba terdiri dari 101 tim dari 27 perguran tinggi. “Kondisi ini lebih baik dari Gemastik 7 tahun lalu yang hanya meloloskan 21 tim,” ujar Senawi.

Corpus ID : 155

Sumber :

http://tekno.kompas.com/read/2015/10/20/11410057/Membangun.Smart.City.Menggunakan.Aplikasi

JAKARTA, KOMPAS.com - Konsep smart city tidak akan terwujud tanpa adanya aplikasi penunjang yang mumpuni. Aplikasi-aplikasi tersebut penting karena merupakan piranti yang membuat kinerja berbagai kegiatan perkotaan menjadi lebih efisien sehingga layak menyandang julukan kota pintar.

Katakanlah berupa aplikasi manjemen kesehatan atau sekadar aplikasi pencari tebengan berangkat kantor. Menggunakan aplikasi tersebut, warga kota bisa mendapatkan informasi soal layanan kesehatan, dokter yang bisa ditemui, dan lain hal yang terkait.

Sedangkan menggunakan aplikasi tebengan, pengguna bisa menemukan transportasi alternatif dan membantu mengurangi kemacetan. Misalnya, ketika ingin berangkat ke kantor cukup mengecek aplikasi untuk menemukan orang yang searah dan bersedia memberikan tumpangan.

Dengan demikian, pengguna tidak perlu membawa kendaraan sendiri. Jika semakin banyak yang melakukannya, tentu jumlah kendaraan yang beredar di jalan akan berkurang sehingga bisa menjadi solusi kemacetan.

Dua aplikasi itu hanya contoh saja, yang sebenarnya juga sudah tercakup dalam deretan aplikasi finalis The NextDev. Ini adalah ajang perlombaan yang diselenggarakan Telkomsel untuk menjaring berbagai aplikasi pendukung smart city.

Bicara soal lomba aplikasi tersebut, sebenarnya masih banyak aplikasi lain. Ketika lomba baru dibuka saja tercatat ada lebih dari 400 aplikasi yang mengusung konsep smart city.

Ajang The NextDev kini sudah mendapatkan 20 aplikasi finalis di enam sub tema yang disyaratkan. Nantinya, tiga tim terbaik akan mendapatkan hadiah berupa dukungan 5M, yaitu Market Access, Marketing Publicity, Mentoring, Management Tri dan Money.

20 finalis dalam enam subtema tersebut adalah:

e-Education: Sekolah Koding (Hilman Ramadhan – Makassar), Rumah Sinau (Abdul Basir – Jakarta), Sarjana.co.id (Setra Yappi – Surabaya).

e-Government: iWatchYou (Arrival D. Sentosa – Bandung), GandengTangan (Nur Roni D. – Jakarta), kolek.in (Danu Pujiachiri), HICO (Doni Prabowo – Yogyakarta), Tilang.in (Laurensius Faleddo – Semarang), SmartGorontalo (Dea Venditama – Gorontalo)

e-Transportation: Nebengers (Andreas Aditya S. – Jakarta)

e-Tourism: BATAMciti (Yolanda Agustina – Batam), Lenscoop (M. Badrullami – Surabaya), Pontinesia (Adhitya T. Nugraha – Pontianak), Jejakku (Ahimsa D. Afrizal – Surabaya), Travis (Diagnosa Fenomena – Surabaya)

e-Health: Healink (Fikri W. – Yogyakarta), Bloodjob (Martin Luter – Jakarta), QCare (Satria D. Raharjo).

e-UKM: ERZ4P (Risa Hisamuddin – Denpasar), Appsterize (Faizal Hitobeli – Jakarta)

Corpus ID : 156

Sumber :

http://tekno.kompas.com/read/2015/10/23/15254777/Menkominfo.Ingin.1.000.NextDev.dan.Coding.Masuk.Kurikulum

JAKARTA, KOMPAS.com - Menteri Komunikasi dan Informatika Rudiantara menyoroti pentingnya kompetisi aplikasi dalam upaya menumbuhkan ekonomi digital di Indonesia. Terutama dengan merangsang munculnya usaha rintisan digital atau startup yang berkualitas.

Aplikasi yang dihasilkan oleh startup berkualitas bakal banyak dipakai dan membuat berbagai lini kehidupan menjadi lebih praktis. Bahkan bisa jadi penguat ekonomi kecil, misalnya dengan membuat aplikasi yang menghubungkan petani langsung ke pembeli produk taninya.

"Kita perlu 1.000 atau lebih NextDev dan kompetisi yang sejenis ini. Coding juga harus masuk kurikulum. Kalau mau maju harus seperti ini," ujar Rudiantara di hadapan 20 tim finalis Telkomsel NextDev, di Gedung KeMenkominfo, Jumat (23/10/2015).

"Nanti juga akan ada hackathon untuk membuat aplikasi dalam 24 jam. Pesertanya banyak, sudah sekitar 1.800 orang dan bisa jadi yang terbesar di dunia," imbuhnya.

Sementara itu Kemenkomifo, menurut Rudiantara, bekerja menyediakan infrastruktur yang mendukung. Salah satunya adalah pembangunan jaringan internet cepat 4G LTE di seluruh Indonesia.

Dia juga mengakui bahwa saat ini, pemerintah belum bisa bergerak selincah negara tetangga dalam hal memberikan dukungan untuk startup. Pendanaan startup di Tanah Air masih berasal dari swasta.

Sementara itu Singapura dan Brunei Darrussalam sudah berani mengucurkan dana untuk menumbuhkan startup.

"Pemerintah Singapura mendukung startup. Kemudian kalau di Brunei, mereka menawarkan 50.000 dollar Brunei (sekitar Rp 490 juta) untuk startup," terang Chief RA, sapaan akrabnya.

"Saya berharap temen-temen di sini gak berhenti di NextDev, ikuti lagi yang lainnya apapun itu. Pemenang di sini (NextDev) bolehlah coba-coba di Brunei, asal eligible dan bisa dipakai di sana," pungkasnya.

Corpus ID : 157

Sumber :

http://tekno.kompas.com/read/2015/10/29/14300077/Microsoft.Kucurkan.Rp.13.Miliar.untuk.Latih.Guru.Pakai.Office.365

KOMPAS.com - Awal September 2015, Microsoft dan Persatuan Guru Republik Indonesia (PGRI) telah menandatangani sebuah kesepakatan. Isinya menyatakan Microsoft akan memberikan akses layanan Office 365 gratis ke 3,5 juta guru di Indonesia.

Untuk memaksimalkan pemanfaatan Office 365 guna menambah kapasitas guru dan peserta didik, Microsoft mengucurkan dana investasi tambahan sebesar 1 juta dollar AS atau setara Rp 13,6 miliar.

Dana itu digunakan untuk memberikan pelatihan pada guru-guru agar dapat memahami, menyebarkan dan mengoptimalkan Office 365 dalam proses belajar-mengajar. Terutama terkait penggunaan teknologi berbasis komputasi awan (cloud) pada layanan tersebut.

"Teknologi tak berarti apa-apa jika tidak digunakan. Inilah sebabnya kami menyumbangkan lagi dana 1 juta dollar AS untuk memberdayakan dan melatih guru-guru Indonesia," kata Presiden Direktur Microsoft Indonesia Andreas Diantoro, dalam pidatonya di Silicon Valley, sebagaimana tertera pada keterangan pers yang dihimpun KompasTekno, Kamis (29/10/2015).

Microsoft tak menampik bahwa pemberian layanan gratis tersebut dilandaskan kepentingan perusahaan. Dengan pendidikan memadai, Microsoft mengklaim ingin melihat talenta-talenta yang bisa dihasilkan untuk masa depan.

"Ini bagian dari misi kami untuk memberdayakan semua potensi di planet ini. Microsoft bangga saat ini memiliki pakar TI dan coders berbakat dari Indonesia yang bekerja untuk kami di seluruh dunia," kata Andreas.

Diketahui, Office 365 adalah layanan Microsoft dengan paket berlangganan yang mengakomodir instalasi aplikasi Office lengkap. Semuanya dapat diinstal pada beberapa perangkat, termasuk PC, Mac, Tabelt Android, ponsel Android, iPad, dan iPhone.

Dengan begitu, pembuatan, pengeditan dan sinkronisasi dokumen lintas perangkat bakal lebih mudah. Selain itu, di Office 365 pengguna mendapatkan layanan tambahan. Seperti penyimpanan online dengan OneDrive dan menit Skype untuk penggunaan di rumah. Semuanya bertujuan untuk menambah produktivitas dalam pendidikan dan ekonomi.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Corpus ID : 158

Sumber :

http://tekno.kompas.com/read/2015/11/03/10460047/Microsoft.Patok.Tanggal.Pensiun.Windows.7.dan.Windows.8

KOMPAS.com - Microsoft memasang tenggat waktu untuk dua sistem operasi lama mereka, WIndows 7 dan WIndows 8.

Menurut Microsoft, kedua sistem operasinya itu tidak akan terpasang di laptop atau PC baru dalam jangka satu tahun lagi.

Dikutip KompasTekno dari Digital Trends, Selasa (3/11/2015), Microsoft baru saja merilis fact sheet lifecycle yang menjadi panduan bagi vendor pembuat PC/laptop (OEM).

Dalam fact sheet tersebut dijelaskan saat ini vendor tetap bisa menjual perangkat dengan sistem operasi Microsoft lama, namun hingga 31 Oktober 2016, semua perangkat baru yang dijual harus sudah menggunakan Windows 10.

"Saat Microsoft meluncurkan versi baru Windows, kami tetap mengizinkan OEM menjual PC dengan sistem operasi versi lama hingga dua tahun setelah tanggal peluncuran versi barunya, kecuali ada perubahan," demikian tulis Microsoft.

Menurut Microsoft, hingga kini jumlah pengguna Windows 10 telah mencapai 120 juta orang di seluruh dunia. Selain itu, hampir 28 persen sistem Windows yang ada di dunia telah menjalankan Windows 10, alih-alih sistem operasi versi lawas.

Walau Windows 7 tidak akan dijual secara pre-installed lagi setelah 2016, namun update untuk sistem operasi tersebut akan terus diberikan oleh Microsoft.

Menurut Microsoft, dukungan update untuk Windows 7 akan diperpanjang hingga tahun 2020. Sementara untuk sistem operasi Windows 8 yang habis pada Januari 2018, akan diberi masa perpanjangan hingga 2023.

Corpus ID : 159

Sumber :

http://tekno.kompas.com/read/2015/10/30/21080027/Microsoft.Sediakan.Launcher.Android.Mirip.iOS.9

KOMPAS.com - Microsoft akhirnya merilis "launcher" Android buatannya, Arrow. Pengguna smartphone Android kini bisa langsung mengunduhnya melalui toko digital Play Store.

Launcher Arrow dikatakan mirip dengan iOS 9 buatan Apple. Salah satu kemiripannya, launcher Arrow merekomendasikan aplikasi dan daftar kontak kepada pengguna dengan cara yang sama dengan fitur pelayanan Spotlight di iOS 9.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Kemiripan lainnya, di layar "Recent" Arrow, menampilkan kemampuan untuk mengakses aplikasi, file, foto, dan daftar kontak yang biasanya dibutuhkan cepat.

Hal-hal tersebut diperlihatkan berdasarkan tingkah laku pengguna. iOS 9 sendiri juga meneliti penggunaan ponsel atau aplikasi apa yang dipakai setiap harinya.

Launcher Arrow ini sendiri bertujuan untuk menyederhanakan tampilan dengan mengedepankan aplikasi yang paling sering digunakan.

Arrow Launcher memiliki tiga layar utama. Sebenarnya, masih ada dua layar yang "disembunyikan". Melalui menu Edit Page, pengguna bisa menampilkan layar "Notes & Reminders" dan "Widget". Selain dua layar itu, pengguna tidak bisa menambah layar baru lagi.

Layar pertama diletakkan di bagian paling kiri dan dinamakan "Recent". Seperti yang dituliskan di atas, layar ini menampilkan aplikasi, file, foto, dan daftar kontak yang paling sering digunakan. Ada juga icon-icon aplikasi yang paling terakhir digunakan.

Layar utama dari launcher ini terletak di bagian tengah dan dinamakan Apps. Layar itu memperlihatkan semua aplikasi yang dimiliki pengguna. Uniknya, peletakkan aplikasi tersebut dimulai dari yang paling sering digunakan.

Sedangkan, layar ketiga dinamakan People. Yang satu ini, berisikan aktivitas terakhir dengan orang tertentu. Sebagai contoh, layar ini menampilkan aktivitas berkirim e-mail dengan satu orang tertentu.

Sama seperti tampilan Android pada umumnya, Microsoft menaruh sebuah dock khusus untuk aplikasi. Di dock tersebut, pengguna bisa meletakkan shortcut ke aplikasi yang sering digunakan.

Di bagian tengah bawah, terdapat simbol yang bisa membawa pengguna ke App Drawer.

Arrow sebenarnya sudah ada sejak Juli lalu. Akan tetapi, ia masih hadir dalam bentuk program "alpha" dan hanya dibatasi penggunaannya di kalangan tertentu untuk pengujian.

Tidak dituliskan perangkat apa saja yang kompatibel dengan launcher ini. Itu artinya, bisa saja semua perangkat Android yang ada saat ini bisa mendukung aplikasi tersebut.

Meski begitu, Arrow hanya bisa dijalankan di Android versi 4.0.3 atau lebih baru.

Aplikasi Arrow saat ini sudah bisa diunduh di Play Store. Besarnya file instalasi aplikasi tersebut tidaklah terlalu besar, hanya sekitar 4,29 MB saja.

Jika Anda bosan dengan tampilan smartphone Android, silahkan coba saja launcher yang satu ini.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Corpus ID : 160

Sumber :

http://tekno.kompas.com/read/2015/10/21/15110067/.OK.Google.Ditendang.dari.Chrome

KOMPAS.com - Pekan ini, Google merilis Chrome versi 46. Tak ada penambahan fitur berarti pada layanan browser tersebut. Sebaliknya, Chrome 46 justru memangkas fitur populer "OK Google" untuk beberapa PC.

Yakni pada Chrome untuk sistem operasi Windows, Mac OS X, dan Linux. Adapun pengguna perangkat Android dan Chromebook masih bisa menjajal fitur tersebut.

Artinya, sebagian besar pengguna Chrome di PC sudah tak bisa lagi mengandalkan komando suara untuk melakukan pencarian cepat. Apa alasan Google menendang fitur tersebut?

Dilaporkan Forbes dan dihimpun KompasTekno, Rabu (21/10/2015), "OK Google" tak dimanfaatkan secara signifikan oleh pengguna Chrome di platform Windows, Mac OS X, dan Linux. Menurut Google, fitur tersebut akan mubazir jika tetap eksis namun tak terpakai.

Pun begitu, bagi sebagian penggguna Windows/Mac OS X/Linux yang selama ini menggunakan "OK Google" dan telah nyaman dengannya, masih ada cara yang bisa ditempuh.

Pengguna bisa secara manual membuka situs pencarian Google dari tab baru. Dari situ, fitur "OK Google" tetap terpatri di kotak pencarian.

Pengguna cukup menekan ikon mikropon lalu memerintah situs pencarian tersebut dengan komando suara.

Diketahui, Google pertama kali meluncurkan pencarian dengan "OK Google" untuk Chrome Pada November 2013. Setelah dua tahun terbenam, fitur tersebut akhirnya harus tersingkirkan.

Corpus ID : 161

Sumber :

http://tekno.kompas.com/read/2015/10/23/08070047/OS.Baru.iPhone.Tampilkan.Emoticon.Kabah.dan.Unicorn

KOMPAS.com - Apple akhirnya merilis iOS 9.1 untuk publik. Salah satu fitur terbarunya adalah tambahan lebih dari 150 karakter emoticon, termasuk kabah, unicorn dan jari tengah.

iOS 9.1 adalah update besar yang pertama kali dilakukan Apple sejak mereka meluncurkan iOS 9 pada September lalu. Versi ini sempat dirilis lima kali dalam bentuk beta untuk developer serta publik.

Hari ini, Kamis (22/10/2015) versi teranyar sistem operasi tersebut sudah bisa diunduh semua pemilik iPhone dan iPad. Apple juga telah merilis Xcode 7.1 dan tvOS untuk TV buatannya

Dilansir KompasTekno dari Mac Rumors, pengguna yang sudah mengunduh akan menemukan beberapa perubahan di perangkatnya. Di antaranya adalah dukungan untuk Unicode 7 dan 8.

Otomatis, dukungan tersebut membuat emotican baru seperti kabah, masjid, unicorn hingga acungan jari tengah sudah bisa terlihat dan dibaca.

Pembaruan lainnya adalah soal aplikasi Live Photo yang kini bisa mengetahui bedanya saat iPhone sedang diangkat atau saat sedang diletakkan.

iOS 9.1 dibekali dengan sejumlah fitur lin dan kecocokan dengan update produk Apple yang baru, misalnya Apple TV generasi 4, iPad Pro dan iPhone.

Bug atau isu yang sebelumnya terjadi dan dikeluhkan pengguna ponsel pun sudah hilang. Apple juga meningkatkan kinerja dan keamanan dalam sistem operasi iOS 9.1.

Corpus ID : 162

Sumber :

http://tekno.kompas.com/read/2015/10/22/14010037/Seperti.Apa.Karya.20.Aplikasi.Finalis.NextDev.

JAKARTA, KOMPAS.com - Kompetisi Telkomsel NextDev sudah mencapai tahap penjurian dengan total 20 aplikasi finalis. Seperti apa aplikasi-aplikasi yang sedang memperebutkan gelar juara itu?

Pada Rabu (20/10/2015) lalu, selama seharian penuh, masing-masing tim pembuat aplikasi diminta untuk mempresentasikan atau pitching ide-ide mereka di hadapan juri dan disaksikan oleh para calon investor.

Tahap ini untuk menentukan siapakah tiga tim terbaik yang bakal mendapatkan hadiah uang tunai dan kunjungan belajar ke pelaku industri telekomunikasi besar di luar negeri. Selain itu ada juga pemenang favorit pilihan publik yang bisa Anda pilih melalui tautan ini.

Nah, sebelum memilih mana tim favorit Anda, berikut ini KompasTekno sajikan sekilas pandang soal karya-karya 20 finalis tersebut.

1. Batamciti

Ini merupakan portal web dan aplikasi berisi informasi mengenai kota Batam. Di dalamnya, pengguna bisa menemukan hal-hal seperti info wisata hingga daftar hotel yang dapat dipilih untuk menginap.

2. ERZ4P

Aplikasi ini berfungsi untuk memperudah proses pencatatan data, laporan keuangan, promosi dan transaksi sebuah perusahaan. ERZ4P dibuat untuk usaha skala kecil menengah.

3. Smartgorontalo

Smartgorontalo dibuat sebagai aplikasi penampil informasi mengenai kota Gorontalo melalui peta dan data. Penggunanya bisa mencari lokasi di daerah kabupaten hingga melihat kontur jalan menuju ke sana.

4. Jejakku

Aplikasi trip organizer ini bisa memudahkan traveler mendapatkan informasi lokasi wisata, harga tiket dan transportasi yang dibutuhkan. Selain itu, di dalamnya terdapat ajakan untuk menuntaskan objektif tertentu dan mengumpulkan poin.

5. SekolahKoding

Ini adalah seuah situs yang bisa diakses siapapun untuk belajar mengenai bahasa pemrograman dan desain, khususnya untuk pembuatan web.

6. Tilang.in

Aplikasi ini dibuat dengan tujuan mengkampanyekan tertib lalu lintas. Pengguna bisa melaporkan kejadian pelanggaran lalu lintas langsung melalui smartphone Android.

7. Pontinesia

Ini merupakan media sosial bertema traveling yang bisa diakses melalui web dan mobile web. Di dalamnya, pengguna bisa menemukan berbagai tema terkait wisata alam, budaya serta kuliner kota Pontianak.

8. Travis

Aplikasi ini dirancang sebagai alat untuk memperoleh informasi mengenai berbagai objek wisata di Indonesia.

9. Qcare

Ini merupakan aplikasi ynag membantu pemanggilan ambulans untuk penanganan kondisi darurat. Saat ini baru bisa dipakai di kota Surabaya saja.

10. Healink

Aplikasi mobile ini dibuat sebagai sistem pendukung untuk layanan kesehatan, seperti rumah sakit, klinik dan sejenisnya.

11. GandengTangan

Dibuat sebagai wadah kolaborasi untuk mempertemukan pemilik usaha sosial dengan publik yang ingin membantu memberikan pinjaman.

12. Kolek.in

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Aplikasi ini berfungsi untuk memudahkan penggunanya melakukan survei di lapangan atau membuat kuesioner secara online.

13. Hybrid Controller

Ini merupakan solusi yang terdiri dari hardware dan software dan digunakan untuk mengendalikan peralatan elektronik. Pengguna bisa mengendalikannya melalui ponsel dengan memanfaatkan jaringan internet.

14. FirstBlood

Apliksi Android ini dibuat untuk membantu memudahkan masyarakat yang ingin ikut serta dalam kegiatan donor darah.

15. Lenscoop

Pengguna bisa memakai aplikasi ini untuk membuat foto panorama 360 derajat atau virtual reality.

16. RumahSinau

Merupakan platform belajar yang berguna mempertemukan pemilik ruang kosong dengan orang yang membutuhkannya untuk kegiatan belajar mengajar atau les.

17. iWatchYou

Adalah sebuah aplikasi Client Server yang bisa dipakai memantau aktivitas anak saat menggunakan komputer dan internet.

18. Nebengers

Awalnya ini merupakan sebuah komunitas yang mewadahi kegiatan memberi dan mencari tumpangan. Selanjutnya komunitas dunia maya tersebut pun dikembangkan menjadi bentuk aplikasi Nebengers ini.

19. Appsterize

Fungsinya untuk membantu usaha kecil dan menengah dalam upaya membuat aplikasi mobile mereka.

20. Sarjana.co.id

Ini merupakan portal pendidikan tinggi yang mendata seluruh institusi pendidikan. Tujuannya untuk membantu anak-anak yang baru lulus SMA agar lebih mudah memilih universitas lanjutannya.

Corpus ID : 163

Sumber :

http://tekno.kompas.com/read/2015/11/01/10190077/snapseed.kini.bisa.olah.foto.raw.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

KOMPAS.com - Aplikasi pengedit foto Snapseed kini bisa dipakai untuk mengutak-atik foto berformat RAW. Tambahan fitur tersebut benar-benar memperkuat aplikasi milik Google ini sebagai solusi editing mobile terlengkap.

Salah satu keunggulan Snapseed adalah berbagai fiturnya yang mudah digunakan, mencakup cukup banyak kebutuhan editing dan bisa diunduh gratis tanpa ada embel-embel in app purchase.

Dilansir KompasTekno dari PhoneArena, Sabtu (31/10/2015), Snapseed versi 2.1 sudah bisa mengedit format RAW. Ini merupakan format yang populer digunakan oleh fotografer maupun penggemar fotografi karena tidak mengalami kompresi.

Phone Arena

Contoh perbandingan hasil edit foto jpg (atas) dengan raw (bawah)

"Foto berformat RAW menyimpan berbagai data orisinil yang ditangkap oleh kamera. Dengan demikian, pengguna bisa mengeditnya dengan cara-cara yang mustahil diterapkan pada file JPG, misalnya memulihkan detil yang rusak karena highlight," jelas teknisi Google Anton Daupert.

Sesuai dengan penjelasannya, foto berformat RAW adalah Gambar yang dihasilkan langsung dari sensor kamera tanpa mengalami pemrosesan. Sehingga datanya utuh dan bisa diedit dengan bebas, memiliki banyak detil dan ukuran data yang lebih besar bila dibandingkan dengan JPG.

Bila menggunakan kamera ponsel, foto RAW akan disimpan dalam format DNG dengan kelebihan yang sama. Kekurangannya adalah biasanya format ini tidak terbaca dan malah sulit diedit menggunakan ponsel.

Sebelum adanya dukungan RAW pada Snapseed, pengguna yang ingin mendapatkan foto terbaik justru kerepotan karena mesti membukanya menggunakan software dekstop seperti Adobe Photoshop atau Lightroom.

Corpus ID : 164

Sumber :

http://tekno.kompas.com/read/2015/11/03/11160017/Snowden.Rekomendasikan.Aplikasi.Chatting.Teraman.di.Android

KOMPAS.com - Aplikasi chatting bertajuk "Signal for Android" resmi diluncurkan awal November 2015 ini. Mantan pegawai Badan Keamanan Nasional Amerika Serikat (NSA) Edward Snowden menganjurkan masyarakat untuk menggunakan aplikasi tersebut.

Menurut dia, Signal for Android memiliki keamanan yang lebih mumpuni untuk menghadang serangan peretas. Aplikasi tersebut diklaim merupakan kombinasi dari aplikasi TextSecure dan RedPhone.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Dilaporkan Engadget dan dihimpun KompasTekno, Selasa (3/11/2015), pengguna Signal for Android dapat lebih aman saat saling bertukar pesan dan menelepon seseorang. Tak ada peretasan informasi pesan ataupun penyadapan suara.

Setidaknya itu yang digembar-gemborkan aplikasi tersebut dan diamini Snowden. "Saya menggunakan Signal setiap hari. #notesforFBI (Catatan: mereka sudah tahu," Snowden menuliskan di akun Twitternya @Snowden.

Signal for Android bisa diaktivasi menggunakan nomor telepon seluler. Tak ada pendaftaran terpisah dengan nama pengguna, kata sandi atau PIN. Pada dasarnnya, mekanisme tersebut sama dengan WhatsApp.

"Kami tak bisa mendengar percakapanmu atau membaca pesanmu, tak ada yang bisa. Apapun di Signal for Android dienkripsi terus menerus, dan dengan susah payah digodok untuk menjamin keamanan berkomunikasi," begitu tertulis pada situs resmi Open Whisper Systems (OWS) yang menggodok aplikasi tersebut.

Diketahui, OWS adalah organisasi non-profit yang proyek-proyek pengembangan produknya didukung oleh donasi komunitas. Aplikasi Signal for Android gratis dan tak berisi iklan.

Untuk mencoba aplikasi tersebut, bisa diunduh melalui Google Play Store.

Corpus ID : 165

Sumber :

http://tekno.kompas.com/read/2015/11/04/08470027/Tombol.Bintang.Twitter.Berubah.Jadi.Hati.Pengguna.Marah

KOMPAS.com - Twitter memberikan kejutan kepada penggunanya pada Rabu (4/11/2015) dini hari dengan mengganti tombol Favorite ikonik berlogo bintang, menjadi tombol Like dengan logo hati.

Twitter Indonesia pun mencuitkan keputusan tersebut pada Rabu pagi. "Hari ini kami memperkenalkan ikon hati (likes) untuk Twitter dan @vine" tulis @TwitterID disertai tautan penjelasan di blog Twitter.

Menurut Twitter dalam postingan di blog resminya, pergantian itu dilakukan untuk mempromosikan simbol yang lebih universal yang bisa diterima oleh semua bahasa, budaya, dan di semua zona waktu.

Tapi ternyata banyak reaksi negatif bermunculan dari kalangan pengguna Twitter setelah tombol berlogo bintang itu digantikan dengan logo hati. Mereka meminta agar Twitter mengembaliknnya seperti semula.

"Mengubah tombol bintang Twitter menjadi hati adalah keputusan produk yang terburuk dalam sejarah internet, (kini) bookmark menjadi seperti endrosement," tulis akun @Jason.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Sementara pengguna Twitter dengan akun @deray menulis, "Sayang sekali Twitter menggunakan tombol "heart" di aplikasinya, sepertinya Twitter kian kehilangan jati dirinya, semoga ini hanya sementara saja."

Bahkan karyawan yang bekerja di Twitter pun mengungkapkan kekecewaannya terhadap keputusan twitter tersebut.

Akun @peterseibel menulis, "Saya bekerja di Twitter namun tidak percaya kami telah mengganti tombol yang terasa lebih netral seperti "favorite" dengan tombol yang penuh makna (banyak dipakai)."

Tombol hati memang telah banyak dipakai jejaring sosial lain, seperti Path dan Instagram, dan sedang diuji coba di Facebook sebagai pengganti Like. Dalam konteks jejaring sosial berbasis foto, tombol hati memang dirasa pas untuk mengekspresikan reaksi terhadap suatu postingan.

Namun berbeda dengan Twitter, di mana sebagian besar posting berbasis teks dan kontennya bisa beragam, yang bisa berarti banyak makna juga. Selain itu, tombol bintang juga selama ini dijadikan sebagai semacam tombol bookmark di Twitter untuk menyimpan suatu posting dalam timeline.

Melihat banyak protes dari pengguna, mungkinkah Twitter mendengar dan mengembalikan tombol favorit dengan logo bintangnya seperti semula?

Corpus ID : 166

Sumber :

http://tekno.kompas.com/read/2015/10/25/17270067/Yuk.Belajar.Bikin.Video.di.YouTube.Broadcast.Box

KOMPAS.com - YouTube dan Loop powered by Telkomsel meluncurkan YouTube Broadcast Box di Jakarta, Sabtu (24/10/2015). Yang membuatnya spesial, ini merupakan Broadcast Box pertama YouTube di Indonesia.

Apakah sebenarnya Broadcast Box itu? Ia merupakan studio mini yang dilengkapi dengan kamera profesional, props, dan peralatan editing. Studio tersebut dapat digunakan untuk berlatih bersama dengan para pakar sehingga kreator Indonesia bisa menciptakan konten yang lebih berkualitas.

Di dalam YouTube Broadcast Box, sebagaimana dijelaskan dalam keterangan pers yang diterima KompasTekno, kreator dapat mempelajari langkah membuat dan mengunggah video ke situs streaming video tersebut. Asyiknya lagi, kreator bakal mendapat panduan langsung dari pakarnya.

Setelah selesai membuat video, kreator tersebut bisa mengikuti workshop yang menampilkan para pakar dari YouTube yang akan berbagi tips dan trik untuk mengembangkan channel melalui Partnership Program dari YouTube.

Ada juga pelajaran mengenai cara melindungi konten original para kreator di dunia maya.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

Para pemberi materi workshop akan terdiri dari kreator YouTube lokal yang telah menguasai seni membuat video viral di dunia maya.

"Meski kami melihat pertumbuhan luar biasa untuk jumlah video yang diunggah di YouTube tahun ini, masih ada banyak potensi yang bisa digali dari kreator dan calon kreator Indonesia untuk meraih sukses melalui YouTube," kata Veronica Utami, Head of Marketing, Google Indonesia.

"Broadcast Box didesain dengan memikirkan karakter kreator Indonesia yang selama ini kami amati, agar mereka bisa memaksimalkan apa yang ditawarkan platform ini," lanjutnya.

YouTube

Suasana dalam YouTube Broadcast Box

Konten video dari Indonesia sendiri diklaim meningkat luar biasa pesat, yakni sebesar 600 persen, sejak tahun lalu. Peningkatan tersebut dikatakan sebagai yang tertinggi di Asia Pasifik. Dengan adanya Broadcast Box ini, diharapkan konten dari Indonesia bisa lebih dilirik lagi.

Inisiatif ini diluncurkan sebagai bagian dari roadshow Telkomsel Loop 2015, nantinya Broadcast Box akan hadir di kota Jakarta, Bandung, dan Surabaya, serta menyediakan workshop digital di Medan dan Makassar.

Di Jakarta, Broadcast Box akan hadir di Loop Station Mahakam, Jakarta Selatan dari 28 Oktober hingga 8 November mendatang. Di Bandung, ia bakal tersedia di Loop Kepo Trans Studio Mall pada 15 November.

Sementara itu, di Surabaya akan tersedia di Loop Kepo DBL Arena pada 22 Oktober.

Sedangkan, workshop di kota Medan akan dilakukan di Plaza Medan Fair pada 8 November 2015 dan kota Makassar di Grapari Telkomsel Makassar pada 28 November.

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

LAMPIRAN B HASIL PENGUJIAN ALGORITMA GENETIK

Kelas	Document Retrieved	Document Ground Truth	Precision	Recall	F-Measure
Siapa	[42, 53, 54, 55, 100, 101, 102, 109, 203, 266, 374, 379, 407]	[0, 1, 33, 47, 51, 52, 57, 58, 59, 61, 77, 96, 158, 208, 234, 236, 238, 264, 273, 276, 289, 308, 374]	0.07692307692307693	0.043478260869565216	0.05555555555555555
Dimana	[37, 39, 43, 56, 58, 59, 60, 69, 71, 72, 77, 85, 86, 91, 92, 93, 94, 95, 103, 106, 107, 112, 118, 129, 130, 138, 141, 142, 147, 151, 155, 156, 158, 160, 161, 162, 166, 166, 169, 169, 170, 172, 179, 188, 190, 193, 199, 202, 203, 206, 209, 218, 219, 220, 222, 223, 230, 232, 239, 240, 243, 247, 249, 264,	[2, 3, 4, 45, 46, 49, 50, 53, 55, 87, 138, 181, 182, 205, 208, 235, 242, 245, 321, 326, 330, 374, 393, 405, 406, 407]	0.02479338842975206 7	0.11538461538461539	0.04081632653061225

	267, 269, 270, 271, 274, 275, 276, 277, 278, 280, 280, 281, 287, 288, 291, 293, 298, 300, 305, 306, 307, 312, 313, 314, 315, 316, 322, 323, 323, 326, 328, 332, 334, 340, 342, 344, 346, 348, 348, 350, 352, 354, 356, 362, 364, 365, 374, 375, 379, 381, 383, 384, 390, 391, 394, 396, 399]				
Berapa	[46, 93, 166, 223, 227, 236, 280, 293, 336, 364, 374, 375]	[7, 8, 33, 34, 71, 90, 94, 95, 105, 110, 111, 112, 123, 136, 137, 138, 140, 142, 178, 189, 194, 205, 207, 208, 215, 218, 236, 243, 244, 250, 266, 267, 271, 276, 282, 286, 294,	0.08333333333333333	0.02272727272727272	0.03571428571428572

		303, 305, 313, 321, 324, 367, 403]			
Apa	[34, 200, 207, 208]	[32, 45, 50, 58, 66, 73, 77, 83, 91, 105, 106, 114, 118, 122, 129, 132, 138, 142, 147, 151, 158, 161, 164, 167, 170, 171, 173, 181, 207, 217, 219, 220, 235, 239, 245, 264, 273, 274, 275, 281, 289, 290, 305, 307, 313, 321, 326, 328, 330, 332, 334, 336, 338, 340, 342, 344, 346, 348, 350, 352, 353, 356, 358, 360, 362, 364, 365, 366, 374, 375, 380, 382, 383, 393, 394]	0.25	0.013333333333333334	0.02531645569620253 5
Kapan	[40, 125, 136, 208, 209, 216, 236, 239, 241, 243, 244, 248, 264, 266, 333, 336, 388]	[33, 42, 46, 49, 51, 54, 58, 59, 74, 84, 86, 93, 105, 163, 172, 182, 208, 220, 273, 283, 284, 287, 288, 308, 312, 315, 322, 367,	0.0588235294117647 05	0.0294117647058823 53	0.03921568627450981

		374, 376, 382, 393, 405, 407]			

LAMPIRAN C TERM

No	Term
1	#notesforfb
2	&
3	0
4	0
5	87
6	1
7	10
8	100
9	101
10	11
11	12
12	120
13	13
14	14
15	15
16	150
17	16
18	166
19	169
20	17
21	18
22	19
23	2
24	20
25	2013
26	2015

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

27	2016
28	2018
29	2020
30	2023
31	21
32	22
33	23
34	24
35	25
36	250
37	256
38	26
39	27
40	28
41	29
42	3
43	30
44	31
45	360
46	365
47	3d
48	3gb
49	4
50	400
51	46
52	490
53	4g
54	5
55	50
56	509

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

57	5m
58	6
59	600
60	62
61	6s
62	7
63	8
64	800
65	9
66	923
67	9to5google
68	@deray
69	@jason
70	@peterseibel
71	@snowden
72	@twitterid
73	@vine
74	a
75	ab
76	abdul
77	abu
78	academy
79	acak
80	acara
81	access
82	acung
83	ada

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

84	adat
85	adhitya
86	aditya
87	adobe
88	adopsi
89	adriansyah
90	adu
91	afrizal
92	agam
93	agama
94	agnosa
95	agustina
96	aham
97	ahimsa
98	ain
99	air
100	ajak
101	ajang
102	ajar
103	aka
104	akarta
105	akhir
106	akibat
107	akomodir
108	akrab
109	akses
110	aktif
111	aktivasi
112	aktivitas

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

113	aku
114	akun
115	akurat
116	alam
117	alas
118	alat
119	album
120	alih
121	along
122	alpha
123	alternatif
124	aman
125	amat
126	ambah
127	ambil
128	ambulans
129	amerika
130	ampik
131	anak
132	andatang
133	andreas
134	android
135	andung
136	aneka
137	anfaat
138	anggap
139	anggil
140	anggota
141	angka
142	angkas

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

143	angkat
144	ani
145	antang
146	antau
147	anti
148	anton
149	antoro
150	anyar
151	aplikasi
152	apliksi
153	aplilkasi
154	app
155	apple
156	apps
157	appsteriz
158	appsterize
159	arah
160	aren
161	arena
162	ari
163	arik
164	arrival
165	arrow
166	arti
167	artis
168	aruh
169	as
170	asa
171	asal

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

172	asang
173	asli
174	asuk
175	asyik
176	atas
177	atik
178	atur
179	audio
180	auditorium
181	augmented
182	avatar
183	awal
184	awan
185	awarding
186	b
187	baca
188	backdoor
189	badrullam
190	bagi
191	bagus
192	bahas
193	bahasa
194	bahaya
195	baid
196	baidu
197	baik
198	baju

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

199	bakal
200	bakat
201	baker
202	bala
203	balas
204	bali
205	bambang
206	band
207	banding
208	bandung
209	bangga
210	bangun
211	bantu
212	banyuwang
213	bar
214	barang
215	barat
216	baru
217	basir
218	basis
219	basitya
220	batam
221	batamcit
222	batas
223	batasi
224	batera
225	bau
226	bawa
227	bayar

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

228	be
229	beasiswa
230	bebas
231	beda
232	bei
233	bekal
234	belakang
235	beli
236	benam
237	benar
238	bencana
239	bentuk
240	beri
241	bersih
242	berta
243	besar
244	besok
245	beta
246	biaya
247	bicara
248	bidang
249	bintang
250	biro
251	biru
252	bisnis
253	blog
254	bloodjob
255	bo
256	bogor
257	book

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

258	bookmark
259	boomerang
260	bosan
261	botol
262	box
263	bpbd
264	broadband
265	broadcast
266	browser
267	brune
268	buat
269	buatanny
270	budaya
271	bug
272	buka
273	buku
274	bulan
275	bursa
276	buruk
277	bus
278	buswa
279	busway
280	butuh
281	by
282	cakap
283	cakup
284	calon

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

285	caltex
286	camat
287	camp
288	candu
289	capa
290	cara
291	card
292	cari
293	catat
294	caya
295	cepat
296	cerdas
297	cewa
298	channel
299	charts
300	chatting
301	chief
302	chin
303	china
304	chrome
305	chromebook
306	citra
307	city
308	cium
309	clash
310	clean
311	clevio
312	client
313	cloud

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

314	co
315	coba
316	coder
317	coders
318	coding
319	cody
320	com
321	communication
322	competition
323	content
324	contest
325	controlle
326	corporate
327	cpu
328	creative
329	creator
330	cuaca
331	cual
332	cukup
333	d
334	d3
335	daerah
336	daftar
337	dagang
338	daki
339	dalam
340	dampak
341	dana

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

342	danu
343	darah
344	darrussalam
345	darurat
346	dasar
347	dasarn
348	data
349	datang
350	daupert
351	daya
352	dbl
353	dea
354	dekat
355	dekstop
356	delap
357	dengan
358	dengar
359	denpasar
360	derajat
361	deret
362	desain
363	desktop
364	detail
365	detektif
366	detik
367	detil
368	dev
369	deva
370	development

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

371	device
372	diam
373	dicek
374	didik
375	dki
376	dng
377	dock
378	dokter
379	dokumen
380	dollar
381	dompet
382	donasi
383	donatur
384	doni
385	donor
386	down
387	download
388	dr
389	drawer
390	drs
391	dua
392	duel
393	dukung
394	durasi
395	e
396	ecah
397	ecek
398	edar
399	edit

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

400	editing
401	editor
402	editors
403	education
404	edward
405	efe
406	efek
407	efek’
408	efisien
409	ekonom
410	ekosistem
411	ekrut
412	eksis
413	ekspansi
414	eleg
415	elektronik
416	elepon
417	eligible
418	elit
419	elola
420	email
421	emas
422	embalikn
423	embang
424	embel
425	embus
426	emoj
427	emotic

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

428	emoticon
429	empu
430	ena
431	enam
432	enang
433	endal
434	endang
435	endara
436	endrosement
437	engadget
438	engah
439	engineer
440	enkripsi
441	environment
442	erima
443	erintah
444	eromoko
445	erti
446	erupsi
447	erus
448	erz4
449	erz4p
450	esan
451	eta
452	etahu
453	etas
454	facebook
455	fact
456	fair

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

457	faizal
458	faleddo
459	favorit
460	favorite
461	fenomena
462	feo
463	fikri
464	file
465	film
466	finalis
467	firma
468	firstbloo
469	fit
470	fitur
471	for
472	forbes
473	forex
474	forma
475	format
476	foto
477	fotograf
478	fotografer
479	founder
480	from
481	fungsi
482	gadget
483	gadjah
484	gagal
485	gak

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

486	galaxy
487	gali
488	gallery
489	Gambar
490	game
491	games
492	gandengtang
493	gandengtanga
494	gandrung
495	ganti
496	garis
497	gawa
498	gedung
499	gelar
500	gelas
501	gemar
502	gemastik
503	gembar
504	generasi
505	genggam
506	genre
507	geofisika
508	geografis
509	gerak
510	gguna
511	giat
512	gif
513	giritontro
514	giriwoyo

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

515	gital
516	glass
517	global
518	go
519	godok
520	golong
521	google
522	gorontalo
523	government
524	gps
525	grafik
526	grafis
527	graha
528	grant
529	grapar
530	graph
531	gratis
532	group
533	guna
534	gunung
535	guran
536	guru
537	habis
538	hackathon
539	hackaton
540	hadang
541	hadap
542	hadiah
543	hadir

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

544	halam
545	hapus
546	harap
547	hardware
548	harga
549	hari
550	haryanto
551	hash
552	hashtag
553	hasil
554	hati
555	head
556	healin
557	healink
558	health
559	heart
560	heboh
561	henti
562	hewan
563	hico
564	hidup
565	highlight
566	hijau
567	hilang
568	hilman
569	himawar
570	hisamuddin
571	hitobel
572	hitung

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

573	holmes
574	hong
575	horisontal
576	hotel
577	http
578	hubung
579	hujan
580	hybrid
581	i
582	icon
583	id
584	ide
585	identifikasi
586	identitas
587	idx
588	iklan
589	ikmat
590	ikon
591	ikonik
592	ikut
593	ila
594	ilih
595	ilik
596	ilustras
597	ima
598	ima
599	imajinasi
600	imbuh

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

601	imessages
602	impor
603	in
604	inap
605	inbound
606	inci
607	inda
608	indeks
609	index
610	indikator
611	indosat
612	industr
613	infeksi
614	info
615	informasi
616	informatika
617	infrastruktur
618	inggris
619	ingkat
620	inisiatif
621	innovation
622	inovasi
623	insititut
624	instagr
625	instagra
626	instagram
627	instal

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

628	instalasi
629	installed
630	instan
631	institusi
632	institut
633	institute
634	inta
635	intah
636	interaksi
637	internet
638	inti
639	intip
640	investasi
641	investor
642	ios
643	ipad
644	iphone
645	irim
646	isi
647	istiwa
648	isu
649	ita
650	iwatchyo
651	iwatchyou
652	iwic
653	jadi
654	jadwal
655	jahat
656	jajal

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

657	jajar
658	jakarta
659	jalan
660	jam
661	jamin
662	jangka
663	janji
664	januar
665	jari
666	jaring
667	jasa
668	jati
669	jatim
670	jauh
671	jawa
672	jawab
673	jejak
674	jejakk
675	jejaring
676	jek
677	jelas
678	jombie
679	jomblo
680	joox
681	jpg
682	jual
683	juara
684	judul
685	juli
686	jumat

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

687	juni
688	juri
689	juta
690	jutan
691	kabah
692	kabupaten
693	kadang
694	kait
695	kal
696	kalang
697	kalang
698	kali
699	kamera
700	kamis
701	kanisme
702	kantor
703	kapasitas
704	karakter
705	kari
706	karya
707	karyaw
708	kata
709	kategor
710	kawan
711	kazi
712	kembang
713	kena
714	kenal
715	kepo

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

716	kerap
717	keras
718	kering
719	kerja
720	kesan
721	khawatir
722	khusus
723	kian
724	kids
725	kilometer
726	kinerja
727	kiri
728	kirim
729	kit
730	klaim
731	klasik
732	klimatolog
733	klinik
734	kocek
735	kode
736	koding
737	koin
738	kolaborasi
739	kolek
740	koleksi
741	komando
742	kombinasi
743	komentar

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

744	komifo
745	kominfo
746	komitmen
747	kompas
748	kompastekno
749	kompastravel
750	kompatibel
751	kompetisi
752	komponen
753	kompresi
754	komputasi
755	komputer
756	komunikasi
757	komunitas
758	kondisi
759	koneksi
760	konfirmasi
761	kong
762	konsep
763	konteks
764	konten
765	kontur
766	kosong
767	kota

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

768	kreasi
769	kreativitas
770	kreator
771	krisis
772	kualitas
773	kuasa
774	kuat
775	kuesioner
776	kuis
777	kuliner
778	kumaeroh
779	kunjung
780	kurang
781	kuratorial
782	kurikulum
783	kurs
784	kustomisasi
785	kutip
786	lagu
787	lajar
788	laku
789	laman
790	lampau
791	langgan
792	langgana
793	langgar
794	langkah

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

795	langsung
796	lanjut
797	lansir
798	lanskap
799	lapang
800	lapor
801	laptop
802	latih
803	launcher
804	laurensius
805	lawan
806	lawas
807	layak
808	layan
809	layar
810	layout
811	le
812	learning
813	lebih
814	lega
815	legal
816	leluasa
817	lengkap
818	lenscoo
819	lenscoop
820	lepas
821	lerang
822	les
823	lewat

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

824	library
825	lifecycle
826	lifestyle
827	lightroom
828	lihat
829	likes
830	lima
831	limpung
832	lin
833	lindung
834	lintas
835	linux
836	lirik
837	lisensi
838	liter
839	live
840	login
841	logis
842	logo
843	lokal
844	lokasi
845	lomba
846	lombok
847	loop
848	lte
849	luar
850	luarga
851	luas
852	lubang
853	lulus

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

854	luncur
855	luter
856	lyric
857	m
858	maaf
859	mac
860	macet
861	mahakam
862	mahasiswa
863	mai
864	mail
865	main
866	maju
867	maka
868	makarim
869	makassar
870	makna
871	maksud
872	mall
873	mampu
874	mamu
875	mana
876	management
877	manager
878	manajemen
879	manjemen
880	mantan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

881	manual
882	manyar
883	marau
884	maret
885	marin
886	market
887	marketing
888	mart
889	martin
890	mas
891	masa
892	masa
893	masalahn
894	mashable
895	masjid
896	massage
897	masuk
898	masyarakat
899	mata
900	mataram
901	mater
902	mati
903	maya
904	mb
905	mbali
906	mej
907	menang
908	meng
909	menit

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

910	menu
911	mesti
912	micro
913	microsoft
914	miftahul
915	mikro
916	mikropon
917	miliar
918	milik
919	milimeter
920	millionaire
921	minat
922	minggu
923	minta
924	mirip
925	miris
926	misal
927	misi
928	mobil
929	mobile
930	mode
931	model
932	money
933	moplus
934	motor
935	mp
936	mp3
937	mubazir
938	muda

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

939	mudah
940	muka
941	mula
942	multi
943	mumpun
944	muncul
945	mundur
946	music
947	musik
948	musim
949	mustahil
950	mycroft
951	nadiem
952	nama
953	namis
954	nasabah
955	nasional
956	ndara
957	nebenger
958	nebengers
959	negara
960	negatif
961	neger
962	netral
963	new
964	news
965	next
966	nextde
967	nextdev

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

968	nextren
969	nge
970	night
971	nikah
972	nikmat
973	nila
974	nomor
975	non
976	nongkrong
977	notes
978	notifikasi
979	notify
980	november
981	nsa
982	nugraha
983	nur
984	nusa
985	nya
986	nyaman
987	nyata
988	objek
989	objektif
990	obrol
991	obyek
992	octacore
993	odifikasi
994	oem
995	of
996	off

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

997	office
998	offline
999	oik
1000	ojek
1001	ok
1002	oktober
1003	olahraga
1004	on
1005	onedrive
1006	onitor
1007	online
1008	onton
1009	open
1010	operasi
1011	operasional
1012	operator
1013	ople
1014	orang
1015	order
1016	organisasi
1017	organizer
1018	organizing
1019	ori
1020	original
1021	orisinil
1022	os
1023	ota
1024	otentikasi

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1025	otomatif
1026	otomatis
1027	otret
1028	out
1029	ows
1030	pa
1031	pacific
1032	page
1033	pagelar
1034	pagi
1035	paka
1036	pakar
1037	paket
1038	pala
1039	pandang
1040	pandu
1041	panggil
1042	panjang
1043	panjang
1044	panorama
1045	pantau
1046	paranggupito
1047	pariwisata
1048	partnership
1049	pas
1050	pasang
1051	pasar

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1052	pasifik
1053	path
1054	patri
1055	payah
1056	pc
1057	peca
1058	pendek
1059	penting
1060	penuh
1061	perak
1062	percay
1063	pergi
1064	perlu
1065	pers
1066	persen
1067	pesan
1068	pesat
1069	peta
1070	pgri
1071	phhhoto
1072	phone
1073	phonearena
1074	photo
1075	photoshop
1076	piala
1077	pialang
1078	pick
1079	pidato
1080	pihak

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1081	pijat
1082	pikir
1083	pilih
1084	pin
1085	pinda
1086	pinjam
1087	pintar
1088	pirant
1089	pisah
1090	pitching
1091	place
1092	plaform
1093	planet
1094	platform
1095	plawang
1096	play
1097	player
1098	playlist
1099	plaza
1100	plus
1101	poem
1102	poems
1103	poin
1104	politeknik
1105	polling
1106	ponsel
1107	pontianak
1108	pontinesi
1109	populer

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1110	portal
1111	portofolio
1112	portrait
1113	posisi
1114	posting
1115	potensi
1116	powered
1117	prabowo
1118	pracimantoro
1119	praktis
1120	pranoto
1121	pre
1122	presentasi
1123	presenter
1124	presiden
1125	presisi
1126	private
1127	pro
1128	produk
1129	produktivitas
1130	profesional
1131	profil
1132	profit
1133	program
1134	promosi
1135	proposal
1136	props

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1137	proses
1138	prosesor
1139	protes
1140	proyek
1141	publicity
1142	publik
1143	publok
1144	pujiachir
1145	pulau
1146	pungkas
1147	putar
1148	putus
1149	qcar
1150	qcare
1151	quick
1152	quiz
1153	quotes
1154	ra
1155	rabu
1156	rachman
1157	radio
1158	raharjo
1159	raih
1160	rama
1161	ramadh
1162	rancang
1163	rangka
1164	rangka
1165	rangkum
1166	rangsang

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1167	ranita
1168	rasa
1169	ratus
1170	raw
1171	rdeka
1172	reaksi
1173	real
1174	reality
1175	recent
1176	redphone
1177	regional
1178	registrasi
1179	rehat
1180	rekam
1181	rektur
1182	releases
1183	remaja
1184	reminders
1185	reply
1186	repot
1187	republik
1188	request
1189	resmi
1190	riau
1191	ribu
1192	ride
1193	rilis
1194	rinci
1195	ringan

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1196	rinjan
1197	rintis
1198	ririek
1199	risa
1200	riset
1201	roadshow
1202	rogoh
1203	rogram
1204	ronde
1205	roni
1206	roses
1207	roussea
1208	rp
1209	ruang
1210	rumah
1211	rumahsina
1212	rumahsinau
1213	rumors
1214	running
1215	rupiah
1216	rusak
1217	s
1218	s2
1219	sabar
1220	sabha
1221	sable
1222	sabtu
1223	sadap

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1224	sahabat
1225	saham
1226	sakit
1227	saldo
1228	samsung
1229	sandang
1230	sangka
1231	sangkut
1232	sanyi
1233	sarana
1234	sarjana
1235	satelit
1236	satu
1237	saudarak
1238	scalability
1239	sd
1240	sdk
1241	searah
1242	search
1243	sebag
1244	sebar
1245	sebesar
1246	secara
1247	secepat
1248	security
1249	sederhana
1250	sedia
1251	sedot
1252	sehar

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1253	sehat
1254	sejal
1255	sejarah
1256	sejat
1257	sejenis
1258	sejuta
1259	sekadar
1260	seki
1261	sekilas
1262	sekolahkodin
1263	selancar
1264	selanjut
1265	selasa
1266	selat
1267	seleksi
1268	selesa
1269	selesa
1270	self
1271	selidik
1272	selincah
1273	seliwer
1274	selter
1275	seluler
1276	semarang
1277	sembarang
1278	sembil
1279	semoga
1280	sempat
1281	sempat

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1282	semua
1283	senaw
1284	senggang
1285	seni
1286	senin
1287	sensitif
1288	sensor
1289	sentosa
1290	sentuh
1291	sepakat
1292	sepeda
1293	september
1294	seputar
1295	serang
1296	serangka
1297	serial
1298	serikat
1299	serta
1300	sertifikat
1301	seru
1302	server
1303	sesi
1304	sesua
1305	set
1306	setara
1307	setengah
1308	setra
1309	setting
1310	seuah

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1311	share
1312	sheet
1313	sherlock
1314	shortcut
1315	shutterstoc
1316	siang
1317	siap
1318	sifat
1319	sign
1320	signal
1321	silicon
1322	simbol
1323	simpan
1324	sinau
1325	sing
1326	singapura
1327	singkat
1328	sinkronisasi
1329	sinyalir
1330	sistem
1331	situs
1332	skor
1333	skype
1334	sma
1335	smart
1336	smartgorontal
1337	smartgorontalo

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1338	smartphone
1339	smp
1340	snapseed
1341	snowden
1342	soal
1343	software
1344	solusi
1345	sonalisasi
1346	sorot
1347	sosial
1348	sourcedna
1349	spesial
1350	spin
1351	spotlight
1352	ssage
1353	ssenger
1354	stagefright
1355	startup
1356	stasiun
1357	station
1358	status
1359	stock
1360	stor
1361	store
1362	streaming
1363	stree
1364	studio
1365	suara

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1366	sub
1367	submit
1368	subtema
1369	suguh
1370	suka
1371	sukses
1372	sulit
1373	sumber
1374	surabaya
1375	surve
1376	susah
1377	susul
1378	swasta
1379	switch
1380	systems
1381	t
1382	tab
1383	Tabel
1384	Tabelt
1385	tag
1386	tahap
1387	tahu
1388	tahu
1389	tahun
1390	tajuk
1391	taktis
1392	takut
1393	talenta
1394	tama
1395	taman

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1396	tambah
1397	tampak
1398	tampil
1399	tanam
1400	tangan
1401	tanggal
1402	tanggap
1403	tanggulang
1404	tanggung
1405	tangkap
1406	tangki
1407	tani
1408	tantang
1409	tantow
1410	tanya
1411	target
1412	taruh
1413	tasking
1414	tatrader
1415	taut
1416	team
1417	tebar
1418	tebeng
1419	teens
1420	tekan
1421	teknisi
1422	tekno
1423	teknolog
1424	teks

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1425	telekomunikasi
1426	telepon
1427	telit
1428	telkom
1429	telkomsel
1430	tema
1431	teman
1432	temen
1433	tempat
1434	tempuh
1435	temu
1436	tengah
1437	tenggara
1438	tenggat
1439	tenteng
1440	teorolog
1441	ter
1442	tera
1443	terang
1444	terap
1445	teri
1446	ternam
1447	tertib
1448	terus
1449	tetangga
1450	tetap
1451	textsecure
1452	the
1453	theme

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1454	ti
1455	tidaknyam
1456	tiga
1457	tiket
1458	tilang
1459	tim
1460	time
1461	timeline
1462	tindak
1463	tinggal
1464	tinggi
1465	tingkah
1466	tingkat
1467	tipis
1468	tips
1469	titik
1470	to
1471	tode
1472	toko
1473	tolak
1474	tombol
1475	tools
1476	top
1477	topik
1478	toring
1479	total
1480	touch
1481	tourism
1482	trade

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1483	trading
1484	tradisi
1485	trans
1486	transaksi
1487	transjakart
1488	transjakarta
1489	transportasi
1490	transportation
1491	traveler
1492	traveling
1493	travi
1494	travis
1495	tren
1496	trend
1497	trends
1498	tri
1499	trik
1500	trip
1501	tugas
1502	tuju
1503	tukar
1504	tulis
1505	tumbuh
1506	tumpang
1507	tuna
1508	tunjang
1509	turut

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1510	tutup
1511	tv
1512	tvos
1513	twitter
1514	uang
1515	uasa
1516	uat
1517	ubah
1518	uda
1519	udah
1520	ujar
1521	uji
1522	uju
1523	ukm
1524	ukur
1525	ulik
1526	ulis
1527	umum
1528	undang
1529	unduh
1530	unggah
1531	unggu
1532	unggul
1533	ungkap
1534	unicode
1535	unicorn
1536	unik
1537	universal
1538	universitas

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1539	university
1540	unjang
1541	untuk
1542	untung
1543	upa
1544	upaya
1545	update
1546	urang
1547	urut
1548	usa
1549	usaha
1550	user
1551	usung
1552	utam
1553	utama
1554	utar
1555	utilities
1556	utuh
1557	utup
1558	valley
1559	vari
1560	venditama
1561	vendor
1562	verge
1563	verifikasi
1564	veronica
1565	versi
1566	vertikal
1567	video

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1568	vip
1569	viral
1570	virtual
1571	vulkanik
1572	w
1573	wadah
1574	wajib
1575	wakil
1576	waktu
1577	wants
1578	warga
1579	warna
1580	warni
1581	warsito
1582	watch
1583	watchos
1584	watson
1585	web
1586	website
1587	whatsapp
1588	who
1589	widget
1590	wilayah
1591	windows
1592	windri
1593	wireless
1594	wisata
1595	wonogir
1596	work

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA

1597	workshop
1598	world
1599	wujud
1600	www
1601	x
1602	xcode
1603	yahya
1604	yappi
1605	ynag
1606	yogyakarta
1607	youm
1608	youtub
1609	youtube
1610	yuni
1611	yusuf
1612	zona
1613	–
1614	—
1615	‘info
1616	‘poems’
1617	“5m”
1618	“kondisi
1619	“saran
1620	”

repository.ub.ac.id
UNIVERSITAS BRAWIJAYA