



TESIS - IF 185401

**DETEKSI AKUN SPAMMER BERDASARKAN
HASHTAG DAN AKTIFITAS KOMUNITAS PADA
TWITTER**

Arif Mudi Priyatno
NRP. 0511185001039

DOSEN PEMBIMBING

Prof. Dr. Agus Zainal Arifin, S.Kom., M.Kom.

**Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
2020**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

Magister Teknik Informatika (M.Kom.)

di

Institut Teknologi Sepuluh Nopember

Oleh:

ARIF MUDI PRIYATNO

NRP. 05111850010039

Tanggal Ujian: 20 Juli 2020

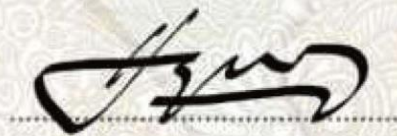
Periode Wisuda: September 2020

Disetujui oleh:

Pembimbing:

Prof. Dr. Agus Zainal Arifin, S.Kom., M.Kom.


NIP. 197208091995121001



Penguji:

Waskitho Wibisono, S.Kom., M.Eng., Ph.D.

NIP. 197410222000031001




Dr. Eng. Chastine Faticah, S.Kom., M.Kom.

NIP. 197512202001122002



Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.

NIP. 1984101620081210002

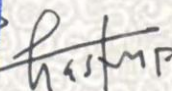


Kepala Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas



Dr. Chastine Faticah, S.Kom, M.Kom.

NIP. 197512202001122002



[Halaman ini sengaja dikosongkan]

DETEKSI AKUN SPAMMER BERDASARKAN HASHTAG DAN AKTIFITAS KOMUNITAS PADA TWITTER

Nama Mahasiswa : Arif Mudi Priyatno
NRP : 05111850010039
Pembimbing : Prof. Dr. Agus Zainal Arifin, S.Kom., M.Kom.

ABSTRAK

Spammer meniru pola perilaku pengguna yang sah untuk menghindari terdeteksi oleh pendeteksi *spam*. *Spammer* membuat banyak *account* palsu dan berkolaborasi satu sama lain untuk membentuk komunitas. Kolaborasi antar *account* menggunakan *tweet* berbeda-beda tetapi menggunakan *hashtag* untuk mencapai tujuan. Kolaborasi *tweet* berbeda ini akan membuat seperti *hashtag natural* yang banyak diperbincangkan orang di *Twitter*. Kolaborasi ini mempersulit deteksi *account spammer* karena berada di antara *account legitimate* yang membuat *tweet* dengan *hashtag*.

Penelitian ini mengusulkan pengembangan ekstraksi fitur berdasarkan *hashtag* dan aktivitas komunitas untuk mendeteksi *account spammer* di *Twitter*. Proses penentuan komunitas menggunakan perangkat lunak *gephi* dengan data *following account* yang membuat *tweet* pada *account Twitter* Komisi Pemberantasan Korupsi Republik Indonesia dari tanggal 01 Agustus 2018 hingga 10 September 2019. Data tersebut dilakukan proses ekstraksi fitur *account*, *tweet*, dan komunitas. *Tweet* dilakukan dengan menggunakan *hashtag*. *Hashtag* digunakan oleh *spammer* untuk meningkatkan popularitas. Pada fitur *account*, *tweet*, dan komunitas dilakukan seleksi fitur untuk mendapatkan fitur optimal dengan menggunakan *Recursive Feature Elimination*. Hasil seleksi fitur digunakan untuk mendapatkan nilai *legitimate* atau *spammer* menggunakan *multi-layer perceptron*. Hasil nilai dilakukan penggabungan dengan menggunakan pembobotan untuk penentuan *spammer*.

Hasil percobaan menunjukkan bahwa pengembangan ekstraksi fitur berdasarkan *hashtag* dan aktivitas komunitas berhasil untuk mendeteksi *account spammer* di *Twitter*. Metode yang diusulkan mendapatkan nilai akurasi, *recall*, presisi, dan g-mean masing-masing 90,6%, 88,0%, 3,2%, dan 16,7%. Hasil fitur seleksi pada fitur *tweet* setiap pembagian persentase dataset tidak mengalami perubahan yaitu 6 fitur. Pada fitur *tweet* terdapat fitur jumlah *hashtag* dan *unique hashtag*. *Unique hashtag* yang tinggi menunjukkan bahwa *account* melakukan *hashtag* berbeda-beda yang tidak memiliki keterkaitan terhadap *spammer*. Hal ini membuktikan bahwa *hashtag* digunakan oleh *spammer* dalam aktivitasnya. Bobot fitur *account* dan *tweet* memiliki keseimbangan dalam keberhasilan penentuan *spammer*. Fitur komunitas dilihat berdasarkan bobot yang digunakan, dengan hasil yang bagus yaitu lebih kecil dari 0,3.

Kata Kunci: *Spammer, Komunitas, Hashtag, Twitter, Multi-Layer Perceptron*

[Halaman ini sengaja dikosongkan]

DETECT SPAMMER ACCOUNTS BASED ON HASHTAGS AND COMMUNITY ACTIVITIES ON TWITTER

Nama Mahasiswa : Arif Mudi Priyatno

NRP : 05111850010039

Pembimbing : Prof. Dr. Agus Zainal Arifin, S.Kom., M.Kom.

ABSTRACT

Spammers imitate legitimate user behavior patterns to avoid being detected by spam detectors. Spammers create lots of fake accounts and collaborate with each other to form communities. Collaboration between accounts uses different tweets but using hashtags to achieve goals. This different tweet collaboration will make it look like a natural hashtag that many people talk about on Twitter. This collaboration makes it difficult to detect spammers' accounts because they are among legitimate accounts that tweet with hashtags.

This research proposed the development of feature extraction based on hashtags and community activities to detect spammer accounts on Twitter. The process of determining the community uses Gephi software by using the following account data which tweeted on the Indonesian Republic's Corruption Eradication Commission's Twitter account from 01 August 2018 to 10 September 2019. Data was extracted about the account, tweet, and community. Tweets done contain a hashtag. Hashtags are used by spammers to increase popularity. Features about accounts, tweets, and community features are selected to get optimal features using Recursive Feature Elimination. The feature selection results are used to get legitimate values or spammers using multi-layer perceptron. The results of the value are merged using weighting to determine the spammer's account.

The experimental results show that the development of feature extraction based on hashtags and community activities has succeeded in detecting spammers' accounts on Twitter. The proposed method obtains accuracy, recall, precision, and g-mean with 90.6%, 88.0%, 3.2% and 16.7% respectively. The results of the selection feature on the tweet feature per division of the dataset does not change, namely 6 features. In the tweet feature there is a hashtag and unique hashtag feature. High unique hashtags indicate that accounts have different hashtags that are not related to spammers. This proves that the hashtag is used by spammers in their activities. The weight of account and tweet features has a balance in determining the success of spammers. Community features based on weights are used that good results are smaller than 0.3.

Keywords : Spammers, Community, Hashtag, Twitter, Multi-Layer Perceptron

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan Tesis yang berjudul “Deteksi *Account Spammer* Berdasarkan *Hashtag* dan Aktivitas Komunitas pada *Twitter*”.

Pengerjaan tesis ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah diperoleh di Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Informatika. Harapan penulis semoga apa yang tertulis di dalam buku tesis ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini. terselesaikannya buku tesis ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT atas semua rahmat yang diberikan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
2. Kedua orang tua penulis atas segala dukungan, motivasi serta doa sehingga penulis dapat mengerjakan tesis ini.
3. Bapak Prof. Dr. Agus Zainal Arifin, S.Kom., M.Kom, selaku dosen pembimbing, atas bimbingan, arahan dan bantuannya dalam pengerjaan tesis ini.
4. Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D., Ibu Dr. Eng. Chastine Fatichah, S.Kom., M.Kom., dan Bapak Dr.Eng. Radityo Anggoro, S.Kom., M.Sc. selaku penguji tesis ini yang telah memberikan saran serta masukan untuk penyempurnaan tesis ini.
5. Bapak Dr.Eng. Darlis Heru Murti, S.Kom., M.Kom., selaku dosen wali penulis atas motivasi yang luar biasa sehingga memperlancar penulis dalam menyelesaikan studi.
6. Bapak dan Ibu Dosen program studi S2 Departemen Informatika yang telah memberi bimbingan perkuliahan sehingga menambah wawasan dan pengetahuan penulis.

7. Tika Permata Sari dan dr. Satria Ekatama selaku adik dan kakak penulis atas segala dukungan dan doa yang telah diberikan sehingga penulis tetap semangat dalam mengerjakan tesis ini.
8. Seluruh teman-teman S2 Informatika ITS angkatan 2018 yang selama ini sudah membantu penulis menyelesaikan tesis.
9. Seluruh teman-teman di kos perumahan dosen blok w no 5 yang telah banyak membantu dan memberikan *support* untuk menyelesaikan tesis ini.
10. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu penulis dalam menyelesaikan tesis ini.

Penulis menyadari bahwa masih terdapat kekurangan, kesalahan, maupun kelalaian yang telah penulis lakukan dalam penelitian ini. Oleh karena itu, saran dan kritik yang membangun sangat dibutuhkan untuk penyempurnaan tesis ini.

Surabaya, Juli 2020

Arif Mudi Priyatno

DAFTAR ISI

ABSTRAK	V
<i>ABSTRACT</i>	VII
KATA PENGANTAR	IX
DAFTAR ISI.....	XI
DAFTAR GAMBAR	XIII
DAFTAR TABEL.....	XV
DAFTAR LAMPIRAN.....	XVII
BAB 1 PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 PERUMUSAN MASALAH.....	4
1.3 TUJUAN PENELITIAN	4
1.4 MANFAAT PENELITIAN	4
1.5 KONTRIBUSI PENELITIAN	4
1.6 BATASAN MASALAH	4
BAB 2 KAJIAN PUSTAKA.....	7
2. 1.DATA.....	7
2.1.1. <i>Twitter</i>	7
2.1.2. <i>Label Propagation</i>	8
2. 2.DETEKSI KOMUNITAS	9
2.3. EKSTRAKSI FITUR.....	14
2.4. <i>RECURSIVE FEATURE ELIMINATION</i>	21
2.5. DETEKSI AKUN <i>SPAMMER</i>	23
2.5.1. <i>Multi-layer Perceptron</i>	28
2.5.2. <i>Support Vector Machine</i>	31
2.5.3. <i>Metode Evaluasi Deteksi Akun Spammer</i>	33
BAB 3 METODOLOGI PENELITIAN.....	35
3. 1.PERANCANGAN.....	35
3.1.1 <i>Pengumpulan Data</i>	36
3.1.2 <i>Deteksi Komunitas</i>	36

3.1.3 Ekstraksi fitur.....	39
3.1.4 Seleksi Fitur.....	43
3.1.5 Deteksi Akun Spammer.....	44
3. 2. IMPLEMENTASI.....	46
3. 3. PENGUJIAN DAN EVALUASI.....	46
3. 4. DOKUMENTASI.....	47
BAB 4 HASIL DAN PEMBAHASAN.....	49
4.1. DATA.....	49
4.2. DETEKSI KOMUNITAS.....	51
4.3. EKSTRAKSI FITUR.....	52
4.4. SELEKSI FITUR.....	59
4.5. DETEKSI SPAMMER.....	62
BAB 5 KESIMPULAN DAN SARAN.....	69
5.1. KESIMPULAN.....	69
5.2. SARAN.....	69
DAFTAR PUSTAKA.....	71
LAMPIRAN.....	76
BIOGRAFI PENULIS.....	110

DAFTAR GAMBAR

Gambar 2.1 Contoh <i>spam</i> pada <i>Twitter</i>	10
Gambar 2.2 Contoh <i>graph</i> 3 komunitas.....	10
Gambar 2.3 Ilustrasi metode <i>louvain</i>	11
Gambar 2.4 Contoh Jaringan Dari <i>Node</i>	16
Gambar 2.5 Contoh Hasil <i>Indegree</i> dan <i>Outdegree</i>	17
Gambar 2.6 Seleksi Fitur <i>Support Vector Machine</i> <i>Recursive Feature Elimination</i>	22
Gambar 2.7 Proses pelatihan <i>Multi-layer Perceptron</i>	23
Gambar 2.8 Ilustrasi <i>Hyperplane</i>	32
Gambar 3.1 Alur Metodologi Penelitian.....	35
Gambar 3.2 Perancangan Penelitian Secara Umum.....	35
Gambar 3.3 Bentuk <i>Following</i> pada Akun	37
Gambar 3.4 Diagram Proses Alur Deteksi Komunitas	38
Gambar 3.5 Daftar <i>Following</i>	38
Gambar 3.6 Arsitektur <i>Multi-Layer Perceptron</i>	44
Gambar 4.1 Contoh <i>tweet</i> dari <i>username @tifsembiring</i>	50
Gambar 4.2 Hasil <i>feature selection</i> dataset 70:30 menggunakan <i>recursive feature elimination</i> : (a) <i>feature accounts</i> , (b) <i>feature tweets</i> , dan (c) <i>feature communities</i>	58
Gambar 4.3 Hasil <i>feature selection</i> dataset 80:20 menggunakan <i>recursive feature elimination</i> : (a) <i>feature accounts</i> , (b) <i>feature tweets</i> , dan (c) <i>feature communities</i>	58
Gambar 4.4 Hasil <i>feature selection</i> dataset 90:10 menggunakan <i>recursive feature elimination</i> : (a) <i>feature accounts</i> , (b) <i>feature tweets</i> , dan (c) <i>feature communities</i>	58
Gambar 4.5 Contoh dataset <i>username</i> , (a) (b) (c) merupakan dataset <i>legitimate</i> , (d) (e) (f) merupakan dataset <i>spammer</i>	63

Gambar 4.6 Perbandingan evaluasi pengujian, (a) akurasi, (b) *recall*, (c) presisi,
dan (d) G-mean..... 65

DAFTAR TABEL

Tabel 2.1 Fitur Jaringan (<i>graph</i>)	14
Tabel 2.2 Fitur Komunitas dan Keterangan	14
Tabel 2.3 Fitur Pada Akun <i>Twitter</i>	15
Tabel 2.4 <i>Kernel</i> pada <i>Support Vector Machine</i>	31
Tabel 3.1 Tabel <i>Confusion Matrix</i> pengujian	47
Tabel 4.1 Contoh <i>Tweet</i> dari halaman @kpk_ri	49
Tabel 4.2 Contoh informasi <i>account username</i>	50
Tabel 4.3 Contoh <i>source</i> dan target pada <i>edge list</i>	51
Tabel 4.4 Contoh hasil deteksi komunitas	51
Tabel 4.5 Hasil Ekstraksi informasi <i>account</i>	53
Tabel 4.6 Hasil Ekstraksi informasi <i>tweet</i>	55
Tabel 4.7 Hasil Ekstraksi informasi komunitas	57
Tabel 4.8 Hasil <i>multi-layer perceptron</i> persentase dataset 70:30, lr 0,1 <i>neuron hidden</i> 15 $\alpha = 0,45$, $\beta = 0,45$, dan $\gamma = 0,1$	65
Tabel 4.9 Nilai bobot α , β , dan γ MLP <i>neuron hidden</i> 15 Lr 0,1 Persentase dataset 70:30	66

[Halaman ini sengaja dikosongkan]

DAFTAR LAMPIRAN

Lampiran 1 Nilai bobot, MLP <i>neuron</i> hidden 15 Lr 0,01	
Persentase dataset 70:30	76
Lampiran 2 Nilai bobot, MLP <i>neuron</i> hidden 15 Lr 0,001	
Persentase dataset 70:30	77
Lampiran 3 Nilai bobot, MLP <i>neuron</i> hidden 15 Lr 0,1	
Persentase dataset 80:20	78
Lampiran 4 Nilai bobot, MLP <i>neuron</i> hidden 15 Lr 0,01	
Persentase dataset 80:20	80
Lampiran 5 Nilai bobot, MLP <i>neuron</i> hidden 15 Lr 0,001	
Persentase dataset 80:20	81
Lampiran 6 Nilai bobot, MLP <i>neuron</i> hidden 15 Lr 0,1	
Persentase dataset 90:10	82
Lampiran 7 Nilai bobot, MLP <i>neuron</i> hidden 15 Lr 0,01	
Persentase dataset 90:10	83
Lampiran 8 Nilai bobot, MLP <i>neuron</i> hidden 15 Lr 0,001	
Persentase dataset 90:10	85
Lampiran 9 Nilai bobot, MLP <i>neuron hidden</i> 20 Lr 0,1	
Persentase dataset 70:30	86
Lampiran 10 Nilai bobot, MLP <i>neuron hidden</i> 20 Lr 0,01	
Persentase dataset 70:30	87
Lampiran 11 Nilai bobot, MLP <i>neuron hidden</i> 20 Lr 0,001	
Persentase dataset 70:30	89
Lampiran 12 Nilai bobot, MLP <i>neuron hidden</i> 20 Lr 0,1	
Persentase dataset 80:20	90
Lampiran 13 Nilai bobot, MLP <i>neuron hidden</i> 20 Lr 0,01	
Persentase dataset 80:20	91
Lampiran 14 Nilai bobot, MLP <i>neuron hidden</i> 20 Lr 0,001	
Persentase dataset 80:20	92

Lampiran 15 Nilai bobot, MLP <i>neuron hidden</i> 20 Lr 0,1	
Persentase dataset 90:10.....	94
Lampiran 16 Nilai bobot, MLP <i>neuron hidden</i> 20 Lr 0,01	
Persentase dataset 90:10.....	95
Lampiran 17 Nilai bobot, MLP <i>neuron hidden</i> 20 Lr 0,001	
Persentase dataset 90:10.....	96
Lampiran 18 Nilai bobot, MLP <i>neuron hidden</i> 25 Lr 0,1	
Persentase dataset 70:30.....	98
Lampiran 19 Nilai bobot, MLP <i>neuron hidden</i> 25 Lr 0,01	
Persentase dataset 70:30.....	99
Lampiran 20 Nilai bobot, MLP <i>neuron hidden</i> 25 Lr 0,001	
Persentase dataset 70:30.....	100
Lampiran 21 Nilai bobot, MLP <i>neuron hidden</i> 25 Lr 0,1	
Persentase dataset 80:20.....	101
Lampiran 22 Nilai bobot, MLP <i>neuron hidden</i> 25 Lr 0,01	
Persentase dataset 80:20.....	103
Lampiran 23 Nilai bobot, MLP <i>neuron hidden</i> 25 Lr 0,001	
Persentase dataset 80:20.....	104
Lampiran 24 Nilai bobot, MLP <i>neuron hidden</i> 25 Lr 0,1	
Persentase dataset 90:10.....	105
Lampiran 25 Nilai bobot, MLP <i>neuron hidden</i> 25 Lr 0,01	
Persentase dataset 90:10.....	107
Lampiran 26 Nilai bobot, MLP <i>neuron hidden</i> 25 Lr 0,001	
Persentase dataset 90:10.....	108

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teknologi yang muncul dari jejaring sosial *online* telah mengarah pada pengembangan berbagai *platform* dimana jutaan entitas sosial berkolaborasi dan berkomunikasi satu sama lain. Misalnya tahun 2016, *Facebook* memiliki 1,5 miliar pengguna aktif, sementara *Twitter* memiliki 320 juta pengguna aktif (Bindu, Mishra, dan Thilagam, 2018). Statistik pengguna jejaring sosial *online* diperkirakan pada tahun 2020 mencapai 2,46 miliar (Inuwa-Dutse, Liptrott, dan Korkontzelos, 2018) hal ini memungkinkan pengguna mereka untuk tetap berhubungan dengan teman-teman, berbagi ide, informasi, mempublikasikan informasi pribadi, melakukan komunikasi terkait pekerjaan, dan lain-lain (Bindu dkk., 2018). Pengguna jejaring sosial *online* ini secara bebas menghasilkan dan mengkonsumsi informasi yang mengarah pada jumlah data yang belum pernah terjadi sebelumnya. Beberapa *domain* telah mengakui peran penting dari analisis jejaring sosial *online* dalam meningkatkan produktivitas dan mendapatkan keunggulan kompetitif. Informasi yang berasal dari jejaring sosial *online* telah digunakan dalam layanan kesehatan untuk mendukung pemberian layanan yang efektif (Rojas dkk, 2016), (Chien, Miils, dan Airey, 2008), dalam bidang olahraga untuk terlibat dengan penggemar (Liu dan 2016), (Bagić Babac dan Podobnik, 2016), dan bidang politik untuk melacak proses pemilihan, mempromosikan keterlibatan yang lebih luas dengan pendukung (Contractor dkk, 2015).

Twitter adalah salah satu jejaring sosial *online* paling populer dan paling cepat berkembang yang didirikan pada tahun 2006. *Twitter* telah muncul sebagai *platform microblogging* paling populer dimana pengguna dapat berbagi berita, media, meme, pandangan, dan pembaruan dalam bentuk *tweet*. *Tweet* adalah tulisan yang berisi teks dan URL HTTP terbatas hingga 280 karakter (Rozen, 2017). Sayangnya, pertumbuhan interaksi sosial *Twitter* telah menarik perhatian para penjahat dunia maya yang mengeksploitasi hubungan kepercayaan di antara

para pengguna untuk mendistribusikan konten jahat ke sejumlah besar korban pada jaringan. Jenis *spamming* yang paling banyak dikenal di *Twitter* adalah menangkap topik yang sedang hangat (Martinez-Romo dan Araujo, 2013). Kapan pun peristiwa itu terjadi, pengguna mencoba untuk mengekspresikan pendapat atau berbagi informasi tentang peristiwa tersebut menggunakan *hashtag* atau kata kunci yang sama. Jika topik tersebut paling banyak di *tweet* pada hari tersebut, maka akan terlihat oleh semua pengguna *Twitter* di beranda *Twitter* sebagai topik yang sedang hangat. *Spammer* menggunakan *hashtag* yang sama untuk dapat dilihat oleh basis pengguna yang besar setelah kejadian tertentu, disertai dengan URL yang tidak diminta dan mengarah ke situs *web* yang tidak terkait dengan kejadian yang sedang populer. Karena keterbatasan 280 karakter di *Twitter*, pengguna biasanya berbagi URL dengan menggunakan layanan pemendek URL.

Spammer biasanya meniru pola perilaku pengguna yang sah untuk menghindari terdeteksi oleh teknik deteksi *spam*. *Spammer* mengembangkan alat dan teknik untuk menghindari teknik deteksi yang ada. Selain itu, tren penelitian saat ini tentang deteksi *spam* memiliki kendala kompleksitas atau memiliki beberapa peringatan yang dapat dilewati oleh *spammer*. Dalam hal ini, sangat diinginkan untuk mendeteksi dan memblokir/menghapus *spammer* dari jejaring sosial seperti *Twitter*, untuk menghemat sumber daya dan upaya manusia dari pengguna yang tidak diinginkan. Termasuk fitur yang lebih kuat dan lebih sulit untuk ditiru serta menggunakan interaksi pengguna di dalam dan di luar struktur komunitas dapat digunakan untuk membangun model klasifikasi *spam* sehingga menyulitkan *spammer*. *Spammer* membentuk banyak akun palsu, dan berkolaborasi satu sama lain membentuk komunitas yang erat untuk meningkatkan kredibilitas mereka. Dengan demikian, akun *spammer* cenderung terhubung secara sosial dengan koefisien pengelompokan yang tinggi (Yang dkk, 2012).

Berbagai cara dan langkah telah dilakukan untuk mendeteksi *spammer* pada *Twitter*. Karakteristik bot *spammer* menurut (Yang, Harkreader, dan Gu, 2011)

ada 5 yaitu *spam* berisi *link* aktif, *spam* berisi promosi produk tertentu, memiliki kesamaan antara *tweet* sebelum dan sesudahnya, akun baru dan *spam* sering menggunakan *hashtag*. Penelitian (Aditya dkk, 2016) melakukan deteksi bot *spammer* dengan melihat karakteristik waktu *posting* dan sentimen dari *tweet* yang dilakukan. Penelitian (Inuwa-Dutse dkk., 2018) melakukan deteksi *spammer* dengan mengoptimalkan serangkaian fitur dari *tweet history* dan informasi akun pengguna. Dari hasil analisa yang dilakukan, diketahui bahwa pengguna *spam* cenderung selektif dalam mengikuti pengguna lain sehingga membentuk koneksi *spammer*. Selain itu rata-rata akun *spam* otomatis memposting setidaknya 12 *tweet* per hari pada periode yang ditentukan dengan baik. Penelitian (Bhat dan Abulaish, 2013) melakukan identifikasi *spammer* pada *Facebook* dengan menggunakan fitur komunitas. Fitur komunitas yang digunakan pada penelitian ini yaitu total *out-degree*, total *reciprocity*, total *in/out ratio*, *community memberships*, *foreign out-degree* dan *foreign in/out ratio*. Dari penelitian tersebut diketahui bahwa penggabungan fitur komunitas dan non-komunitas mampu meningkatkan hasil deteksi *spammer* yang signifikan. Penelitian (Sarlati, Hashemi, dan Mozaffari, 2015) mengadopsi fitur komunitas untuk mendeteksi *spammer* pada *Facebook*. Penelitian tersebut juga menggunakan berbagai fitur selain komunitas dan menggunakan pemilihan fitur *Principal Component Analysis* untuk mengurangi volume fitur yang digunakan. Majalah (Chen dkk, 2016) mengatakan bahwa koordinasi *spammer* mempersulit untuk dilakukan pendeteksian. Penelitian (Bindu dkk., 2018) menemukan bahwa *spammer* bekerja secara kolektif untuk menyebarkan *spam* dan menghindari teknik deteksi *spammer* pada *Twitter*.

Pada penelitian ini diusulkan pengembangan ekstraksi fitur berdasarkan *hashtag* dan aktivitas komunitas untuk deteksi akun *spammer* pada *Twitter*. Anggota komunitas *spammer* akan menggunakan *hashtag* untuk meningkatkan kepopuleran dan aktivitas yang dilakukan. Fitur yang banyak dari komunitas dan anggota komunitas akan dilakukan seleksi fitur, sehingga diperoleh fitur terbaik

untuk mendeteksi. Deteksi *spammer* diharapkan mampu mendeteksi *account spammer* pada *Twitter*, sehingga *Twitter* dapat meningkatkan kredibilitas informasi yang ada.

1.2 Perumusan Masalah

Rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana mendeteksi komunitas yang ada pada media sosial *Twitter*?
2. Bagaimana cara mendapatkan fitur *hashtag* dan aktivitas komunitas?
3. Bagaimana cara seleksi fitur untuk mendapatkan fitur yang terbaik?
4. Bagaimana mendeteksi akun *spammer* pada *Twitter*?

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah pengembangan ekstraksi fitur berdasarkan *hashtag* dan aktivitas komunitas untuk deteksi akun *spammer* pada *Twitter*.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat membantu untuk mengetahui akun *spammer* pada *Twitter* dan memberikan kredibilitas informasi yang terdapat di *Twitter*, sehingga informasi di *Twitter* dapat digunakan untuk berbagai hal seperti penelitian, politik, dan lain sebagainya.

1.5 Kontribusi Penelitian

Kontribusi penelitian ini adalah pengembangan ekstraksi fitur berdasarkan *hashtag* dan aktivitas komunitas untuk deteksi akun *spammer* pada *Twitter*. Hal ini dikarenakan *spammer* membentuk komunitas untuk meningkatkan kredibilitas dan *spammer* akan menggunakan *hashtag* untuk meningkatkan kepopuleran dan aktivitas yang dilakukan baik *tweet*, URL, ataupun yang lainnya.

1.6 Batasan Masalah

Untuk menghindari meluasnya permasalahan, penelitian ini memiliki batasan permasalahan yaitu:

1. Data yang digunakan adalah data dari akun *Twitter* @kpk_ri.
2. Data diambil dari tanggal 01 Agustus 2018 hingga 10 September 2019.
3. Data menggunakan bahasa indonesia
4. Tidak menggunakan *tweet* awal yang berasal dari @kpk_ri

[Halaman ini sengaja dikosongkan]

BAB 2

KAJIAN PUSTAKA

Pada bab ini akan dijelaskan mengenai referensi terkait penyelesaian masalah dalam penelitian. Beberapa hal yang dijelaskan pada bab ini yaitu tentang *Twitter*, deteksi komunitas, *label propagation*, ekstraksi fitur, kajian penelitian deteksi *account spammer* pada *Twitter*, metode *recursive feature elimination*, metode deteksi akun *spammer*, dan metode evaluasi.

2. 1. Data

Data merupakan fakta atau informasi *empiric* yang dikumpulkan oleh peneliti dengan tujuan untuk memecahkan masalah atau menjawab pertanyaan dari penelitian yang dilakukan. Data penelitian dapat diambil dari berbagai sumber yang diperoleh dengan berbagai cara selama kegiatan penelitian dilakukan. Data dibagi menjadi 2 berdasarkan sumbernya, yaitu data primer dan data sekunder.

Data primer merupakan data yang diambil secara langsung oleh peneliti. Data primer disebut juga data *up to date*. Contoh data primer yaitu proses mengambil data secara langsung pada *Twitter*, wawancara, atau lain sebagainya. Data sekunder merupakan data yang dikumpulkan oleh peneliti dari berbagai sumber yang ada seperti laporan, jurnal, dan sebagainya.

2.1.1. *Twitter*

Twitter merupakan salah satu jejaring sosial *online* paling populer dan paling cepat berkembang yang didirikan pada tahun 2006. *Twitter* telah muncul sebagai *platform microblogging* paling populer dimana pengguna dapat berbagi berita, media, meme, pandangan, dan pembaruan dalam bentuk *tweet*. *Tweet* adalah tulisan yang berisi teks dan URL HTTP terbatas hingga 280 karakter (Rozen, 2017). Banyak mesin pencari populer seperti *Yahoo*, *Microsoft Bing*, dan *Google* menggunakan aliran *Twitter* untuk melacak pembaruan langsung dari

kejadian di seluruh dunia untuk memberikan informasi secara praktis tanpa penundaan.

Twitter memiliki kebijakan untuk melarang kegiatan *spam* pada aplikasinya (*Twitter*, 2019). Hal ini dilakukan karena *Twitter* ingin memberikan layanan yang aman, handal dalam memberikan informasi, dan memberikan kebebasan para pengguna untuk saling berkomunikasi. Beberapa contoh pelanggaran dalam kebijakan *Twitter* yaitu *spam* dengan motivasi *komersial*, dimana biasanya diarahkan kepada *web* lain baik produk, layanan, ataupun program dengan menggunakan URL. *User* dilarang untuk membuat keterlibatan yang tidak sah dengan membuat akun atau konten yang terlihat aktif atau populer. *User* dilarang untuk melakukan koordinasi untuk mempengaruhi percakapan dengan menggunakan lebih dari satu akun (akun palsu), otomatisasi, ataupun penggunaan *script*. Beberapa hal yang dianggap mengganggu informasi oleh *Twitter* yaitu *user* yang mengoperasikan akun lebih dari satu. hal ini ditandai dengan data yang identik atau konten yang sangat mirip. *User* menjalankan beberapa akun yang terlibat komunikasi yang bertujuan menaikkan ataupun memanipulasi suatu *tweet* dengan tujuan untuk membuat pembahasan menjadi menonjol. *User* melakukan pengiriman konten yang duplikat seperti *tweet* dan *hashtag* yang memiliki tingkat kemiripan tinggi. Contoh hal yang mengganggu informasi pada *Twitter* atau *spam* diperlihatkan pada **Gambar 2.1**.

2.1.2. Label Propagation

Algoritma pembelajaran *semi-supervised* bergantung pada geometri data yang disebabkan oleh contoh berlabel dan tidak berlabel untuk meningkatkan pada metode yang diawasi yang hanya menggunakan data berlabel (Chapelle, Scholkopf, dan Zien, 2006). Geometri ini dapat secara alami diwakili oleh grafik empiris $G(V, E)$ dimana *node* $V\{1..n\}$ mewakili data pelatihan dan tepi E mewakili kesamaan di antara mereka. Kesamaan ini diberikan oleh matriks bobot $W: W_{ij}$ bukan-nol jika X_i dan X_j adalah "*neighbours*", yaitu tepi (i, j) berada di E

(dibobotkan oleh W_{ij}). Matriks bobot W dapat berupa, misalnya matriks tetangga k -terdekat: $W_{ij}=1$ jika x_i adalah di antara k -nearest neighbours x_j atau sebaliknya (dan 0 jika tidak). Matriks bobot tipikal lainnya diberikan oleh kernel *Gaussian* dengan lebar σ , dinotasikan dengan Persamaan 2.1.

$$W_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \quad (2.1)$$

Secara umum, W_{ij} diberikan oleh fungsi positif simetris W_X (tergantung pada kumpulan data $X=(x_1, \dots, x_n)$) oleh $W_{ij}=W_X(x_i, x_j) \geq 0$.

2. 2. Deteksi Komunitas

Suatu komunitas adalah sekelompok simpul grafik yang “termasuk bersama” menurut beberapa kriteria yang didefinisikan secara tepat yang dapat diukur. Banyak definisi telah diajukan, secara umum komunitas didefinisikan sebagai kelompok simpul sedemikian rupa sehingga kerapatan antara simpul kelompok lebih tinggi daripada kerapatan rata-rata dalam grafik. Dalam teks atau modul istilah, kluster juga digunakan ketika merujuk ke komunitas (Fortunato dan Castellano, 2012).

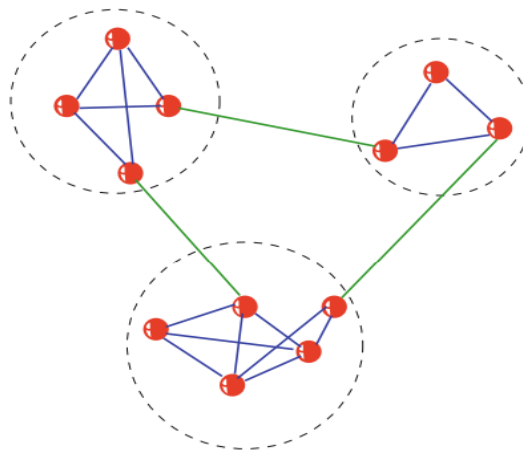
Contoh *graph* 3 komunitas dapat dilihat seperti pada **Gambar 2.2**, setiap komunitas dilingkari dengan garis lingkaran putus-putus dan antar komunitas memiliki keterkaitan satu dengan yang lainnya. Proses deteksi komunitas menggunakan aplikasi *gephi*. Metode pada *gephi* yang digunakan yaitu metode *louvain*. Metode *louvain* merupakan metode untuk ekstraksi komunitas pada jejaring dengan ukuran besar. Proses *louvain* dilakukan dengan mengoptimasi *modularity*.

Hal ini menunjukkan keunggulan dari waktu komputasi (Blondel dkk, 2008). Metode *louvain* melakukan optimasi terhadap modularitas dengan perubahan komunitas secara lokal. Setelah mendapatkan modularitas yang tertinggi, komunitas awal dilakukan agregasi untuk mendapatkan komunitas

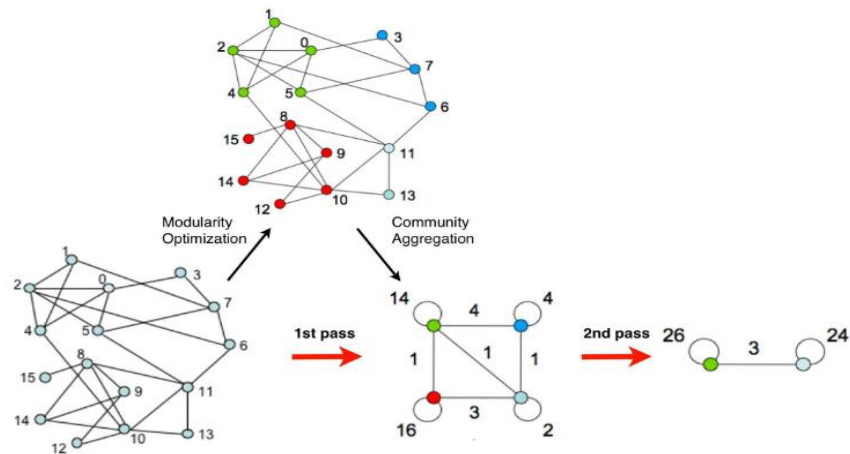
terbaru dan dianggap sebagai satu pass. Gambaran metode *louvain* dapat dilihat pada **Gambar 2.3**.

Id	Tweet
T1	RT @user: Retweet to win up to 121+ followers must be following me 🙏
T2	Retweet this for 81+ free follows 🌸 🌿
T3	Retweet for 125 free follows 🌿 '\n'Retweet and Fav 🍷 this if you have my post notifications on! 🍷 For 125 free followers 🍷 🍷
T4	Watch and like this video for free 80 followers 🍷 url
T5	Retweet to win up to 130+ free followers 🍷 ' '@user
T6	RT @user: Retweet this to gain followers faster 🍷 🍷 🍷
T7	Follow everyone who FAV this 🍷
T8	@user @user @user @user @user @user follow everyone who likes this 🍷 #SolarEclipse2017 \

Gambar 2.1 Contoh *spam* pada *Twitter* (Inuwa-Dutse dkk., 2018)



Gambar 2.2 Contoh *graph* 3 komunitas (Fortunato dan Castellano, 2012)



Gambar 2.3 Ilustrasi metode *louvain* (Blondel dkk., 2008)

Blondel dkk. memperkenalkan algoritma yang menemukan partisi modularitas tinggi dari jaringan besar dalam waktu singkat dan membuka struktur komunitas lengkap untuk jaringan, sehingga memberikan akses ke resolusi yang berbeda dari deteksi komunitas. Berlawanan dengan semua algoritma deteksi komunitas lainnya, batas ukuran jaringan yang dihadapi dengan algoritma ini adalah karena kapasitas penyimpanan yang terbatas daripada waktu komputasi yang terbatas.

Algoritma dibagi dalam dua fase yang dilakukan secara berulang. Asumsikan bahwa dimulai dengan jaringan N node. Pertama, algoritma menetapkan komunitas yang berbeda untuk setiap *node* jaringan. Jadi, dalam partisi awal ini terdapat komunitas sebanyak *node*. Kemudian, untuk setiap simpul i , algoritma mempertimbangkan tetangga j dari i , dan algoritma akan mengevaluasi perolehan modularitas yang akan terjadi dengan menghapus i dari komunitasnya dan dengan menemukannya di komunitas j . *Node* i kemudian ditempatkan pada komunitas yang mendapatkan keuntungan maksimum, tetapi hanya jika kenaikan ini positif. Jika tidak ada keuntungan positif yang mungkin terjadi, *node* tetap berada pada komunitas aslinya. Proses ini diterapkan berulang kali dan berurutan untuk semua *node* sampai tidak ada perbaikan lebih lanjut yang

dapat dicapai, dan fase pertama kemudian selesai. Algoritma menekankan fakta bahwa sebuah simpul mungkin, dan sering dipertimbangkan beberapa kali. Fase pertama ini berhenti ketika maksimal lokal dari modularitas tercapai, yaitu ketika tidak ada gerakan individu yang dapat meningkatkan modularitas. *Output* dari algoritma tergantung pada urutan di mana *node* dipertimbangkan. Hasil awal pada beberapa kasus uji tampaknya menunjukkan bahwa *node* tidak memiliki pengaruh signifikan terhadap modularitas yang diperoleh. Namun *node* dapat mempengaruhi waktu perhitungan.

Bagian efisiensi algoritma dihasilkan dari fakta bahwa gain dalam modularitas ΔQ yang diperoleh dengan memindahkan simpul terisolasi i ke dalam komunitas C dapat dengan mudah dihitung dengan Persamaan 2.2.

$$\Delta Q = \left[\frac{\sum in + K_{i,in}}{2m} - \left(\frac{\sum tot + K_i}{2m} \right)^2 \right] - \left[\frac{\sum in}{2m} - \left(\frac{\sum tot}{2m} \right)^2 - \left(\frac{K_i}{2m} \right)^2 \right] \quad (2.2)$$

Dimana $\sum in$ adalah jumlah dari bobot tautan di dalam C , $\sum tot$ adalah jumlah dari bobot insiden tautan ke simpul di C , k_i adalah jumlah bobot dari insiden tautan ke simpul i , $k_{i,in}$ adalah jumlah dari bobot tautan dari i ke simpul di C dan m adalah jumlah dari bobot semua tautan di jaringan. Ekspresi yang sama digunakan untuk mengevaluasi perubahan modularitas ketika *node* dihapus dari komunitasnya. Dalam praktiknya, algoritma mengevaluasi perubahan modularitas dengan mengeluarkan i dari komunitasnya dan kemudian memindahkannya ke komunitas tetangga.

Algoritma fase kedua terdiri dari membangun jaringan baru yang simpulnya adalah komunitas yang ditemukan pada fase pertama. Untuk melakukannya, bobot tautan antara simpul baru diberikan dengan jumlah bobot tautan antara simpul dalam dua komunitas yang sesuai. Tautan antara simpul-simpul dari komunitas yang sama mengarah ke *loop* dirinya untuk komunitas ini pada jaringan baru. Setelah fase kedua ini selesai, maka dimungkinkan untuk menerapkan kembali fase pertama dari algoritma ke jaringan berbobot yang dihasilkan dan untuk beralih. Hal tersebut akan ditunjukkan dengan "*pass*" kombinasi dari dua fase ini.

Dengan konstruksi, jumlah komunitas meta berkurang pada setiap *pass*, dan sebagai konsekuensinya sebagian besar waktu komputasi digunakan pada *pass* pertama. Lintasan diulang (lihat **Gambar 2.3**) sampai tidak ada lagi perubahan dan maksimum modularitas tercapai. Algoritma ini mengingatkan pada sifat serupa dari jaringan kompleks dan secara alami menggabungkan gagasan hierarki, karena komunitas-komunitas dibangun selama proses. Ketinggian hierarki yang dibangun ditentukan oleh jumlah lintasan dan umumnya sejumlah kecil.

Algoritma sederhana ini memiliki beberapa keunggulan. Pertama, langkah-langkahnya intuitif serta mudah diimplementasikan, dan hasilnya tidak diawasi. Selain itu algoritma ini sangat cepat, yaitu simulasi komputer pada jaringan modular *ad-hoc* besar menunjukkan bahwa kompleksitasnya linear pada data *typical* dan *sparse*. Hal ini disebabkan oleh fakta bahwa kemungkinan keuntungan dalam modularitas mudah untuk dihitung dengan Persamaan 2.1 di atas, dan bahwa jumlah komunitas berkurang secara drastis setelah hanya beberapa lintasan sehingga sebagian besar waktu berjalan terkonsentrasi pada iterasi pertama. Apa yang disebut masalah batas resolusi modularitas juga tampaknya dapat diatasi berkat sifat intrinsik *multi-level* dari algoritma. Memang, sudah terkenal bahwa optimasi modularitas gagal mengidentifikasi komunitas yang lebih kecil dari skala tertentu, sehingga mendorong batas resolusi pada komunitas yang terdeteksi oleh pendekatan optimasi modularitas murni. Pengamatan ini hanya relevan sebagian dalam kasus karena fase pertama dari algoritma ini melibatkan perpindahan *node* tunggal dari satu komunitas ke komunitas lainnya. Akibatnya, probabilitas dua komunitas yang berbeda dapat digabung dengan memindahkan *node* satu per satu sangat rendah. Komunitas-komunitas ini dapat digabungkan dalam operan selanjutnya setelah blok *node* telah dikumpulkan. Namun, algoritma menyediakan dekomposisi jaringan ke komunitas untuk berbagai tingkat organisasi.

Tabel 2.1 Fitur Jaringan (*graph*)

No	Fitur	Keterangan
1	<i>Indegree</i>	Jumlah <i>graph</i> yang masuk ke dalam <i>node</i>
2	<i>Eigenvector centrality</i>	Ukuran pengaruh suatu <i>node</i> terhadap suatu jaringan

Tabel 2.2 Fitur Komunitas dan Keterangan

No	Fitur	Keterangan
1	Jumlah <i>indegree</i>	Jumlah <i>indegree</i> pada satu komunitas
2	Jumlah anggota komunitas	Jumlah akun pada satu komunitas

2.3. Ekstraksi Fitur

Platform Twitter memfasilitasi koneksi global dan interaksi beragam kepada pengguna (Rosengrendkk, 2011). Berbagai atribut dibentuk pada *Twitter* baik individu, koneksi, ataupun komunitas. Setiap atribut pada *Twitter* dapat digunakan sebagai fitur untuk mendeteksi *spammer*. Fitur-fitur tersebut satu dengan yang lainnya saling berkaitan. Fitur – fitur dikategorikan menjadi 3, yaitu:

1. Fitur jaringan (*graph*) pada penelitian (Masood dkk., 2019) diantaranya yaitu *indegree*. Keterangan fitur jaringan dapat dilihat pada **Tabel 2.1**.
2. Fitur komunitas pada penelitian (Bhat dan Abulaish, 2013) dan (Sarlati dkk., 2015) diantaranya jumlah *indegree*, dan jumlah anggota komunitas. Keterangan fitur komunitas dapat dilihat pada **Tabel 2.2**.
3. Fitur akun pada penelitian (Inuwa-Dutse dkk., 2018), (Masood dkk., 2019) dan (Bindu dkk., 2018) diantaranya yaitu umur akun, jumlah *follower*, jumlah *following*, jumlah status, jumlah digit nama, rata-rata panjang *tweet*, panjang *username*, panjang *screen name*, rasio URL, rasio *mention*, *LexRichWithUU*,

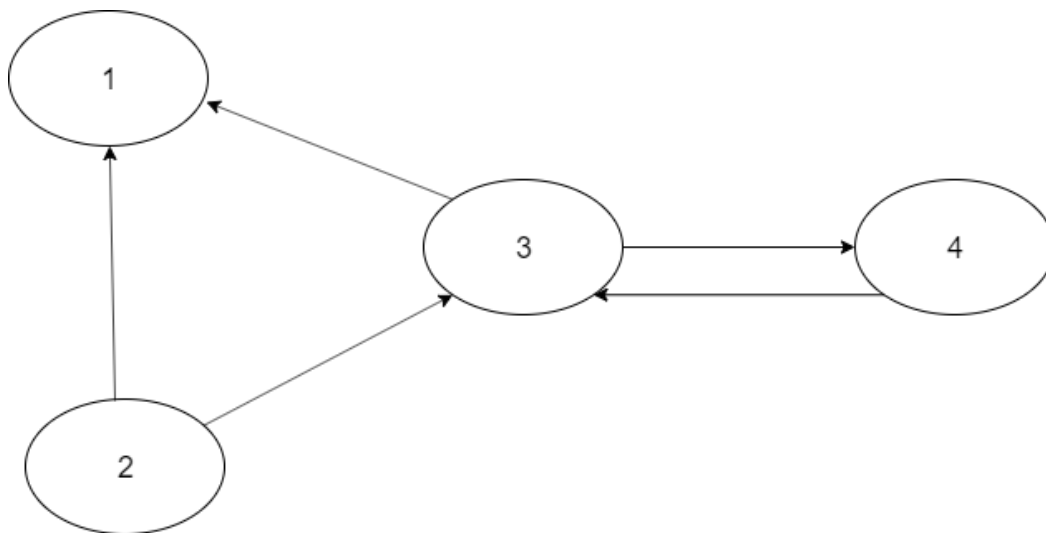
Tabel 2.3 Fitur Pada Akun Twitter

No.	Fitur	Keterangan
1	Umur akun	Hari sejak pembuatan akun hingga tanggal pengumpulan.
2	Jumlah <i>follower</i>	Jumlah <i>follower</i> pada meta data akun
3	Jumlah <i>following</i>	Jumlah <i>following</i> pada meta data akun
4	Jumlah status	Jumlah <i>post</i> yang dibuat dari awal <i>post</i> hingga pengumpulan.
5	Jumlah <i>digit</i> nama	Banyaknya angka pada <i>screen name</i>
6	Rata-rata panjang <i>tweet</i>	Jumlah semua karakter <i>tweet</i> dibagi banyaknya <i>tweet</i>
7	Panjang <i>username</i>	Jumlah karakter pada <i>username</i>
8	Panjang <i>screen name</i>	Jumlah karakter pada <i>screen name</i>
9	Rasio URL	Jumlah karakter URL dibagi panjang <i>tweet</i>
10	Ratio <i>mention</i>	Jumlah karakter <i>mention</i> dibagi panjang <i>tweet</i>
11	<i>Similarity username</i> dan <i>screen name</i>	Tingkat kesamaan <i>username</i> dan <i>screen name</i>
12	<i>LexRichWithUU</i>	Tingkat kekayaan token pada <i>tweet</i>
13	Rasio <i>following</i>	Hasil <i>following</i> dibagi <i>follower</i>
14	Rasio <i>follower</i>	Hasil <i>follower</i> dibagi <i>following</i>
15	<i>Interestingness</i>	Hasil jumlah <i>like</i> dibagi jumlah status
16	Keaktifan akun	Hasil bagi jumlah status dan umur akun
17	<i>LexRichWithOutUU</i>	Tingkat kekayaan token tanpa <i>mention</i> dan URL pada <i>tweet</i>
18	Rasio nama	Hasil bagi <i>screen name</i> dan <i>username</i>
28	Rasio <i>Unique</i> URL	Jumlah <i>unique</i> URL dibagi total URL

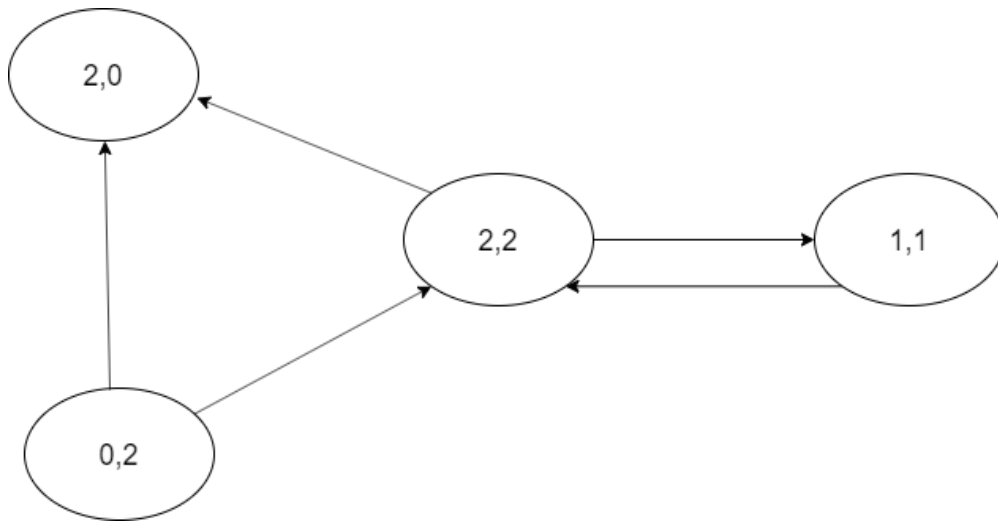
similarity username dan *screen name*, rasio *following*, rasio *follower*, *interestingness*, keaktifan akun, *LexRichWithOutUU*, jumlah *favorite*, dan rasio nama. Keterangan fitur akun dapat dilihat pada **Tabel 2.3**.

Indegree (ID) adalah banyaknya jaringan (*graph*) yang masuk kedalam *node*. $ID = \{ID_1, ID_2, ID_3 \dots ID_{ni}\}$. Sedangkan *outdegree (OD)* adalah banyaknya jaringan (*graph*) yang keluar dari *node* tersebut. $OD = \{OD_1, OD_2, OD_3, \dots OD_{no}\}$. **Gambar 2.4** merupakan contoh jaringan yang saling berkaitan antara satu *node* dengan yang lainnya.

Gambar 2.5 merupakan contoh hasil dari *indegree (ID)* dan *outdegree (OD)*. Hasil *node* 1 mendapatkan (*ID, OD*) yaitu (2,0). Nilai 2 merupakan jumlah banyaknya jaringan yang masuk kedalam *node* dan nilai 0 menandakan tidak adanya jaringan yang keluar dari *node* tersebut. Contoh lain *node* 3 menghasilkan *indegree, outdegree* sebesar (2,2). Nilai 2 awal merupakan jumlah *indegree* yang diperoleh *node* 3 dan nilai 2 akhir merupakan jumlah *outdegree* dari *node* 3 tersebut.



Gambar 2.4 Contoh Jaringan Dari *Node*



Gambar 2.5 Contoh Hasil *Indegree* dan *Outdegree*

Eigenvector centrality (C_E) merupakan ukuran pengaruh suatu *node* terhadap suatu jaringan. Jika sebuah simpul ditunjukkan oleh banyak simpul (yang juga memiliki *centralities* vektor yang tinggi) maka simpul tersebut akan memiliki *centralities* vektor yang tinggi. Dalam *centralities* vektor *Eigen* pertama-tama *centralities* setiap *node*, $C_E(x)$ diinisialisasi sebagai 1, maka aturan Persamaan 2.3 yang digunakan:

$$C_E(x) = \frac{1}{\lambda} \sum_{y \rightarrow x} w \cdot C_E(y) \quad (2.3)$$

Jumlah *indegree* (*JID*) merupakan total dari *indegree* seluruh anggota pada komunitas (K) tersebut. Jumlah *indegree* dinotasikan pada Persamaan 2.4.

$$JID = \{ \sum_{j \in K_i} ID_j \mid K_i \in K \} \quad (2.4)$$

Jumlah anggota komunitas (*JAK*) merupakan jumlah akun (A) yang berada pada satu komunitas (K) secara bersamaan. $A = \{A_1, A_2, A_3, \dots, A_{na}\}$. Jumlah anggota komunitas dinotasikan pada Persamaan 2.5.

$$JAK = \{n(K_i) \mid K_i \in K\} \quad (2.5)$$

Umur akun (*UA*) merupakan salah satu kriteria yang telah dibahas pada penelitian (Yang dkk., 2011). Penelitian tersebut menyampaikan bahwa *spammer*

kadang menggunakan akun baru dan akun palsu. Penelitian (Inuwa-Dutse dkk., 2018) juga melakukan analisis bahwa usia akun mempengaruhi aktivitas dan keaktifan dari akun tersebut. Sehingga dapat dicurigai sebagai *spammer*. Usia akun dihitung dengan mengurangkan antara waktu pengambilan (T_t) data dan waktu pembuatan (T_c) dari akun tersebut. Usia akun dinotasikan dengan Persamaan 2.6.

$$UA = T_t - T_c \quad (2.6)$$

Follower (FR) menurut *Twitter* merupakan teman-teman yang berlangganan atau mengikuti akun. $FR = \{FR_1, FR_2, FR_3, \dots, FR_{nfr}\}$. *Follower* secara natural pada akun baru sedikit dan sesuai dengan tingkat kepopuleran dari akun. Sedangkan, *following* merupakan akun-akun yang diikuti atau langganan. $FG = \{FG_1, FG_2, FG_3, \dots, FG_{nfg}\}$. Jumlah *follower* dan *following* menjadi salah satu kriteria *spammer* berdasarkan (Inuwa-Dutse dkk., 2018) yaitu, suatu akun baru akan memiliki perbedaan antara jumlah *follower* dan *following* yang dilakukannya. Jumlah *follower (JFR)* dan jumlah *following (JFG)* dapat dilihat langsung pada akun *Twitter*. Jumlah *follower* dan *following* dapat dinotasikan pada Persamaan 2.7 dan Persamaan 2.8.

$$JFR = n(FR), \forall A_i \in A \quad (2.7)$$

$$JFG = n(FG), \forall A_i \in A \quad (2.8)$$

Status (S) atau *tweet* pada akun *Twitter* merupakan kegiatan *post* yang dilakukan pemilik akun tersebut. $S = \{S_1, S_2, S_3, \dots, S_{ns}\}$. Jumlah status (JS) yang dilakukan oleh pengguna menjadi kriteria *spammer* jika berlebihan dalam menggunakannya. Hal ini dilihat pula umur akun *Twitter* tersebut. Jumlah status dinotasikan pada Persamaan 2.9.

$$JS = n(S), \forall A_i \in A \quad (2.9)$$

Digit (Dg) merupakan angka-angka yang terdapat pada bilangan atau kata. $Dg = \{Dg_1, Dg_2, Dg_3, \dots, Dg_{ndg}\}$. Jumlah *digit* nama (JDN) merupakan banyaknya

digit pada panjangnya *Screen name* (*SN*) akun *Twitter*. Jumlah *digit* nama dinotasikan dengan Persamaan 2.10.

$$JDN = n(Dg), Dg_i \subseteq SN \quad (2.10)$$

Panjang *tweet* merupakan banyaknya karakter (*k*) pada *tweet* (S_i). $k = \{k_1, k_2, k_3, \dots, k_{nk}\}$. Rata-rata panjang *tweet* (*RPT*) merupakan hasil bagi jumlah karakter *tweet* (*JKT*) dan jumlah status (*JSA*) pada akun. Rata-rata panjang *tweet* dinotasikan dengan Persamaan 2.12.

$$JKT = n(k), \forall S_i \in S \quad (2.11)$$

$$RPT = \frac{JKT}{JS} \quad (2.12)$$

Panjang *username* (*PU*) merupakan banyaknya karakter yang menyusun *username* (*US*) pada akun *Twitter*. Panjang *username* dinotasikan dengan Persamaan 2.13.

$$PU = n(k), k \subseteq US \quad (2.13)$$

Panjang *screen name* (*PSN*) merupakan banyaknya karakter yang menyusun *screen name* (*SN*) suatu akun. Panjang *screen name* dinotasikan dengan Persamaan 2.14.

$$PSN = n(k), k \subseteq SN \quad (2.14)$$

Rasio URL (*RU*) merupakan hasil pembagian dari karakter penyusun URL dan jumlah karakter penyusun *tweet*. Rasio URL dinotasikan dengan Persamaan 2.15.

$$RU = \frac{n(k)}{JKT}, k \subseteq URL \quad (2.15)$$

Rasio *mention* (*RM*) merupakan hasil pembagian dari jumlah karakter penyusun *mention* (*Mn*) dan panjangnya *tweet* suatu akun. Rasio *mention* dinotasikan dengan Persamaan 2.16.

$$RM = \frac{n(k)}{JKT}, k \subseteq Mn \quad (2.16)$$

Similarity username dan *screen name* merupakan tingkat kesamaan karakter antara *username* dan *screen name*. *Similarity username* dan *screen name* menggunakan *cosine similarity*.

LexRichWithUU (*LUU*) merupakan tingkat kekayaan dalam penggunaan token pada *tweet* yang dilakukan. Tingkat kekayaan yang rendah menunjukkan penggunaan kata yang berulang-ulang, sehingga akan berbeda dari topik yang dibahas. *LexRichWithUU* dinotasikan dengan Persamaan 2.17.

$$LUU = \frac{|total\ type|}{|total\ token|} \times 100 \quad (2.17)$$

Rasio *following* (*RFG*) merupakan rasio hasil pembagian dari jumlah *following* dengan jumlah *follower* pada akun *Twitter*. Rasio *following* dinotasikan dengan Persamaan 2.18.

$$RFG = \frac{JFG}{JFR} \quad (2.18)$$

Rasio *follower* (*RFR*) merupakan rasio hasil pembagian dari jumlah *follower* dengan jumlah *following* pada akun *Twitter*. Rasio *follower* dinotasikan dengan Persamaan 2.19.

$$RFR = \frac{JFR}{JFG} \quad (2.19)$$

Interestingness (*I*) merupakan tingkat kepercayaan akun *Twitter*. *Interestingness* dihitung dengan membagi jumlah *favorite* (*JFE*) dan jumlah status yang dilakukan akun *Twitter*. *Interestingness* dinotasikan dengan Persamaan 2.20.

$$I = \frac{JFE}{JS} \quad (2.20)$$

Keaktifan akun (*KA*) merupakan tingkat seringnya akun melakukan *posting* pada berandanya. Keaktifan akun dihitung dengan cara membagi jumlah status dengan umur akun. Keaktifan akun dinotasikan dengan Persamaan 2.21.

$$KA = \frac{JS}{UA} \quad (2.21)$$

LexRichWithOutUU (LOUU) merupakan tingkat kekayaan kata pada *tweet* dengan membuang *mention* dan URL. *LexRichWithOutUU* dinotasikan dengan Persamaan 2.22.

$$LOUU = \frac{\text{lexical words}}{\text{total number words}} \times 100 \quad (2.22)$$

Rasio nama (*RNA*) merupakan hasil dari pembagian antara panjang *screen name* dan panjang *username* yang terdapat pada suatu akun. Rasio nama dinotasikan dengan Persamaan 2.23.

$$RNA = \frac{PSN}{PU} \quad (2.23)$$

Rasio *Unique URL (RUU)* merupakan hasil bagi antara jumlah *unique URL* dengan jumlah total URL pada akun. Rasio *unique URL* dinotasikan dengan Persamaan 2.24.

$$RUU = \frac{\text{unique url}}{\text{jumlah url akun}} \quad (2.24)$$

2.4. Recursive Feature Elimination

Guyon dan Weston, 2019 memperkenalkan algoritma pemilihan fitur yang disebut *Support Vector Machine (SVM) - Recursive Feature Elimination (RFE)* yang telah dipatenkan olehnya. Pemilihan fitur dilakukan untuk menghilangkan variabel tidak informatif yang pada gilirannya harus mengarah pada kinerja generalisasi yang lebih baik, yaitu SVM harus mencapai kinerja klasifikasi yang lebih baik pada pola yang sebelumnya. Selain itu, serangkaian fitur yang berkurang juga dapat memberikan wawasan yang lebih baik tentang model yang mendasarinya untuk dipelajari dan percepatan komputasi. Manfaat lain adalah pengurangan biaya, dalam aplikasi biologis misalnya, dimana hanya sebagian kecil gen harus diukur untuk mendeteksi penyakit tertentu dengan akurasi yang sama seperti sebelumnya.

SVM RFE adalah aplikasi RFE yang menggunakan besaran bobot sebagai kriteria peringkat. Gambar 2.6 merupakan alur proses dari SVM RFE. Berikut ini algoritma SVM RFE.

1. *Input:*

Data *training* $X = [x_1, x_2, x_3, \dots, x_k]^T$

Label kelas : $y = [y_1, y_2, y_3, \dots, y_k]^T$

2. *Proses:*

Subset fitur: $s = [1, 2, \dots, n]$

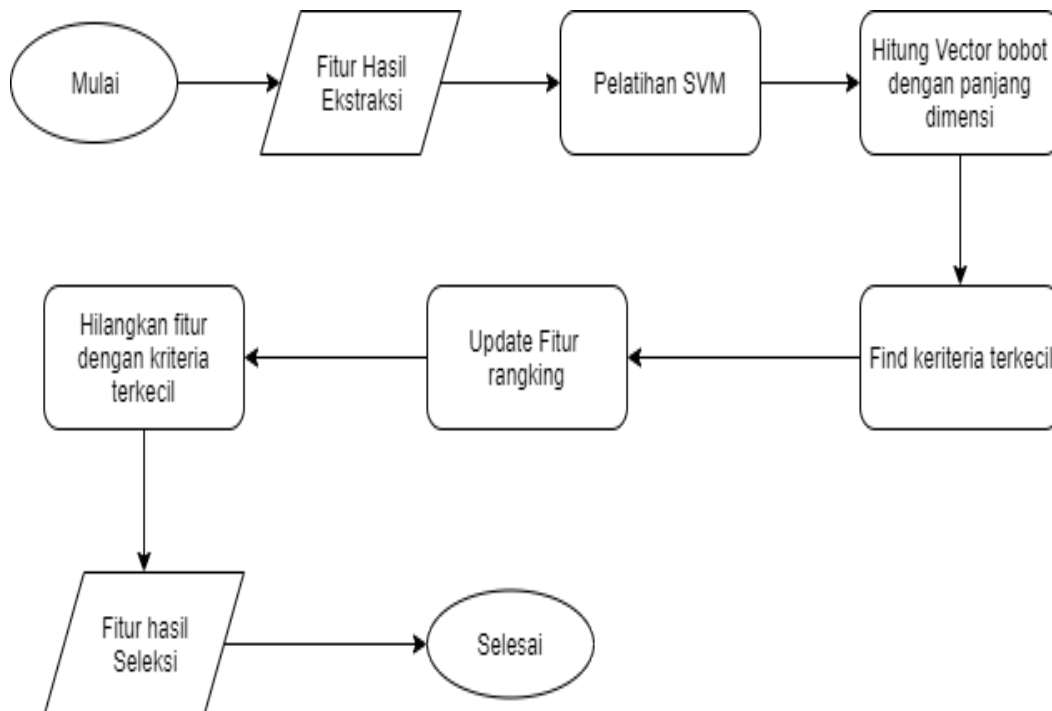
List fitur peringkat: $r = []$

Perulangan hingga mencapai $s = []$

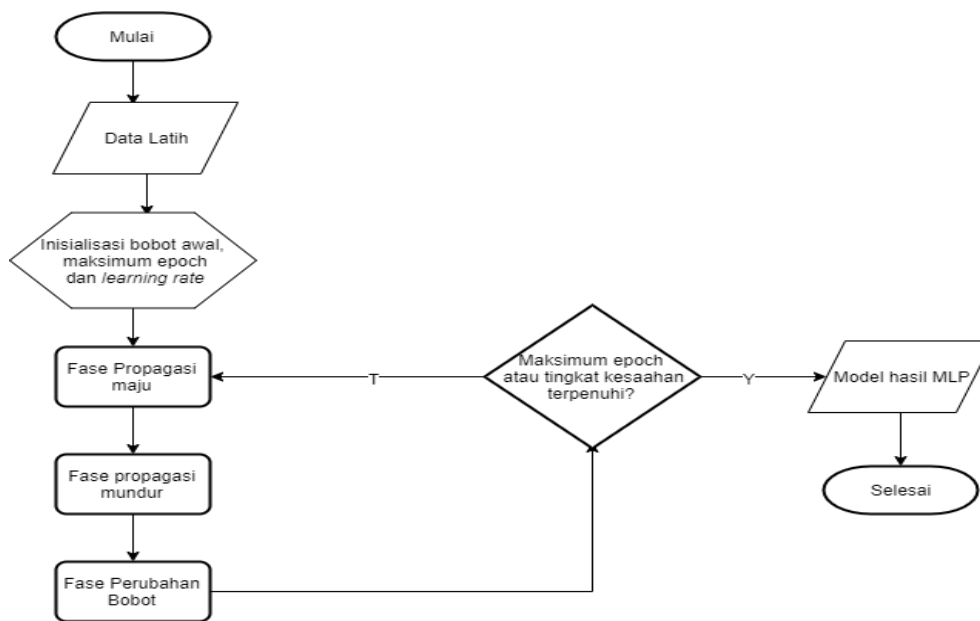
Batasi contoh pelatihan untuk indeks fitur yang baik: $X = x_0(:s)$

Pelatihan: $a = svm_{train}(x, y)$

Hitung vektor bobot dengan panjang dimensi (s): $w = \sum_k a_k y_k x_k$



Gambar 2.6 Seleksi Fitur *Support Vector Machine Recursive Feature Elimination*



Gambar 2.7 Proses pelatihan *Multi-layer Perceptron*

Hitung kriteria ranking: $c_i = (w_i)^2$ untuk semua i

Temukan fitur dengan kriteria peringkat terkecil: $f = \text{argmin}(c)$

Perbarui daftar peringkat fitur: $r = [s(f), r]$

Hilangkan fitur dengan kriteria peringkat terkecil:

$$s = s(1: f - 1, f + 1: \text{length}(s))$$

3. *Output*:

List peringkat fitur r .

2.5. Deteksi Akun *Spammer*

Benevenuto dkk. 2010 memeriksa masalah deteksi *spammer* di *Twitter*. Benevenuto memeriksa kumpulan besar data *Twitter* yang berisi lebih dari 5400 juta pengguna, 1.8 miliar *tweet*, dan 1.9 miliar tautan dikumpulkan. Setelah itu, sejumlah fitur yang terkait dengan konten *tweet* dan karakteristik pengguna dikenali untuk mendeteksi *spammer*. Fitur-fitur ini dianggap sebagai karakteristik proses pembelajaran mesin untuk mengkategorikan pengguna, yaitu untuk mengetahui apakah mereka *spammer* atau tidak. Untuk mengenali pendekatan

deteksi *spammer* di *Twitter*, koleksi berlabel pra-klasifikasi *spammer* dan *non-spammer* telah dilakukan. *Twitter* telah diluncurkan untuk mengumpulkan ID pengguna, yaitu sekitar 80 juta. *Twitter* mengalokasikan ID numerik untuk setiap pengguna yang secara khusus mengidentifikasi profil setiap pengguna. Berikutnya, langkah-langkah tersebut diambil yang diperlukan untuk pembangunan koleksi berlabel dan memperoleh berbagai properti yang diinginkan. Dengan kata lain, langkah-langkah yang penting diperiksa untuk mengembangkan koleksi pengguna yang dapat diberi label sebagai *spammer* atau *non-spammer*. Pada akhirnya, atribut pengguna diidentifikasi berdasarkan perilaku mereka. Misalnya, dengan siapa mereka berinteraksi dan berapa frekuensi interaksi mereka. Untuk mengkonfirmasi hal ini, fitur-fitur dari pengguna koleksi berlabel telah diperiksa. Dua set atribut yang dipertimbangkan yaitu atribut konten dan atribut perilaku pengguna, untuk membedakan satu pengguna dengan pengguna yang lain. Atribut konten memiliki properti dari kata-kata *tweet* yang diposting oleh pengguna yang mengumpulkan fitur yang relevan dengan cara pengguna menulis *tweet*. Di sisi lain, atribut perilaku pengguna mengumpulkan fitur tertentu dari perilaku pengguna dalam konteks frekuensi *posting*, interaksi, dan dampak pada *Twitter*. Atribut berikut dianggap sebagai karakteristik pengguna, yang meliputi jumlah total *follower* dan *following*, usia akun, jumlah *mention*, berapa kali pengguna *retweet*, jumlah *tweet* yang diterima, dan waktu rata-rata di antara *tweet* pengguna, dan rata-rata *tweet* harian dan mingguan. Keseluruhan 23 atribut dari perilaku pengguna telah dipertimbangkan. Hasil metodologi yang diusulkan menunjukkan bahwa bahkan dengan set atribut yang dibedakan, kerangka kerja ini mampu mendeteksi *spammer* dengan frekuensi tinggi.

Selain itu, penelitian (Eshraqi, Jalali, dan Moattar, 2015) ditentukan berbagai fitur untuk mendeteksi *spam* dan kemudian dengan bantuan algoritma pengelompokan berbasis *stream*, mengenali *tweets spam*. Beberapa akun pengguna dipilih dari berbagai *dataset* dan setelah itu *tweet* acak dipilih dari akun.

Tweet kemudian dikategorikan sebagai *spam* dan *non-spam*. Para penulis mengklaim bahwa algoritma ini dapat membagi data menjadi *spam* dan *non-spam* dengan akurasi tinggi, dan *tweet* palsu mungkin dikenali dengan akurasi dan presisi tinggi. Berbagai fitur dapat digunakan untuk menentukan *spam*. Misalnya, fitur berdasarkan grafik adalah keadaan dimana *Twitter* dibentuk sebagai model sosial grafik. Jika jumlah pengikut rendah dibandingkan dengan jumlah yang diikuti, maka kredibilitas akun rendah dan kemungkinan akun tersebut adalah *spam* relatif tinggi. Selain itu, fitur berdasarkan konten mencakup reputasi *tweet*, tautan HTTP, sebutan dan balasan, dan topik yang sedang hangat. Untuk fitur waktu, jika banyak *tweet* dikirim oleh akun pengguna dalam *interval* waktu tertentu, maka itu adalah akun *spam*. *Dataset* penelitian terdiri dari 50.000 akun pengguna. Pendekatan mengidentifikasi *spammer* dan *tweet* palsu dengan akurasi tinggi.

Chen dkk. 2015 melakukan evaluasi algoritma pembelajaran mesin untuk mendeteksi *tweets spam*. Para penulis menganalisis dampak dari berbagai fitur pada kinerja deteksi *spam*, misalnya: (i) rasio *spam* terhadap *non-spam*, (ii) ukuran *dataset* pelatihan, (iii) data terkait waktu, (iv) diskritisasi faktor, dan (v) pengambilan sampel data. Untuk mengevaluasi deteksi, pertama, sekitar 600 juta *tweet* publik dikumpulkan dan kemudian penulis menerapkan sistem reputasi *web mikro trend* untuk mengidentifikasi *tweet spam* sebanyak mungkin. Sebanyak 12 fitur ringan juga dipisahkan untuk membedakan *tweet non-spam* dan *spam* dari *dataset* yang diidentifikasi ini. Karakteristik fitur yang diidentifikasi diwakili oleh angka cdf. Fitur-fitur ini dipahami oleh klasifikasi *spam* berbasis pembelajaran mesin, yang kemudian digunakan dalam percobaan untuk mengevaluasi deteksi *spam*. Empat *dataset* sampel untuk mereproduksi skenario yang berbeda. Karena tidak ada set data yang tersedia untuk tugas ini, beberapa set data digunakan dalam penelitian sebelumnya. Setelah mengidentifikasi *tweets spam*, 12 fitur dikumpulkan. Fitur-fitur ini dibagi menjadi dua kelas, yaitu fitur berbasis pengguna dan fitur berbasis *tweet*. Fitur berbasis pengguna diidentifikasi melalui

berbagai objek seperti usia akun dan jumlah favorit pengguna, daftar, dan *tweet*. Fitur berbasis pengguna yang diidentifikasi diuraikan dari struktur JSON. Di sisi lain, fitur berbasis *tweet* termasuk jumlah (i) *retweet*, (ii) *tagar*, (iii) sebutan pengguna, dan (iv) URL.

Deteksi akun *spammer* memiliki data yang tidak seimbang. Data *legitimate* jauh lebih besar dibandingkan data *spammer*. Proses penyeimbangan data *training* dilakukan dengan *oversampling* data. Data *training* yang telah seimbang dilakukan deteksi akun *spammer* dengan menggunakan *multilayer perceptron* dan *support vector machine*.

2.5.1. *Oversampling Data*

Jumlah data tidak seimbang dikarenakan pada satu kelas memiliki jumlah yang lebih besar dari pada kelas lainnya. Kelas yang sedikit disebut kelas minor, sedangkan sebaliknya disebut kelas mayor. Data tidak seimbang memiliki pengaruh terhadap model *training* yang dihasilkan. *Training* tidak menghiraukan data yang tidak seimbang, sehingga model *training* terpengaruh dari kelas mayor (Chawla dkk., 2002).

Cara mengatasi data tidak seimbang dapat dilakukan dengan menggunakan metode *Synthetic Minority Oversampling Technique (SMOTE)*. Metode *Synthetic Minority Oversampling Technique (SMOTE)* diusulkan oleh (Chawla dkk., 2002). Metode tersebut menyeimbangkan data kelas minor agar sama dengan kelas mayor dengan melakukan pembangkitan. *Pseudo-Code* algoritma *smote* (Chawla dkk., 2002) yaitu:

Input: jumlah *sample* kelas *minor* (T), jumlah *smote* ($N\%$), jumlah tetangga terdekat (k).

Output: $(N/100) \times T$

- 1) (* Jika N kurang dari 100%, acak sampel kelas minor karena hanya persentase acak dari mereka yang akan dibangkitkan. *)

- 2) **if** $N < 100$
- 3) **then** random dari sampel T kelas minor
- 4) $T = (N/100) * T$
- 5) $N = 100$
- 6) **endif**
- 7) $N = (int) (N / 100) (* \text{Jumlah SMOTE diasumsikan dalam kelipatan integral } 100. *)$
- 8) $k = \text{jumlah tetangga terdekat}$
- 9) $numattrs = \text{jumlah atribut}$
- 10) *Sampel* [] []: array untuk sampel kelas minor asli
- 11) *newindex*: membuat hitungan jumlah sampel *generated* yang dihasilkan, diinisialisasi ke 0
- 12) *Sintetis* [] []: array untuk sampel sintetis (* Hitung k tetangga terdekat untuk setiap sampel kelas minor. *)
- 13) **for** $i \leftarrow 1 \text{ to } T$
- 14) hitung k tetangga terdekat untuk i , dan simpan index di *nnarray*
- 15) Populasi ($N, i, nnarray$) (* fungsi menghasilkan sintetis *)
- 16) **endfor**
- 17) **while** $N \neq 0$
- 18) Memilih angka *random* antara 1 dan k , yang disebut nn . Pada tahap ini memilih 1 k tetangga terdekat dari i
- 19) **for** $attr \leftarrow 1 \text{ to } numattrs$
- 20) hitung: $dif = \text{sampel}[nnarray[nn]][attr] - \text{sampel}[i][attr]$
- 21) hitung: $gap = \text{random}$ angka antara 0 dan 1
- 22) $\text{Synthetic}[newindex][attr] = \text{sampel}[i][attr] + gap * dif$
- 23) **endfor**
- 24) $newindex++$
- 25) $N = N - 1$
- 26) **endwhile**

27) return (* akhir dari populasi. *)

Akhir dari *Pseudo-Code*.

2.5.2. *Multi-layer Perceptron*

Multi-layer perceptron adalah metode untuk melakukan pemecahan terhadap permasalahan yang rumit yang metodenya menggunakan sistem pelatihan yang terawasi (Siang, 2005). Maksud dari terawasi adalah ketika proses pelatihan berdasarkan target yang diinginkan (label). Menurut Fauset tahun 1994 *multi-layer perceptron* adalah jaringan syaraf tiruan *multilayer* yang terdiri dari *layer* inputan, *layer hidden* dan *layer output*.

Menurut (Siang, 2005) *multi-layer perceptron* memiliki tiga fase pelatihan. Fase pelatihan seperti yang ditunjukkan pada Gambar 2.7 yaitu:

1) Fase Propagasi Maju

Fase propagasi maju adalah setiap signal masukan akan dihitung maju ke *hidden layer* hingga ke *layer output* dengan menggunakan fungsi aktivasi yang telah ditentukan sebelumnya.

2) Fase Propagasi Mundur

Fase propagasi mundur yaitu perhitungan kesalahan yang terjadi karena selisih di antara *output* dengan target yang telah ditentukan. Propagasi mundur dimulai dari garis yang terjadi hubungan langsung dengan unit *output*.

3) Fase Perubahan Bobot

Fase perubahan bobot adalah proses perubahan bobot yang memiliki fungsi untuk melakukan penurunan kesalahan yang terjadi. Ketiga fase di atas dilakukan secara berulang-ulang sampai pada perulangan yang terakhir, dimana akhir perulangan itu telah ditentukan.

Berikut ini merupakan tahapan dalam pelatihan.

Tahap 1 Inisialisasi

Melakukan inisialisasi bobot serta bias (dilakukan pengesetan dengan sembarang angka yang diambil secara acak (*random*). Serta melakukan inisialisasi maksimal iterasi, *learning rate* serta toleransi *error*.

Tahap 2 Melakukannya selama kondisi untuk berhenti belum terpenuhi.

Untuk kondisi berhenti dapat dilakukan dengan maksimum iterasi atau dengan toleransi *error* yang diberikan. Apabila kondisi telah sampai pada iterasi maksimum maka proses berhenti. Atau apabila kondisi toleransi *error* telah kurang atau telah sama maka proses berhenti.

Tahap 3 Unit *input* (X_1, X_2, \dots, X_i) mendapat *signal input* lalu melakukan penyebaran ke semua *hidden layer* yang ada.

Tahap 4 *Hidden* (Z_1, Z_2, \dots, Z_i) melakukan perhitungan *signal input* beserta bobot dan biasanya menggunakan Persamaan 2.25.

$$Z_{in_j} = V_0j + \sum_{i=1}^n x_i v_{ij} \quad (2.25)$$

Lalu dengan memakai fungsi aktivasi yang sudah ditentukan maka akan diperoleh *output* dari *hidden* yang ada. Persamaan yang digunakan yaitu Persamaan 2.26.

$$y_k = f(z_{inj}) \quad (2.26)$$

Tahap 5 *Output* (Y_1, Y_2, \dots, Y_i) melakukan perhitungan dari *signal hidden* bersama dengan bias dan bobotnya menggunakan Persamaan 2.27.

$$Z_{in_k} = V_0k + \sum_{j=1}^p z_j w_{jk} \quad (2.27)$$

Lalu dengan memakai fungsi aktivasi yang sudah ditentukan dari awal maka diperoleh *signal output*.

$$y_k = f(y_{ink}) \quad (2.28)$$

Tahap 6 Selanjutnya dilakukan perhitungan kesalahan antara target dengan *output* yang dikeluarkan dengan menggunakan metode *Mean Absolute Persentase Error*. Apabila belum terpenuhi syarat tersebut maka dilakukan perhitungan koreksi *error* (δ_k) menggunakan Persamaan 2.29.

$$\delta_k = (t_k - y_k) f'(y_{ink}) \quad (2.29)$$

Setelah mendapatkan koreksi *error* maka selanjutnya dilakukan perhitungan untuk perubahan bobot keluaran (Δw_{kj}) dengan menggunakan *learning rate* (α).

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (2.30)$$

Tahap 7 Perhitungan kesalahan pada *hidden* ($Z_1, Z_2 \dots Z_j$) dengan melakukan perhitungan bobot yang telah dikirim oleh unit *output*. Untuk kondisi pertama maka menggunakan Persamaan 2.31.

$$\delta_{inj} = \sum_{k=1}^n \delta_k w_{jk} \quad (2.31)$$

Lalu hasil dari persamaan di atas dilakukan perkalian dengan turunan dari fungsi aktivasi agar memperoleh faktor *error*. Persamaan yang digunakan yaitu Persamaan 2.32.

$$\delta_j = \delta_{inj} f'(z_{inj}) \quad (2.32)$$

Setelah mendapatkan nilai koreksi, maka proses selanjutnya dilakukan perhitungan nilai *update* bobot pada bobot *hidden* (Δv_{ij}).

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.33)$$

Tahap 8 *Output* ($Y_1, Y_2 \dots Y_k$) melakukan perubahan bobot dari setiap unit *hidden* dengan Persamaan 3.34.

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (2.34)$$

Begitu juga dengan *hidden* (Z_1, Z_2, \dots, Z_j) dilakukan perubahan bobot dengan Persamaan 2.35.

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad (2.35)$$

Tahap 9 Melakukan pemeriksaan kondisi berhenti.

2.5.2. Support Vector Machine

Support vector machine merupakan metode yang diperkenalkan oleh Cortes dan Vapnik tahun 1995. Cortes dan Vapnik mengusulkan metode ini dengan ide memetakan vektor *input* ke beberapa ruang fitur dimensi tinggi Z melalui beberapa pemetaan non-linear memilih apriori. Di ruang ini permukaan keputusan linier dibangun dengan properti khusus yang memastikan kemampuan generalisasi jaringan yang tinggi.

Support Vector Machine (SVM) adalah pengklasifikasi diskriminatif yang secara formal didefinisikan oleh *hyperplane*. Dengan kata lain, diberikan data pelatihan berlabel (*supervised learning*), algoritma menghasilkan *hyperplane* optimal yang mengkategorikan data baru. Dalam dua ruang dimensi *hyperplane* ini adalah garis yang membagi sebuah *hyperplane* menjadi dua bagian, dimana pada setiap kelas terletak di kedua sisi. *Hyperplane* diilustrasikan pada Gambar 2.8.

Fungsi diskriminasi untuk model diskriminatif *Support Vector Machine* seperti pada Persamaan 2.36.

Tabel 2.4 Kernel pada *Support Vector Machine*

No	Kernel	Fungsi
1	<i>Linear</i>	$K(x_i, x_j) = x_i \cdot x_j$
2	<i>Radial Basis Function (RBF)</i>	$K(x_i, x_j) = \exp(-\gamma x_i - x_j ^2), \gamma > 0$
3	<i>Polynomial</i>	$K(x_i, x_j) = (\gamma x_i^t x_j + r)^p, \gamma > 0$

$$g(x) = W^T \phi(x_i) + b \quad (2.36)$$

Dimana b merupakan bias, $\phi(x_i)$ merupakan ruang masukan dimensi fitur tinggi dan w merupakan bobot *vector*. Parameter tersebut dilakukan pembelajaran secara otomatis menggunakan *maximized margin principle* pada data latih.

$$\min \frac{1}{2} W^T W + C \sum_{i=1}^N \varepsilon_i \quad (2.37)$$

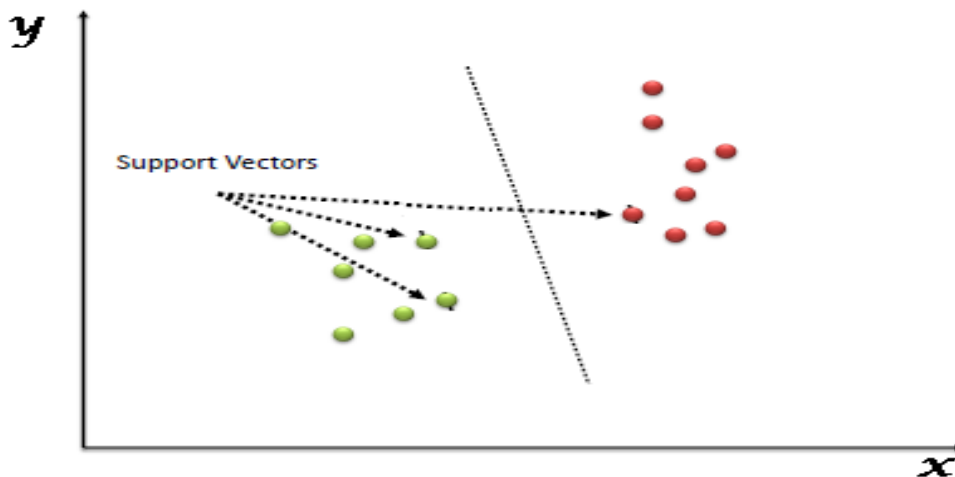
$$y_i(W^T \phi(x_i) + b) \geq 1 - \varepsilon_i \quad (2.38)$$

$$\varepsilon_i \geq 0 \quad N = 1, 2, \dots, N$$

C dan ε_i merupakan koefisien penalti dan *slack variable*. Dengan diperkenalkannya fungsi *kernel*, fungsi *discriminant* didefinisikan sebagai Persamaan 2.39.

$$g(x) = \sum_{i=1}^N \phi(x_i) y_i K(X_i, X) \quad (2.39)$$

Fungsi kernel merupakan kernel untuk mengatasi masalah non-linear yang sering terjadi di dunia nyata. Kernel digunakan pada setiap data untuk melakukan pemetaan data non-linear ke dalam ruang dimensi tinggi (*higher-dimensional space*). Fungsi kernel yang umum digunakan yaitu *linear*, *polynomial* dan *Radial*



Gambar 2.8 Ilustrasi *Hyperplane*

Basis Function (RBF), kernel-kernel tersebut yang ditunjukkan oleh Tabel 2.4.

2.5.3. Metode Evaluasi Deteksi Akun *Spammer*

Metode evaluasi yang digunakan yaitu *confusion matrix*. *Confusion matrix* merupakan penjelasan ringkas dari hasil pengujian klasifikasi. Jumlah hasil pengujian bernilai benar atau salah dirangkum menjadi satu pada setiap kelasnya. Beberapa istilah yang terdapat pada *confusion matrix* yaitu *true positive (TP)*, *false negative (FN)*, *true negative (TN)*, dan *false positive (FP)*. Istilah-istilah tersebut dapat menghasilkan perhitungan *accuracy*, *recall*, *precision* dan *g-mean*.

Akurasi merupakan hasil dari jumlah data yang benar dibagi dengan jumlah semua data yang dilakukan pengujian. Akurasi dinotasikan dengan Persamaan 2.40.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.40)$$

Recall merupakan hasil dari data *true positive* dibagi dengan seluruh jumlah data yang positif. *Recall* dinotasikan dengan Persamaan 2.41.

$$recall = \frac{TP}{TP+FN} \quad (2.41)$$

Presisi merupakan hasil dari data *true positive* dibagi dengan semua hasil prediksi yang positif. *Presisi* dinotasikan dengan Persamaan 2.42.

$$precision = \frac{TP}{TP+FP} \quad (2.42)$$

G-mean merupakan pengukuran yang mempertimbangan keseimbangan antara kelas positif dan kelas negatif (Japkowicz, 2013). *G-mean* dinotasikan dengan Persamaan 2.43.

$$G - mean = \sqrt{recall \times precision} \quad (2.43)$$

[Halaman ini sengaja dikosongkan]

BAB 3

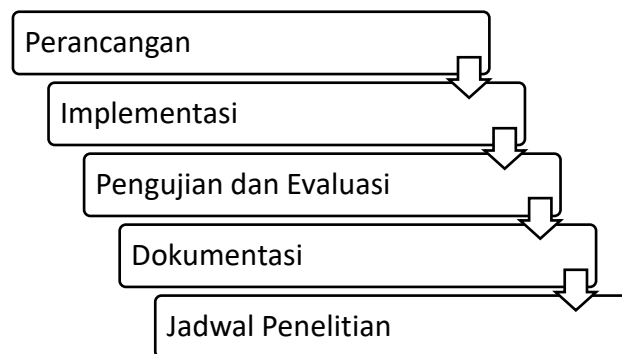
METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan alur metodologi yang digunakan pada penelitian. Metodologi yang dilakukan terdiri dari perancangan metode yang diajukan, implementasi, pengujian dan evaluasi, dan terakhir merupakan penyusunan laporan. **Gambar 3.1.** merupakan ilustrasi alur yang dilakukan.

Pada Sub-bab berikut ini akan dibahas tahapan metodologi penelitian secara lebih rinci.

3.1. Perancangan

Pada tahap perancangan akan dijelaskan langkah-langkah dari proses pengumpulan data hingga berhasil mendeteksi *account spammer*. Pada penelitian ini terdapat lima bagian utama seperti yang ditunjukkan pada **Gambar 3.2**, yaitu pengumpulan data dan pelabelan, deteksi komunitas, ekstraksi fitur, seleksi fitur dan deteksi *spammer*. **Gambar 3.2** merupakan perancangan secara keseluruhan yang akan dilakukan pada penelitian ini.



Gambar 3.1 Alur Metodologi Penelitian



Gambar 3.2 Perancangan Penelitian Secara Umum

3.1.1 Pengumpulan Data

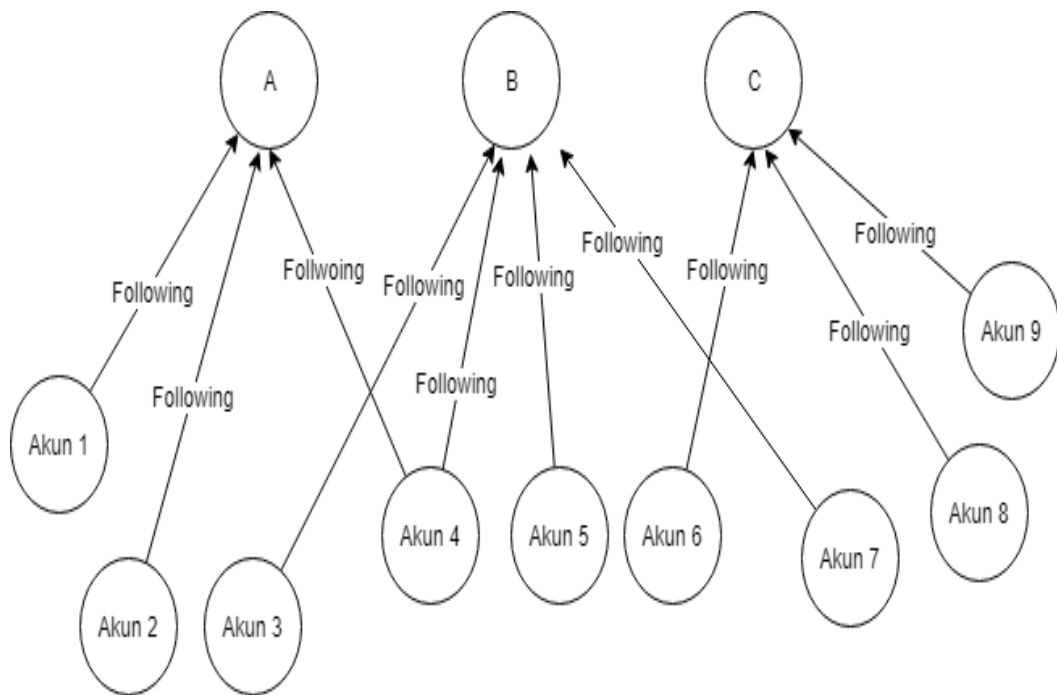
Pengumpulan data dilakukan pada *Twitter* dengan target minat *tweet* tentang “korupsi” pada *account* @kpk_ri. Pengumpulan data dari *Twitter* tidak menggunakan api resmi dari *Twitter*, akan tetapi menggunakan pustaka *python GetOldTweet3*. Karena jika menggunakan api resmi dari *Twitter* maka data yang diperoleh hanya 7 hari terakhir. Atribut *Twitter* yang akan diambil yaitu *date (datetime)*, *text (str)*, *username (str)*, *id (str)*, *to (str)*, *mentions (str)*, *hashtags (str)*, *retweets (int)*, *favourite (int)*, dan *permalink (str)*. Setelah pengumpulan data *tweet* tentang korupsi pada akun kpk, proses selanjutnya yaitu pengambilan *username* yang terlibat pada minat *tweet* “korupsi”. Kemudian dilakukan pengambilan *following* pada setiap *username* yang telah diperoleh. **Gambar 3.3** merupakan contoh proses pengambilan *following* pada *username*.

Proses pengambilan *following* pada *username* menggunakan pustaka *python twint*. Setelah mendapatkan semua *username* yang terhubung, selanjutnya mengambil data seluruh *username* yang diperlukan. Dari *username* yang telah diperoleh akan digunakan untuk membentuk komunitas pada tahap selanjutnya. Setelah seluruh data diperoleh, selanjutnya dilakukan pelabelan menggunakan *semi-supervised label propagation* dengan *k-nearest neighbours*.

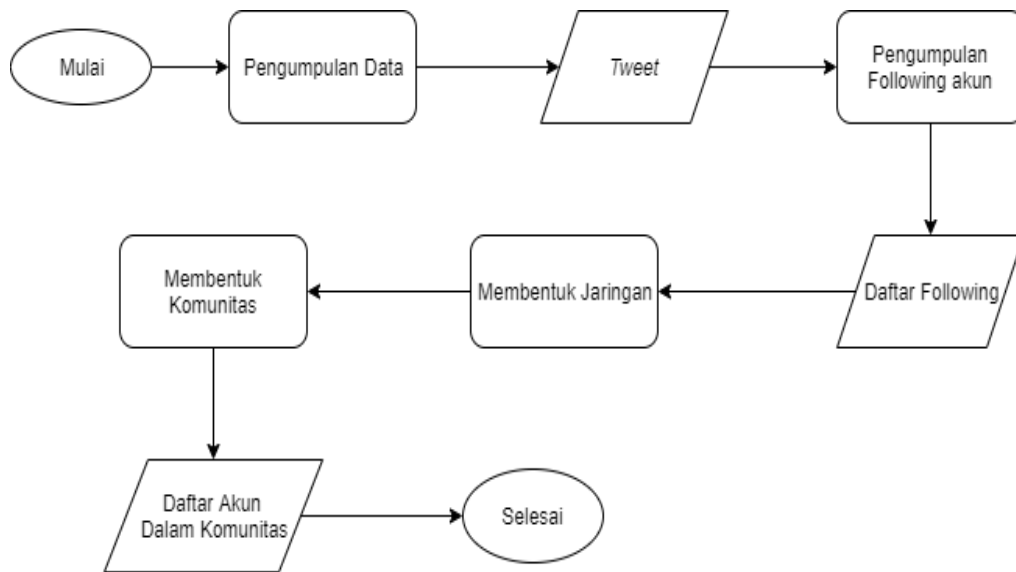
3.1.2 Deteksi Komunitas

Proses deteksi komunitas seperti yang ditunjukkan pada **Gambar 3.4**. Proses dimulai dari pengumpulan data pada *Twitter* dari tanggal 01 Agustus 2018 hingga 10 September 2019. Hasil dari proses ini yaitu mendapatkan *tweet-tweet* yang terdapat pada beranda @kpk_ri. *Tweet* ini tidak hanya berasal dari akun kpk saja, tetapi terdapat pula *tweet* dari akun yang melakukan *mention* kepada @kpk_ri. Setelah mendapatkan *tweet* dengan rentang waktu tersebut, tahap selanjutnya peneliti mengambil akun atau *username* yang saling berkaitan pada *tweet* tersebut. Dari *username* yang diperoleh tersebut dilakukan pengambilan *following-following* dari setiap akun. Contoh pengambilan *following* yaitu *username* arif23,

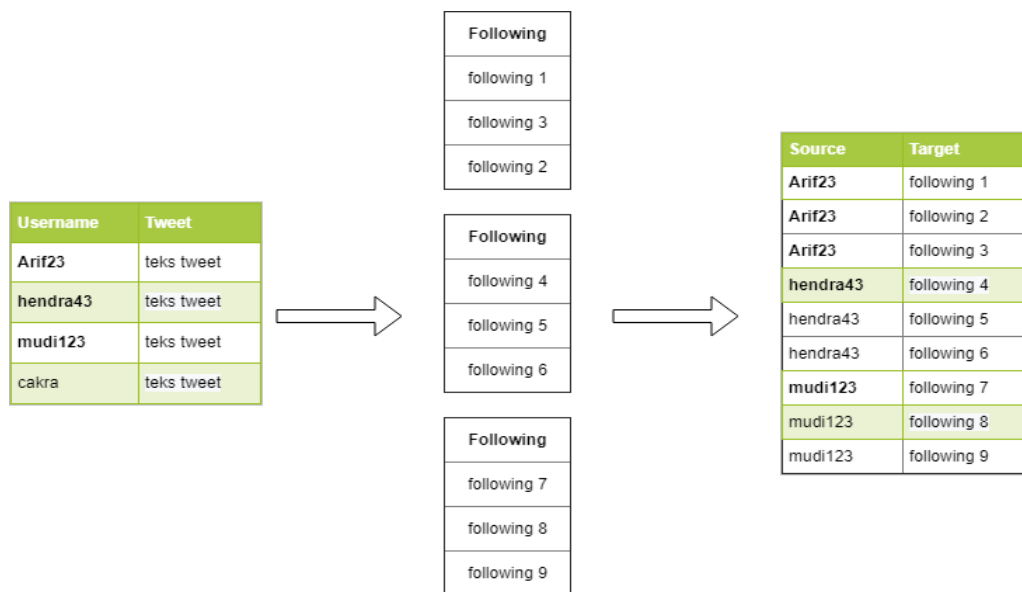
hendra43 dan mudi123. *Username* arif23 ternyata memiliki tiga *following* yaitu *following* 1, *following* 2 dan *following* 3. *Username* hendra43 memiliki tiga *following* yaitu *following* 4, *following* 5 dan *following* 6. *Username* mudi123 memiliki 3 *following* yaitu *following* 7, *following* 8 dan *following* 9. Hasil pengambilan *following* dari setiap *username* disimpan pada *csv* masing-masing. Setelah mendapatkan *following-following* tersebut, langkah selanjutnya menyatukan pada satu daftar *csv* yang berisi *source username* yang diambil *following*nya dan target yang berisi *following-following* yang didapatkan. Satu daftar *csv* ini disebut dengan *edge list*.



Gambar 3.3 Bentuk *Following* pada Akun



Gambar 3.4 Diagram Proses Alur Deteksi Komunitas



Gambar 3.5 Daftar *Following*

Proses ini merupakan bagian dari tahap pengumpulan akun yang bertujuan untuk mengetahui jaringan ataupun komunitas yang ada pada akun-akun tersebut. Setelah mendapatkan daftar *following* pada semua akun, selanjutnya melakukan pembentukan komunitas dengan menggunakan metode *louvain* dari penelitian (Blondel dkk., 2008) dengan judul “*Fast unfolding of communities in large*

networks” yang telah diaplikasikan pada perangkat lunak *gephi*. Proses pembentukan komunitas dijelaskan lebih jauh pada kajian pustaka tentang deteksi komunitas sub-bab 2.2. Tahap selanjutnya setelah mendapatkan komunitas, peneliti melakukan pengelompokan anggota-anggota yang terdapat pada setiap komunitas.

3.1.3 Ekstraksi fitur

Ekstraksi fitur merupakan tahap selanjutnya setelah mendapatkan komunitas-komunitas yang terbentuk pada proses sebelumnya (sub-subbab 3.1.2) dan telah mendapatkan data informasi akun serta *tweet*-nya. Proses ekstraksi fitur dilakukan dengan mendapatkan tiga kelompok besar yang terdiri dari fitur *tweet*, fitur komunitas, dan fitur *account*. Beberapa fitur tersebut telah disebutkan pada sub-bab 2.4 yang diperoleh dari referensi (Masood dkk., 2019), (Inuwa-Dutse dkk., 2018), (Bindu dkk., 2018), (Bhat dan Abulaish, 2013), (Sarlati dkk, 2015), dan (Fletcher dan Wennekers, 2018). Tambahan ekstraksi fitur yang dilakukan yaitu *hashtag* dan aktivitas komunitas. Fitur tambahan tersebut diantaranya jumlah interaksi *hashtag unique* pada komunitas, jumlah *hashtag* akun, jumlah *hashtag* komunitas, rasio *hashtag* komunitas, rasio *hashtag* akun, rasio *unique hashtag* akun, rasio *unique hashtag* pada komunitas, jumlah URL *unique* pada komunitas, jumlah URL pada komunitas, rasio URL pada komunitas, rasio *unique URL* komunitas, jumlah *eigen vector* komunitas, rasio *eigen vector* komunitas, rasio kata *spam* pada akun, jumlah kata *spam* pada akun, rasio *unique kata spam* pada akun, rasio kata *spam* pada komunitas, jumlah kata *spam* pada komunitas, dan rasio *unique kata spam* pada komunitas.

Jumlah *hashtag unique* (*JHUK*) pada komunitas merupakan banyaknya *hashtag* yang *unique* pada satu komunitas. Jumlah interaksi *hashtag unique* dinotasikan dengan Persamaan 3.1.

$$JHUK = \left\{ \sum_{j \in K_i} \text{unique hashtag}_j \mid K_i \in K \right\} \quad (3.1)$$

Jumlah *hashtag* akun (HA) merupakan banyaknya *hashtag* (H) yang digunakan oleh akun tersebut. Jumlah *hashtag* akun dinotasikan dengan Persamaan 3.2.

$$HA = n(H) \quad (3.2)$$

Jumlah *hashtag* komunitas (HK) merupakan banyaknya *hashtag* yang digunakan oleh seluruh anggota komunitas. Jumlah *hashtag* komunitas dinotasikan dengan Persamaan 3.3.

$$HK = \{\sum_{j \in K_i} HA_j \mid K_i \in K\} \quad (3.3)$$

Rasio *hashtag* komunitas (RHK) merupakan hasil bagi antara jumlah rasio *hashtag* akun satu komunitas dengan jumlah anggota komunitas. Rasio *hashtag* komunitas dinotasikan dengan Persamaan 3.4.

$$RHK = \frac{\sum_{i=1}^n RHA_i}{JAK} \quad (3.4)$$

Rasio *hashtag* akun (RHA) merupakan hasil bagi panjang karakter *hashtag* (kH) dengan jumlah karakter *tweet*. Rasio *hashtag* akun dinotasikan dengan Persamaan 3.5.

$$RHA = \frac{n(kH)}{JKT} \quad (3.5)$$

Rasio *unique hashtag* akun ($RUHA$) merupakan hasil bagi jumlah *unique hashtag* dengan jumlah total *hashtag* pada akun. Rasio *unique hashtag* akun dinotasikan dengan Persamaan 3.6.

$$RUHA = \frac{\text{jumlah unique hashtag}}{HA} \quad (3.6)$$

Rasio *unique hashtag* pada komunitas (UHK) merupakan hasil bagi antara jumlah rasio *unique hashtag* akun dengan jumlah anggota komunitas. Rasio *unique hashtag* pada komunitas dinotasikan dengan Persamaan 3.7.

$$UHK = \frac{\sum_{i=1}^n UHA_i}{JAK} \quad (3.7)$$

Jumlah URL *unique* pada komunitas ($JUUK$) merupakan hasil bagi antara jumlah URL *unique* pada akun ($JUUA$) dengan jumlah total anggota komunitas. Jumlah URL *unique* pada komunitas dinotasikan dengan Persamaan 3.8.

$$JUUK = \frac{\sum_{i=1}^n JUUA_i}{JAK} \quad (3.8)$$

Jumlah URL pada komunitas (JUK) merupakan banyaknya URL pada satu komunitas. Jumlah URL pada komunitas dinotasikan dengan Persamaan 3.9.

$$JUK = \{ \sum_{j \in K_i} URL(A)_j \mid K_i \in K \} \quad (3.9)$$

Rasio URL pada komunitas (RUK) merupakan jumlah total rasio pada komunitas. Rasio URL pada komunitas dinotasikan dengan Persamaan 3.10.

$$RUK = \{ \sum_{j \in K_i} RU(A)_j \mid K_i \in K \} \quad (3.10)$$

Rasio *unique* URL komunitas ($RUUK$) merupakan hasil pembagian dari jumlah *unique* URL dengan jumlah URL pada komunitas. Rasio *unique* URL dinotasikan dengan Persamaan 3.11.

$$RUUK = \frac{JUUK}{JUK} \quad (3.11)$$

Jumlah *eigenvector* komunitas (JEK) merupakan banyaknya jumlah keseluruhan *eigenvector* pada satu komunitas. Jumlah *eigenvector* komunitas dinotasikan dengan Persamaan 3.12.

$$JEK = \{ \sum_{j \in K_i} Ce_j \mid K_i \in K \} \quad (3.12)$$

Rasio *eigenvector* komunitas (REK) merupakan hasil bagi antara jumlah *eigenvector* komunitas dengan jumlah anggota pada komunitas. Rasio *eigenvector* dinotasikan dengan Persamaan 3.13.

$$REK = \frac{JEK}{JAK} \quad (3.13)$$

Rasio kata *spam* pada akun (*RSA*) merupakan hasil bagi jumlah karakter *spam* (*ks*) dan jumlah panjang karakter *tweet*. Rasio kata *spam* pada akun dinotasikan dengan Persamaan 3.14.

$$RSA = \frac{n(ks)}{JKT} \quad (3.14)$$

Jumlah kata *spam* pada akun (*JKA*) merupakan banyaknya kata *spam* yang terdapat pada suatu akun. Jumlah kata *spam* pada akun dinotasikan dengan Persamaan 3.15.

$$JKA = n(kata\ spam) \quad (3.15)$$

Rasio *unique* kata *spam* pada akun (*UKA*) merupakan hasil bagi antara *unique* kata *spam* dengan jumlah kata *spam* pada akun. Rasio *unique* kata *spam* pada akun dinotasikan dengan Persamaan 3.16.

$$UKA = \frac{n(unique\ kata\ spam)}{JKA} \quad (3.16)$$

Rasio kata *spam* pada komunitas (*RKK*) merupakan banyaknya jumlah rasio kata *spam* akun yang terdapat dalam satu komunitas. Rasio kata *spam* pada komunitas dinotasikan dengan Persamaan 3.17.

$$RKK = \{\sum_{j \in K_i} RSA(A)_j \mid K_i \in K\} \quad (3.17)$$

Jumlah kata *spam* pada komunitas (*JKK*) merupakan banyaknya kata-kata *spam* yang terdapat pada satu komunitas. Jumlah kata *spam* pada komunitas dinotasikan dengan Persamaan 3.18.

$$JKK = \{\sum_{j \in K_i} JKA(A)_j \mid K_i \in K\} \quad (3.18)$$

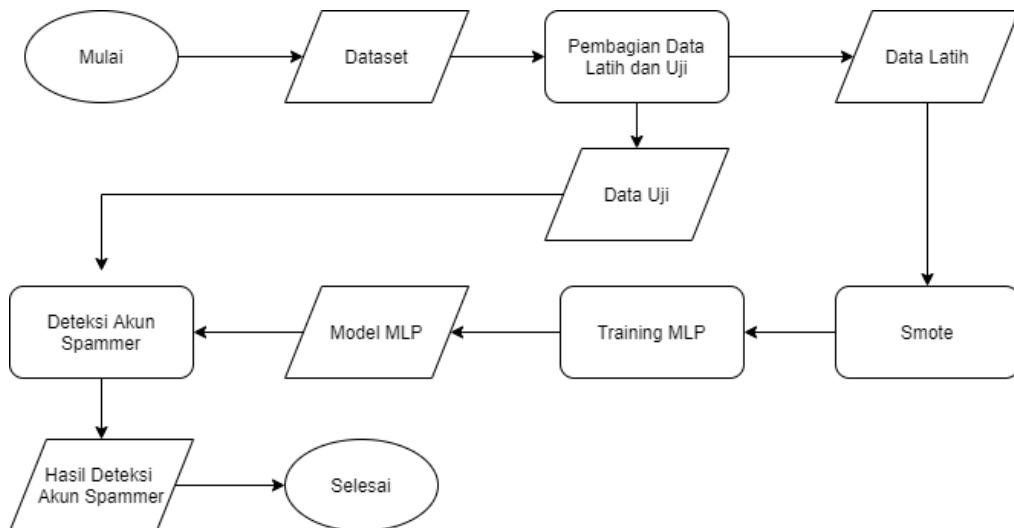
Rasio *unique* kata *spam* pada komunitas (*UUK*) merupakan banyaknya rasio *unique* kata *spam* yang terdapat pada satu komunitas. Rasio kata *spam* pada komunitas dinotasikan dengan Persamaan 3.19.

$$UUK = \{\sum_{j \in K_i} UKA(A)_j \mid K_i \in K\} \quad (3.19)$$

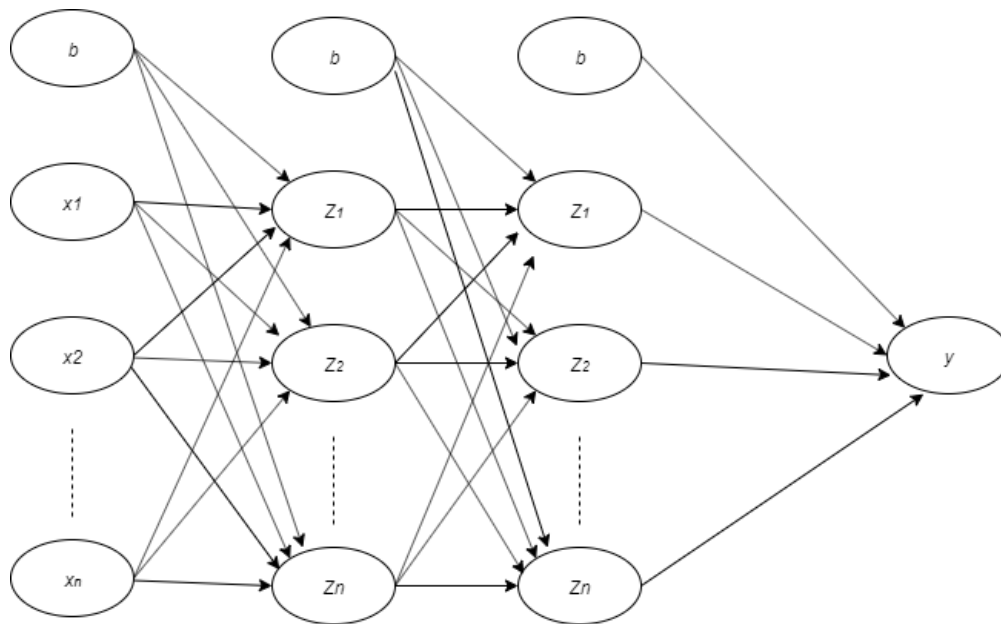
3.1.4 Seleksi Fitur

Proses seleksi fitur dilakukan setelah proses ekstraksi fitur selesai. Hasil ekstraksi fitur yang terbagi menjadi tiga kelompok yaitu jaringan (*graph*), akun, dan komunitas tersebut dilakukan penyatuan. Selanjutnya, fitur dilakukan proses seleksi fitur untuk mendapatkan fitur-fitur yang optimal. Proses seleksi fitur menggunakan *support vector machine-recursive feature elimination* yang diadopsi dari penelitian (Park dkk, 2018) dan (Guyon, Weston, Barnhill, dan Vapnik, 2002). *Support vector machine-recursive feature elimination* melakukan seleksi fitur dengan cara *backward*.

Proses *support vector machine-recursive feature elimination* dimulai dengan melakukan proses pelatihan *support vector machine*, sehingga diperoleh bobot hasil dari pelatihan. Kemudian, hasil pelatihan dilakukan perhitungan hasil bobot pelatihan terhadap panjangnya dimensi *dataset*. Dari hasil proses perhitungan bobot tersebut dilakukan proses pencarian kriteria terkecil, yang kemudian akan digunakan untuk proses peringkatan fitur. Jika peringkatan fitur telah dilakukan, maka proses dilanjutkan dengan melakukan *update* urutan peringkat dari fitur-fitur yang ada. Setelah proses *update* peringkat, proses



Gambar 3.6 Alur Deteksi *account Spammer*



Gambar 3.7 Arsitektur *Multi-Layer Perceptron*

dilanjutkan dengan menghapus fitur yang memiliki kriteria terkecil sehingga diperoleh fitur-fitur terbaik yang sesuai dengan peringkatnya. Hasil dari proses seleksi fitur ini selanjutnya akan digunakan pada proses deteksi akun *spammer*.

3.1.5 Deteksi Akun *Spammer*

Proses deteksi *spammer* dilakukan sesuai dengan Gambar 3.6 dengan menggunakan *multi-layer perceptron* (MLP) setelah mendapatkan fitur-fitur terbaik dari proses seleksi fitur dengan menggunakan RFE. *Dataset* dengan fitur hasil RFE dilakukan proses pembagian data menjadi data latih dan data uji. Kemudian dilakukan proses *Smote* pada data latih untuk menyeimbangkan data *legitimate* dan *spammer*. *Smote* menggunakan penelitian (Chawla dkk., 2002). Proses *smote* dilakukan sesuai *pseudo-code* pada sub-bab 2.5.1. Hasil *Smote* digunakan sebagai *input* pada proses *multi-layer perceptron*.

Multi-layer perceptron yang digunakan diadopsi dari penelitian (Jain, Sharma, dan Agarwal, 2019). Proses *multi-layer perceptron* memiliki tiga tahapan besar yaitu proses *forward*, proses *backward*, dan juga proses perubahan bobot.

Seperti Gambar 3.7, *multi-layer perceptron* menggunakan jumlah inputan yang sesuai dengan jumlah fitur setelah dilakukan proses seleksi fitur, jumlah *hidden layer* yaitu 2 *hidden layer* dan beberapa percobaan *node hidden* yaitu (15,15), (20,20) dan (25,25). Maksimum *epoch* yang digunakan yaitu 1000. Tingkat kesalahan terendah yaitu 0.0001. Proses *multi-layer perceptron* seperti yang dijelaskan pada sub-bab 2.7, dimana fitur-fitur yang didapatkan pada proses seleksi fitur dengan menggunakan *support vector machine–recursive feature selection* dilakukan proses *forward* (propagasi maju) ke *hidden layer* hingga ke *output layer* dengan menggunakan fungsi aktivasi *sigmoid biner*. Setelah proses *forward* telah mencapai *layer output*, maka proses selanjutnya yaitu *backward* (propagasi mundur). *Backward* dilakukan untuk menghitung nilai kesalahan yang diperoleh dari selisih *output layer* dan target yang ditentukan. Proses *backward* dilakukan untuk semua *layer*, dimulai dengan mencari kesalahan/*error* pada *layer*. Setelah mendapatkan nilai kesalahan pada semua *layer*, selanjutnya dilakukan proses perubahan bobot yang dihitung berdasarkan nilai kesalahan pada setiap *layer*-nya. Proses ini dilakukan terus menerus hingga titik nilai pemberhentian yang ditentukan, baik nilai minimal kesalahan atau maksimum iterasi. Setelah proses pelatihan selesai, maka akan didapatkan model dari *multi-layer perceptron* tersebut. Model ini selanjutnya akan digunakan untuk pengujian.

Data pengujian merupakan data hasil pembagian dari data utama yang dibagi menjadi dua bagian, yaitu data pelatihan dan data pengujian. Proses pengujian pada *multi-layer perceptron* dilakukan pada fase propagasi maju (*forward*), yang keluaran/hasilnya akan diperoleh setelah fase tersebut dilakukan.

Nilai hasil pengujian dilakukan penggabungan antara MLP *account*, MLP *tweet*, dan MLP komunitas. Proses penggabungan dilakukan dengan menambahkan hasil perkalian dari *output perceptron multi-layer* dengan masing-masing bobot. Perkalian tersebut adalah seperti berat fitur *account* (α) * hasil MLP fitur *account* (A), bobot fitur *tweet* (β) * hasil MLP fitur *tweet* (B), dan bobot fitur komunitas (γ) * hasil MLP fitur komunitas (C). Total bobot α , β , dan γ adalah

satu. Hasil proses penggabungan dilakukan klasifikasi dengan menggunakan *threshold* untuk mendapatkan hasil klasifikasi. Hasil proses penggabungan dinyatakan sebagai *spammer* jika skor lebih kecil dari *threshold* dan bukan *spammer* jika skor lebih besar dari *threshold*. Proses penggabungan dilakukan dengan Persamaan 3.10 dan Persamaan 3.21.

$$\text{score} = \delta * (AB) + (1 - \gamma) * C \quad (3.20)$$

$$\text{Tweet account score } (AB) = \alpha * A + (\delta - \beta) * B \quad (3.21)$$

3. 2. Implementasi

Secara keseluruhan, deteksi akun *spammer* berdasarkan *hashtag* dan aktivitas komunitas pada *Twitter* dibangun dengan menggunakan bahasa pemrograman *python* versi 3.6. Proses pengembangan dengan menggunakan *environment jupyter notebook*. *Library* yang digunakan diantaranya yaitu *pandas*, *GetOldTweet3*, *twint*, *scikit-learn*, *numpy* dan *library* lainnya. *Hardware* yang akan digunakan yaitu *PC processor AMD ryzen 5*, *RAM 32 GB DDR4* dan sistem operasi *ubuntu 18.04*.

3. 3. Pengujian dan Evaluasi

Proses pengujian dilakukan dengan membagi antara data latih dan data uji. Jumlah persentase pembagian data latih dan data uji yaitu 70:30, 80:20 dan 90:10. Fitur yang akan diuji dibagi menjadi 3 kategori yaitu fitur akun, fitur komunitas, dan fitur *tweet*. Proses pengujian *multi-layer perceptron* dengan melakukan perubahan pada jumlah *node hidden layer* serta menggunakan hasil seleksi fitur yang telah diperoleh. Sehingga akan menampilkan analisa akurasi, *precision*, *recall*, dan *g-mean*. Parameter yang dilakukan perubahan diantaranya yaitu *learning rate* dengan nilai 0.1, 0.01, dan 0.001. Jumlah *node hidden layer* yaitu 15, 20, dan 25. Selain itu akan dilakukan pengujian menggunakan *support vector machine*. Kernel yang akan dilakukan perubahan yaitu *linear*. Pada proses pengujian ini akan dilakukan pengukuran performa hasil deteksi akun *spammer*

Tabel 3.1 Tabel *Confusion Matrix* pengujian

		Kelas Asli	
		Akun <i>Spammer</i>	Akun Sah
Kelas Prediksi	Akun <i>Spammer</i>	?	?
	Akun Sah	?	?

menggunakan *confusion matrix* pada Tabel 3.1 dengan melakukan pengukuran akurasi, *recall*, presisi dan *g-mean*. Pengukuran akurasi dilakukan dengan Persamaan 2.40, dimana akan dihitung jumlah keseluruhan kebenaran baik *true positive* (akun *spam* yang dinyatakan *spam*) dan *true negative* (akun bukan *spam* dinyatakan akun bukan *spam*), dibagi dengan keseluruhan data yang diuji. Pengukuran *recall* menggunakan Persamaan 2.41 yang dihitung dengan membagi *true positive* dengan jumlah dari *true positive* ditambah *false negative*. *Precision* merupakan ketepatan dalam melakukan deteksi. *Precision* dihitung dengan menggunakan Persamaan 2.42. Pengukuran *precision* dilakukan dengan membagi *true positive* dengan jumlah *true positive* dan *false negative*. *G-mean* merupakan pengukuran yang mempertimbangan keseimbangan antara kelas positif dan kelas negatif. *G-mean* dihitung dengan menggunakan Persamaan 2.43.

Penelitian ini akan dibandingkan dengan Penelitian (Perdana, Muliawati, dan Alexandro, 2015) dengan Judul “*Bot Spammer Detection in Twitter Using Tweet Similarity and Time Interval Entropy*”.

3. 4. Dokumentasi

Tahapan terakhir dari penelitian ini yaitu melakukan penulisan dan penyusunan laporan. Tahapan ini memiliki tujuan untuk mendokumentasikan semua proses dan hasil penelitian yang telah dilakukan.

[Halaman ini sengaja dikosongkan]

BAB 4

HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan hasil dan pembahasan mengenai implementasi dan pengujian yang telah dilakukan. Hasil dan pembahasan tersebut adalah data, deteksi komunitas, ekstraksi fitur, seleksi fitur, dan deteksi akun *spammer*.

4.1. Data

Data digunakan pada penelitian ini yaitu data dari *Twitter*. Data diambil tidak menggunakan API resmi dari *Twitter*, tetapi proses pengambilan menggunakan pustaka python *GetOldTweet3*. Proses pengambilan tidak menggunakan API resmi *Twitter* dikarenakan API *Twitter* memberikan izin hanya untuk pengambilan data selama 7 hari terakhir. *Tweet* diambil dari halaman akun *Twitter* Komisi Pemberantasan Korupsi (KPK). Histori waktu *tweet* dilakukan proses pengambilan yaitu tanggal 01 Agustus 2018 hingga 10 September 2019. Jumlah *tweet* yang diperoleh yaitu 22.281 *tweet*. Tabel 4.1 merupakan contoh

Tabel 4.1 Contoh *Tweet* dari halaman @kpk_ri

No	Username	<i>Tweet</i>
1	PratamaTraveL2	Ga kapok2 !! Coba kalo undang2 korupsi diterapkan hukum Islam bakalan jera tuh mereka yg suka maling uang Rakyat , di negara2 kafir saja yg korupsi ada yg menerapkan HUKUMAN MATI ,kok dinegara kita ga berani ya Uda Rizal Ramli?
2	netytarigan	Ini adalah contoh pilih wakil rakyat (DPR) bersih! Agar negara kita bebas dari korupsi
3	NandhaBung	Bang Gimana cara tahapan nya memeriksa Rektor yg terindikasi Korupsi ? tolong Di jawab ya Bang
4	bajigurceleng	Inilah partai dajjal juaranya KORUPSI @KPK_RI , katanya mau umumin tersangka baru korupsi EKTP
5	AJudakusumah	Semoga @KPK_RI mkn intens selamatkan dana utk rakyat, tdk sekedar memberantas tp juga mencegah semua btk korupsi. Biadab, dana kesehatan milik rakyat kecil msh dijarah juga. Cc @jokowi @DivHumas_Polri @LaodeMSyarif @StPiobang @DBSelamun @bdm2502 @karniilyas @SMIndrayanto @ILCtv1

Tabel 4.2 Contoh informasi *account username*

No	Name	Username	Join_date	Tweets	Follo wing	Follo wers	Likes	Verified
1	Arfai 71	iriyant89135653	5 Oct 2018	765	463	16	954	0
2	Tifatul Sembiring	tifsembiring	20 Oct 2009	37813	2094	1672416	378	1
3	Joni_AS	Jonias1717	12 Apr 2019	1069	525	77	1328	0

	date	time	tweet	mentions	urls	hashtags	replies_count	retweets_count	likes_count
0	2019-09-16	14:55:08	Permendag 29/2019 ini yg bertentangan dg UU. F...	['ivanarifa']			1	0	0
1	2019-09-16	14:53:59	Si mbak bisa baca link ini: https://m.republi...	['ivanarifa']	['https://m.republika.co.id/berita/pxu06o349/a...']		2	1	2
2	2019-09-16	09:39:14	Justu permen yg baru ini bertentangan dg UU. ...	['ivanarifa']			2	0	1
3	2019-09-15	20:23:40	Makanya situ mikir, lu pilih yg lain, maka pil...	['cangkir_plastik', 'ari1686', 'msaid_didu']			3	0	3
4	2019-09-15	17:28:38	Astaghfirullah... Derita satwa, karena ulah ke...				53	460	1242

Gambar 4.1 Contoh *tweet* dari *username* @tifsembiring

tweet yang diperoleh. Seperti contoh pada Tabel 4.1, dilakukan pengambilan *username*. Total *username* yang diperoleh yaitu 22.281 *username*. Total *username* tersebut kemudian diambil *username unique* sehingga diperoleh 10.961 *username*. *Username* sebanyak 10.961 dilakukan proses pengambilan followingnya. Proses pengambilan *following* menggunakan *library python twint*. Total *username* yang berhasil dilakukan pengambilan yaitu 10.211 dari 10.961 *username*. Total *username* yang gagal diambil followingnya yaitu 861 *username*, hal ini dikarenakan *username* tersebut tidak diizinkan diambil (*private*) datanya. Total *username* yang diperoleh dari *following* yaitu 4.995.357 *username*. Total *username* tersebut tidak *unique*, karena setiap *following username* diletakkan pada csv masing-masing *username* yang diambil. Total *username unique* hasil *following* yaitu 1.392.841 *username*. Pada *username* ini dilakukan 3 proses yaitu proses pembentukan *edge list* untuk deteksi komunitas, proses pengambilan

informasi *account username*, dan proses pengambilan *tweet* pada *username*. Tabel 4.2 merupakan contoh hasil pengambilan data *account username*. Gambar 4.1 merupakan contoh pengambilan *tweet* pada *username @tifsembiring*.

4.2. Deteksi Komunitas

Username hasil pengambilan dengan jumlah 10.211 dilakukan pembentukan *edge list*. *Edge list* tersebut dibuat berdasarkan *username (source)* dan *following* pada *username (target)*. Tabel 4.3 merupakan contoh dari hasil *edge list*. *Edge list* digunakan sebagai inputan pada aplikasi *gephi*. Tabel 4.4 merupakan Contoh hasil

Tabel 4.3 Contoh *source* dan *target* pada *edge list*

Source	Target
pratamatravel2	pak_jk
pratamatravel2	cakkhum
pratamatravel2	mohammadimronu1
netytarigan	dianisadiawati
netytarigan	bps_statistics
netytarigan	infoalid
nandhabung	dahnilanzar
nandhabung	sandiuno
nandhabung	aniesbaswedan
bajigurceleng	fatinpane
bajigurceleng	dr_koko28
bajigurceleng	kangsyam78

Tabel 4.4 Contoh hasil deteksi komunitas

No	Id	modularity_class	indegree	eigencentality
1	teh_tariq	0	163	0,027427
2	diniisme	1	4	0,000673
3	ceritaayahku	2	7	0,001178
4	lgshop	3	13	0,002187
5	kamardewasa	4	16	0,002692

deteksi komunitas dengan aplikasi *gephi*. Total komunitas yang diperoleh yaitu 76 komunitas. Tabel 4.4 menunjukkan komunitas pada kolom *modularity_class*. Jumlah komunitas terbesar berjumlah 259.148 *username*. Jumlah *indegree* terbesar yaitu 5943. *Indegree* terbesar terdapat pada *account @kpk_ri*, hal ini dikarenakan pengambilan *tweet* tentang korupsi.

4.3. Ekstraksi Fitur

Ekstraksi fitur dilakukan dengan menggunakan data *tweet*, data informasi *account*, dan data komunitas. Tabel 4.5, Tabel 3.6, dan Tabel 4.7 merupakan hasil ekstraksi fitur *account*, *tweet*, dan komunitas. Proses ekstraksi dilakukan sesuai sub bab 2.3 dan sub bab 3.1.3. Proses mendapatkan *indegree* dan *eigenvector centrality* seperti diperlihatkan pada Tabel 4.4 kolom *indegree* dan *eigenvector centrality*. *Indegree* yang diperoleh sebanyak 163 dan *eigenvector* sebanyak 0,027427 untuk id *teh_tarik*. Jumlah *indegree* diperoleh dari penjumlahan *indegree* pada satu komunitas. Contoh proses perhitungan jumlah *indegree* untuk id *teh_tarik* yaitu $163 + 2310 + \dots + 3850 = 696910$ *degree*.

Jumlah anggota komunitas diperoleh dari banyaknya anggota pada satu komunitas. Contoh jumlah anggota komunitas untuk id *teh_tarik* yaitu 165518. Umur akun diperoleh dengan menghitung lama akun dari waktu pembuatan (Tabel 4.2 kolom *Join_date*) hingga waktu pengambilan data yaitu 01 Agustus 2019. Contoh umur akun dimiliki oleh *username iriyant89135653* yaitu 444 hari. Jumlah *follower*, *following*, dan jumlah status diperoleh dari Tabel 4.2. Contoh jumlah *follower*, *following*, dan jumlah status *username iriyant89135653* yaitu 16 *follower*, 463 *following*, dan 765 *tweet*.

Jumlah digit nama diperoleh dari banyaknya angka pada *screen name*. Contoh jumlah digit nama pada “Arfai 71” yaitu 2 digit (7 dan 1). Panjang *username* dihitung dari panjang karakter pada *username*. Contoh panjang *username* “iriyant89135653” yaitu 15 karakter. Panjang *screen name* dihitung dari panjang karakter pada nama. Contoh panjang *screen name* “Arfai 71” yaitu 8 karakter. *Similarity username* dan *screen name* diperoleh dari kemiripan

Tabel 4.5 Hasil Ekstraksi informasi *account*

No	<i>username</i>	Umur <i>account</i>	Follower	Follower	Jumlah Status	Digit Nama	Panjang <i>Username</i>	Panjang <i>Screen</i> Name	<i>Similarity</i> <i>Username</i> dan <i>Screen</i> Name	Rasio Follower	Rasio Follower	<i>Interest</i> <i>ingness</i>	Keaktifan Akun	<i>Rasio</i> Nama
1	iriyant891 35653	444	16	463	765	2	15	8	0,436436	28,93 75	0,034 557	1,247	0.172297	0,533333
2	tifsembiri ng	4386	1672 416	2094	37813	0	12	17	0,914138	0,001 252	798,6 7	0,0099	0,86213	1,416667
3	jonias171 7	240	77	525	1069	0	10	7	0,606092	6,818 182	0,146 667	1,2423	4,454167	0,7
4	najwashih ab	4824	3710 167	500	25162	0	11	12	1,0	0,000 135	7420, 334	0,0002 78	0,5216	1,090909
5	priyobudis	3954	1987 42	1115	30860	0	10	18	0,849208	0,005 61	178,2 43946	0,3610 5	0,780475	1,8
6	pak_jk	3072	2949 321	732	4085	0	6	11	0,441942	0,000 248	4029, 127	0,0019 58	0,132975	1,833333

antara *username* dan *screen name*. Contoh hasil *similarity username* dan *screen name* dari “tifsembiring” dan “Tifatul Sembiring” yaitu 0,914138.

Rasio *following* diperoleh dari pembagian *following* dan *follower*. Contoh rasio *following* tifsembiring yaitu $2094/1672416 = 0,001252$. *Interestingness* diperoleh dari pembagian antara jumlah *like* dengan jumlah *tweet*. Contoh *interestingness* pada *username* jonias1717 yaitu $1328/1069 = 1,242283$. Tingkat keaktifan *account* diperoleh dari pembagian jumlah status dengan umur akun. Contoh tingkat keaktifan *account* pada *username* iriyant89135653 yaitu $765/4440 = 0,172297$. Rasio nama diperoleh dari pembagian antara panjang *screen name* dan *username*. Contoh rasio nama pada *username* jonias1717 yaitu $7/10 = 0,7$.

Rata-rata panjang *tweet* diperoleh dari jumlah panjang *tweet* dengan jumlah *tweet* yang dilakukan. Contoh rata-rata panjang *tweet* pada *username* jonias1717 yaitu $50844/494 = 103$. Rasio URL diperoleh dari pembagian jumlah karakter URL dengan panjang *tweet* dan diambil rata-ratanya. Contoh rasio URL pada *username* jonias1717 yaitu $189,213 /351 = 0,539$. Rasio *mention* diperoleh dari pembagian jumlah karakter *mention* dengan panjang *tweet* dan diambil rata-ratanya. Contoh rasio *mention* pada *username* jonias1717 yaitu $33,16176/127 = 0,2611$. *Lexrichwithuu* merupakan tingkat kekayaan *tweet* pada *account*. Contoh *lexrichwithuu* pada *username* jonias1717 yaitu 0,975641. *Lexrichwithuu* semakin mendekati 1 berarti tingkat kekayaan token semakin tinggi pada *tweet*. *Lexrichwithoutuu* merupakan tingkat kekayaan token pada *tweet* dengan menghilangkan *mention* dan URL. Contoh *lexrichwithoutuu* pada jonias1717 yaitu 0,643978.

Rasio *unique* URL diperoleh dari jumlah URL *unique* dibagi dengan jumlah URL pada *account*. Contoh rasio *unique* URL pada *username* jonias1717 yaitu $493/494 = 0,99797$. Jumlah *hashtag* adalah banyaknya *hashtag* pada akun. Contoh jumlah *hashtag* pada *username* jonias1717 yaitu 309. Rasio *hashtag* diperoleh dari pembagian panjang karakter *hashtag* dengan panjang karakter *tweet*. Rasio *hashtag account* diambil dari jumlah rasio *hashtag* dibagi

Tabel 4.6 Hasil Ekstraksi informasi *tweet*

No	<i>username</i>	Rata panjang	Rasio URL	Rasio <i>mention</i>	Lexri chwit huu	Lexrich without uu	Rasio <i>unique</i> URL	Jumlah <i>hashtag</i>	Rasio <i>hashtag</i>	Rasio <i>unique</i> <i>hashtag</i>	Rasio kata spam	Jumlah kata spam	Rasio <i>unique</i> kata spam
1	iriyant89135653	107	0,488014	0,204525	0,940633	0,575686	1,0	13	0,323876	0,846154	0,018612	104	0,211538
2	tifsembiring	113	0,427696	0,288653	0,975511	0,712141	1,0	2	0,088502	1,0	0,015976	70	0,342857
3	jonias1717	103	0,539071	0,261116	0,975641	0,643978	1,0	309	0,195569	0,174757	0,055704	230	0,217391
4	najwashihab	260	0,327841	0,056243	0,96281	0,614872	1,0	765	0,112529	0,240523	0,049285	537	0,249534
5	priyobudis	107	0,586225	0,169411	0,978818	0,622548	0,966667	4	0,134321	1,0	0,044501	22	0,545455
6	pak_jk	108	0,40005	0,132428	0,984872	0,759707	1,0	97	0,13374	0,525773	0,062635	435	0,114943

banyak *tweet* yang terdapat *hashtag*. Contoh rasio *hashtag account* pada *username* pak_jk yaitu $12,304 / 92 = 0,1337$. Rasio *unique hashtag* merupakan hasil bagi dari jumlah *hashtag unique* dengan jumlah *hashtag* pada *account*. Contoh rasio *unique hashtag* pada *username* priyobudis yaitu $4/4 = 1$.

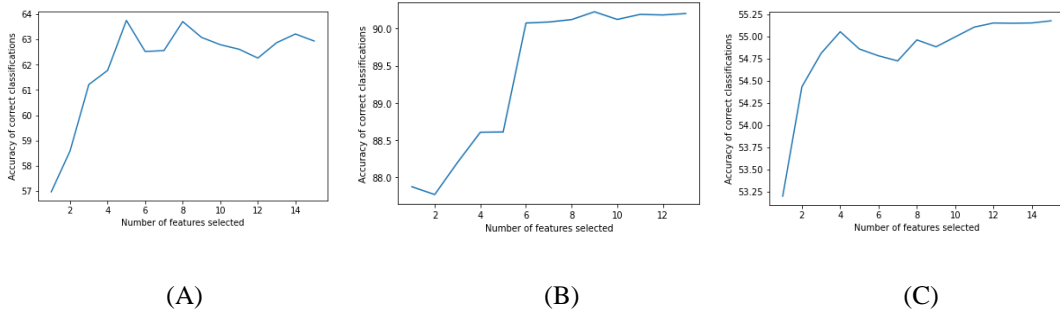
Jumlah anggota komunitas merupakan banyaknya anggota komunitas pada suatu komunitas. Contoh anggota komunitas pada *username* yang terdapat pada komunitas 14 yaitu 259180. Jumlah *unique hashtag* merupakan banyaknya *hashtag unique* pada suatu komunitas. Contoh jumlah *unique hashtag* pada *username* yang terdapat pada komunitas 14 yaitu 606479. Jumlah *hashtag* komunitas merupakan banyaknya *hashtag* pada suatu komunitas. Contoh jumlah *hashtag* pada *username* yang terdapat pada komunitas 14 yaitu 25099798.

Rasio *hashtag* komunitas merupakan jumlah rasio *hashtag* suatu komunitas dibagi dengan banyaknya anggota komunitas. Contoh rasio *hashtag* komunitas pada *username* yang terdapat pada komunitas 14 yaitu 0,155154. Rasio *unique hashtag* merupakan jumlah *unique hashtag* pada satu komunitas dibagi jumlah anggota komunitas. Contoh rasio *unique hashtag* komunitas pada *username* yang terdapat pada komunitas 14 yaitu 0,528336. Jumlah URL komunitas merupakan banyaknya URL pada suatu komunitas. Contoh jumlah URL pada *username* yang terdapat pada komunitas 14 yaitu 30504220.

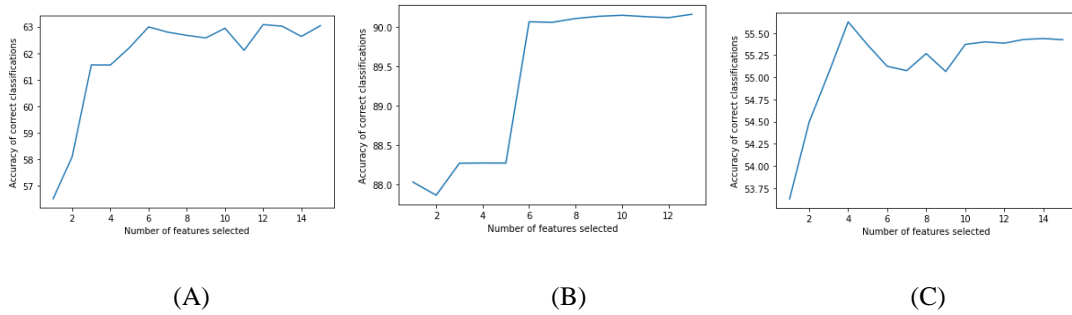
Rasio *unique URL* komunitas merupakan jumlah rasio *unique URL* pada suatu komunitas dibagi jumlah anggota komunitas. Contoh rasio *unique URL* komunitas pada *username* yang terdapat pada komunitas 14 yaitu 0,842187. Rasio URL komunitas merupakan jumlah rasio URL pada suatu komunitas dibagi jumlah anggota komunitas. Contoh rasio URL komunitas pada *username* yang terdapat pada komunitas 14 yaitu 0,389157. Jumlah *eigenvector* komunitas merupakan banyaknya *eigenvector* pada suatu komunitas. Contoh jumlah *eigenvector* komunitas pada *username* yang terdapat pada komunitas 14 yaitu 239,888171.

Tabel 4.7 Hasil Ekstraksi informasi komunitas

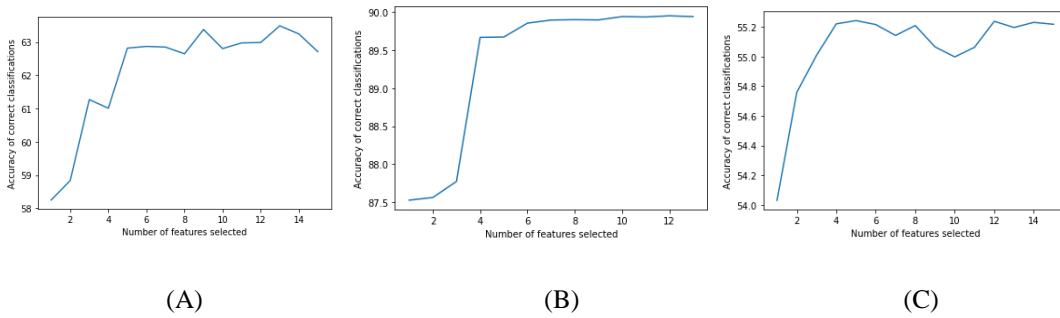
No	<i>username</i>	Jumlah <i>indegree</i>	Jumlah anggota	Jumlah <i>hashtag</i> <i>unique</i>	Jumlah <i>hashtag</i>	Rasio <i>hasht</i> <i>ag</i>	Rasio <i>unique</i> <i>hashtag</i>	Jumlah URL	Rasio URL	Rasio <i>unique</i> URL	Jumlah <i>eigenve</i> <i>ctor</i>	Rasio <i>eigenve</i> <i>ctor</i>	Rasio kata spam	Rasio <i>unique</i> kata spam
1	teh_tariq	696910	165518	423146 4	125476 05	0,220 172	0,55675 5	127062 03	0,45781 6	0,82959 5	89,548 028	0,00091 9	0,080905	0,313984
2	diniisme	19406	15334	423146 4	917579	0,162 465	0,59133 4	132110 6	0,39375 8	0,83035	2,0014 64	0,00023 1	0,101634	0,318838
3	ceritaayahku	2751	2105	698034 7	187185	0,179 702	0,53676 2	214360	0,39150 6	0,81219 1	0,3089 25	0,00023 7	0,073808	0,318564
4	Ignshop	5975	4304	243303	545657	0,147 469	0,52764 6	658848	0,35229 7	0,80945 5	0,7401 03	0,00024 9	0,123361	0,296479
5	kamardewasa	15912	11731	464838	575291	0,168 393	0,58551 3	908727	0,40140 3	0,80988 7	1,5838 62	0,00025 5	0,066118	0,3572
6	pak_jk	1695463	259180	606479	250997 98	0,155 154	0,52833 6	305042 20	0,38915 7	0,84218 7	239,88 8171	0,00153 1	0,089475	0,307602



Gambar 4.2 Hasil *feature selection* dataset 70:30 menggunakan *recursive feature elimination*: (a) *feature accounts*, (b) *feature tweets*, dan (c) *feature communities*.



Gambar 4.3 Hasil *feature selection* dataset 80:20 menggunakan *recursive feature elimination*: (a) *feature accounts*, (b) *feature tweets*, dan (c) *feature communities*.



Gambar 4.4 Hasil *feature selection* dataset 90:10 menggunakan *recursive feature elimination*: (a) *feature accounts*, (b) *feature tweets*, dan (c) *feature communities*.

Tabel 4.8 Fitur Optimal dataset 70:30

Nama fitur	Fitur turunan
Account	Umur akun
	Panjang <i>screen name</i>
	<i>Similarity screen name dan username</i>
	Rasio <i>following</i>
	Keaktifan <i>account</i>
Tweet	Rata-rata panjang <i>tweet</i>
	Rasio <i>unique URL</i>
	Jumlah <i>hashtag</i> pada <i>account</i>
	Rasio <i>hashtag</i> pada <i>account</i>
	Rasio <i>unique hashtag</i> pada <i>account</i>
	Jumlah kata <i>spam</i>
Komunitas	Jumlah <i>indegree</i>
	Rasio <i>unique hashtag</i> komunitas
	Rasio <i>URL</i> komunitas
	Rasio <i>unique URL</i> komunitas

4.4. Seleksi Fitur

Gambar 4.2, Gambar 4.3, dan Gambar 4.4 merupakan grafik dari hasil seleksi fitur dengan menggunakan *recursive feature elimination*. Seleksi fitur dibagi menjadi 3 sesuai dengan pembagian persentase *dataset*. Persentase pembagian *dataset* tersebut yaitu 70:30, 80:20, dan 90:10.

Tabel 4.8 merupakan fitur terbaik dengan persentase dataset 70:30. Fitur tersebut diantaranya fitur *account* sesuai Gambar 4.2 (a) yaitu 0,63746. Jumlah fitur terbaik tersebut 5 fitur yaitu umur akun, panjang *screen name*, *similarity screen name dan username*, rasio *following*, dan keaktifan *account*. Hasil seleksi fitur pada fitur *tweet* sesuai Gambar 4.2 (b) yaitu 0,9007. Jumlah fitur optimal diperoleh 6 fitur yaitu rata-rata panjang *tweet*, rasio *unique URL*, jumlah *hashtag* pada *account*, rasio *hashtag* pada *account*, rasio *unique hashtag* pada *account*, dan jumlah kata *spam*. Hasil seleksi fitur pada fitur komunitas sesuai Gambar 4.2 (c) yaitu 0,5505. Jumlah fitur optimal diperoleh 4 fitur yaitu jumlah *indegree*, rasio

unique hashtag komunitas, rasio URL komunitas, dan rasio *unique URL* komunitas.

Hasil seleksi fitur dengan persentase 80:20 pada fitur *account* sesuai Gambar 4.3 (a) yaitu 0,6299. Jumlah fitur optimal diperoleh 6 fitur yaitu *eigenvector*, umur *account*, panjang *screen name*, *similarity screen name* dan *username*, rasio *following*, dan keaktifan *account*. Hasil seleksi fitur dengan persentase 80:20 pada fitur *tweet* sesuai Gambar 4.3 (b) yaitu 0,9007. Jumlah fitur optimal diperoleh 6 fitur yaitu rata-rata panjang *tweet*, rasio *unique URL*, jumlah *hashtag* pada *account*, rasio *hashtag* pada *account*, rasio *unique hashtag* pada *account*, dan jumlah kata *spam*. Hasil seleksi fitur persentase 90:10 pada fitur komunitas sesuai Gambar 4.3 (c) yaitu 0,5563. Jumlah fitur optimal diperoleh 4 fitur yaitu jumlah anggota komunitas, rasio URL komunitas, rasio *unique URL* komunitas, dan jumlah *eigenvector* komunitas.

Hasil seleksi fitur dengan persentase 90:10 pada fitur *account* sesuai Gambar 4.4 (a) yaitu 0,6338. Jumlah fitur optimal diperoleh 9 fitur yaitu *eigenvector*, umur *account*, panjang *screen name*, *similarity screen name* dan *username*, rasio *following*, *interestingness*, keaktifan *account*, rasio *screen name*, dan *indegree*. Hasil seleksi fitur dengan persentase 90:10 pada fitur *tweet* sesuai Gambar 4.4 (b) yaitu 0,8986. Jumlah fitur optimal diperoleh 6 fitur yaitu rata-rata panjang *tweet*, rasio *unique URL*, jumlah *hashtag* pada *account*, rasio *hashtag* pada *account*, rasio *unique hashtag* pada *account*, dan jumlah kata *spam*. Hasil seleksi fitur dengan persentase 90:10 pada fitur komunitas sesuai Gambar 4.4 (c) yaitu 0,5522. Jumlah fitur optimal diperoleh 4 fitur yaitu jumlah *indegree* komunitas, jumlah anggota komunitas, rasio URL komunitas, dan rasio *unique URL* komunitas.

Fitur optimal *account* berdasarkan hasil tersebut diketahui bahwa terdapat 4 fitur yang selalu ada yaitu umur *account*, panjang *screen name*, *similarity screen name* dan *username*, dan rasio *following*. Umur *account* pada *account* terverifikasi memiliki angka yang besar atau diartikan telah lama dibuat. Hal ini lah yang

membuat umur akun menjadi penyebab kenapa umur akun menjadi penting. Contoh umur *account* terverifikasi @pak_jk yaitu 3072 hari.

Fitur *similarity screen name* dan *username* menunjukkan bahwa keterkaitan antara nama yang dipasang dan *username* tersebut tidak berbeda jauh. Fitur rasio *following* memiliki pengaruh karena *account* terverifikasi memiliki jumlah *following* yang sedikit dibandingkan dengan jumlah *follower*. Contoh rasio *following* @pak_jk yaitu 0,000248. Hal ini menunjukkan bahwa jumlah *follower* atau pengikut lebih banyak dari jumlah *following*.

Fitur optimal *tweet* berdasarkan hasil bahwa fitur optimal *tweet* memiliki 6 fitur yang tidak berubah-ubah pada pembagian persentase data. Fitur optimal tersebut yaitu rata-rata panjang *tweet*, rasio *unique URL*, jumlah *hashtag* pada *account*, rasio *hashtag* pada *account*, rasio *unique hashtag* pada *account*, dan jumlah kata *spam*. Rata-rata panjang *account* terverifikasi yaitu lebih dari 100 karakter. Hal ini menunjukkan bahwa akun terverifikasi melakukan *tweet* untuk hal-hal yang perlu disampaikan dan bukan sekedar *tweet* pendek. Contoh rata-rata panjang *tweet* @pak_jk yaitu 108 karakter.

Rasio *unique URL* memperlihatkan bahwa *account* terverifikasi jarang mengulang *tweet* dengan URL yang sama. Hal ini membuat rasio *unique URL* mendekati 1 bahkan ada yang 1. Contoh rasio *unique URL* @pak_jk yaitu 1. Jumlah *hashtag* pada *account* terverifikasi digunakan hanya sedikit, *hashtag* digunakan untuk hal-hal penting. Hal ini berbeda dengan *account* biasa yang menggunakan *hashtag* sesuai keperluannya, sehingga bisa lebih dari satu *hashtag* yang digunakan.

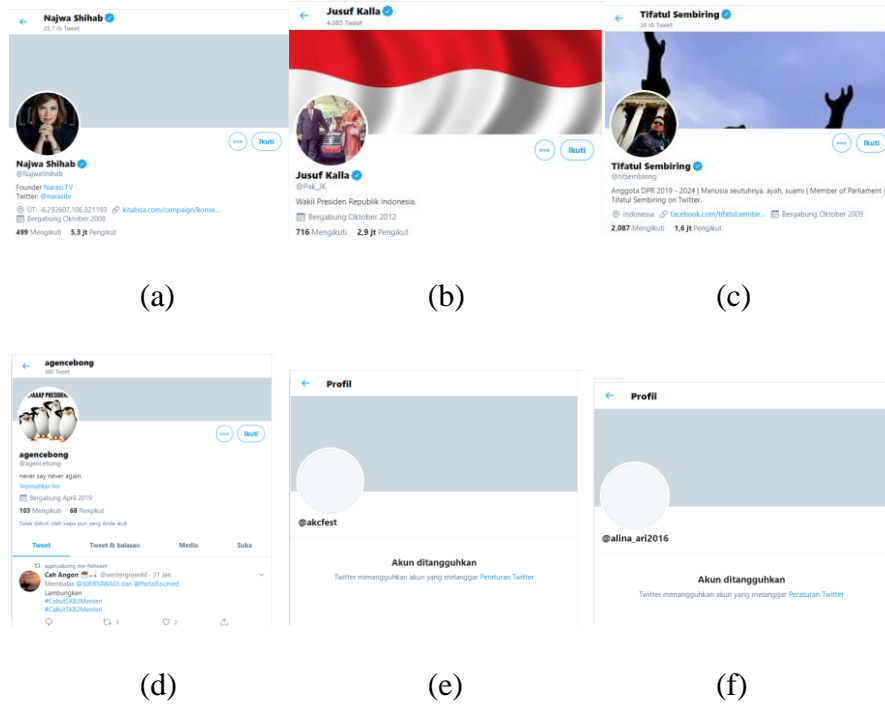
Rasio *hashtag account* memperlihatkan *tweet* pada *account* terverifikasi memiliki nilai kecil. Hal ini dikarenakan panjang karakter *tweet* selain *hashtag* memiliki nilai tinggi. Contoh rasio *hashtag account* @pak_jk yaitu 0,13374. Rasio *unique hashtag account* menunjukkan bahwa *account* terverifikasi memiliki tingkat *unique hashtag* lebih besar dari 0,5 bahkan mendekati 1. Hal ini

memperlihatkan *hashtag* yang digunakan tidak dilakukan secara berulang-ulang secara berlebihan. Contoh rasio *unique hashtag account @pak_jk* yaitu 0,52577.

Rasio *unique hashtag* komunitas menjadi fitur optimal pada fitur komunitas. Hal ini dikarenakan tingkat *unique hashtag* komunitas memberikan tanda mengenai *hashtag* yang digunakan pada suatu komunitas. Jumlah anggota komunitas juga menjadi fitur optimal. Jumlah anggota komunitas menunjukkan besar kecilnya suatu komunitas. Jumlah *indegree* komunitas menjadi bagian yang tidak terpisahkan dari jumlah anggota komunitas, karena jumlah *indegree* komunitas akan terpengaruh oleh jumlah anggota komunitas.

4.5. Deteksi Spammer

Hasil dan Analisa deteksi *spammer* dibagi 2 yaitu pertama perbandingan hasil deteksi *spammer* menggunakan *multi-layer perceptron* (MLP), *support vector machine* (SVM), dan paper perbandingan (Perdana, Muliawati, dan Alexandro, 2015) dengan Judul “*Bot Spammer Detection in Twitter Using Tweet Similarity and Time Interval Entropy*”. Kedua Analisa terhadap bobot digunakan pada deteksi *spammer* dengan menggabungkan fitur *account*, *tweet*, dan komunitas yaitu *alfa*, *beta*, dan *gamma*. Jumlah *dataset* digunakan yaitu 575.851 *username*. *Dataset spammer* yaitu 2.312 *username*. *Dataset legitimate* atau sah yaitu 573.539 *username*. Pembagian *dataset* dengan *persentase* 70:30 menghasilkan data latih 403.095 *username* dan data uji 170.444 *username*. Data uji terdiri atas 694 *username* data *spammer* dan 172.062 *username* data *legitimate*. Data latih terdiri atas 1.618 *username* data *spammer* dan 401.477 *username* data *legitimate*. Data latih dilakukan proses penyeimbangan data *spammer* dan *legitimate* dengan menggunakan *smote*. Hasil proses *smote* yaitu data *spammer* 401.477 *username* dan data *legitimate* 401.477 *username*. Gambar 4.5 merupakan contoh *username* pada *dataset*.



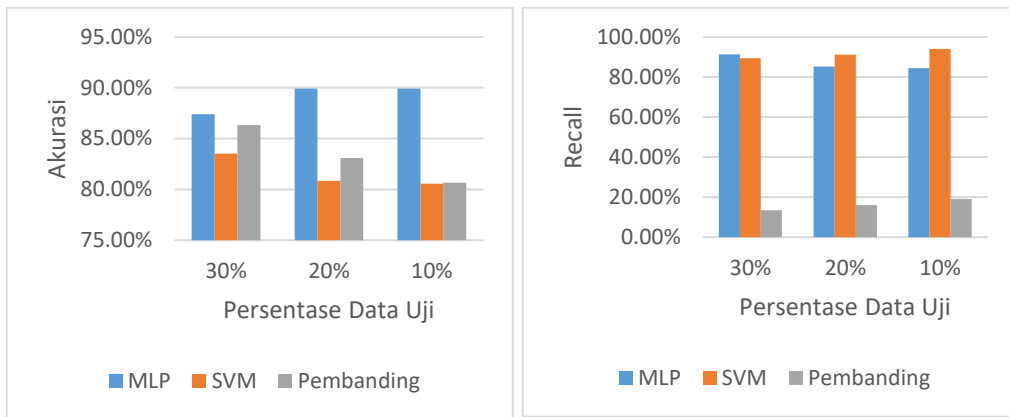
Gambar 4.5 Contoh *dataset username*, (a) (b) (c) merupakan *dataset legitimate*, (d) (e) (f) merupakan *dataset spammer*

A. Perbandingan MLP, SVM dan pembandingan

Pengujian dan perbandingan ini dilakukan untuk mengetahui tingkat keberhasilan dari deteksi *spammer*. Pembandingan yang digunakan yaitu paper (Perdana, Muliawati, dan Alexandro, 2015) dengan Judul “*Bot Spammer Detection in Twitter Using Tweet Similarity and Time Interval Entropy*”. Pengukuran perbandingan evaluasi yang digunakan yaitu *G-mean*. Evaluasi *G-mean* digunakan karena pada deteksi *spammer* juga harus mempertimbangkan kelas *legitimate* selain kelas *spammer*. Evaluasi pengukuran *G-mean* digunakan karena sesuai dengan prinsip dari *G-mean*. *G-mean* merupakan pengukuran yang mempertimbangkan kelas positif dan kelas negatif. Selain *G-mean*, evaluasi menampilkan hasil dari akurasi, *recall*, dan presisi. Gambar 4.6 merupakan hasil perbandingan evaluasi akurasi, *recall*, presisi, dan *G-mean*. Evaluasi akurasi secara keseluruhan mendapatkan nilai lebih dari 80 persen. Evaluasi *recall multi-*

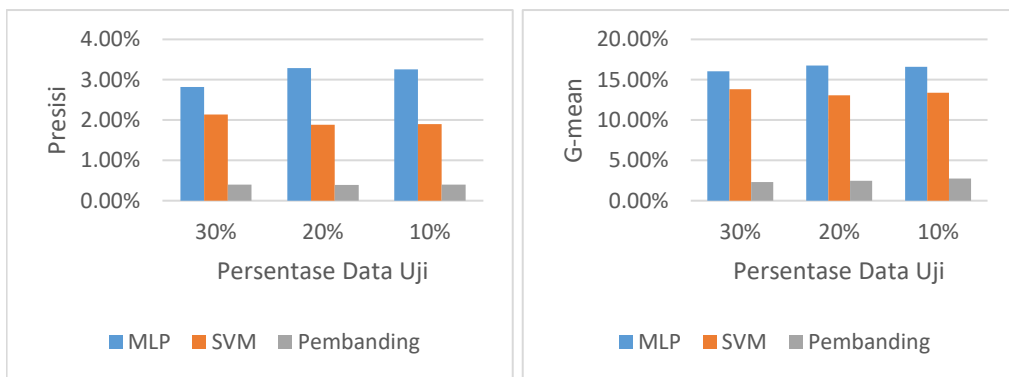
layer perceptron dan *support vector machine* mendapatkan nilai *recall* diatas 80 persen, sedangkan paper pembandingan mendapatkan hasil dibawah 80 persen. Hal ini memperlihatkan keberhasilan dari metode usulan dalam mendeteksi *spammer* pada data *spammer* lebih baik daripada paper pembandingan. Evaluasi presisi keseluruhan dibawah 10 persen. Hal ini dikarenakan jumlah *account legitimate* yang dikategorikan sebagai *spammer* memiliki jumlah yang jauh lebih besar daripada *account spammer*. Jumlah yang jauh lebih besar tersebut dikategorikan sebagai *spammer*. *Account legitimate* terdeteksi sebagai *spammer* dikarenakan nilai rata-rata panjang *tweet* kecil. Selain itu, *account* resmi seperti @provjateng membuat *tweet* dengan menyertakan *hashtag* dan URL/link. Fungsi dan tujuan *hashtag* dibuat untuk memberikan informasi tetapi selalu berulang-ulang dan sama. Bahkan terdapat *tweet* yang hanya berisi *hashtag* dan gambar atau *hashtag* dan *link*. Hal-hal tersebut juga terdapat pada *account legitimate* lainnya yang terdeteksi *spammer*, seperti @PoldaJogja, @pd_panjang, dan lain lain. Tetapi, *multi-layer perceptron* dan *support vector machine* mendapatkan hasil presisi lebih baik daripada paper pembandingan. Tabel 4.8 memperlihatkan presisi rendah, dimana hasil prediksi *spammer* mendapatkan *legitimate* lebih banyak. Evaluasi *g-mean* menunjukkan usulan lebih baik daripada paper pembandingan. Hal ini dihitung berdasarkan *recall* dan presisi. *G-mean* pada paper pembandingan mendapatkan nilai antara 2-3 persen. *G-mean* usulan mendapatkan 16-17 persen.

Gambar 4.6 (b) dan (c) menunjukkan bahwa jarak antara *recall* dan presisi antara usulan dan paper pembandingan memiliki jarak atau *gap* yang jauh. Hal ini dikarenakan prediksi *account spammer* banyak didapatkan tetapi yang benar-benar *account spam* sangat sedikit. Hal ini dikarenakan paper pembandingan menggunakan dua masukan informasi yaitu *time interval entropy* dan *similarity tweet*. Pengaruh *tweet similarity* sangat kuat untuk menyatakan bahwa suatu akun dinyatakan *account spam* padahal akun tersebut *legitimate*. Contohnya *account* pemerintah kabupaten Ciamis. *Account* pemerintahan kabupaten Ciamis melakukan *tweet* dengan foto dan keterangan *hashtag* yang kadang selalu sama



(a)

(b)



(c)

(d)

Gambar 4.6 Perbandingan evaluasi pengujian, (a) akurasi, (b) *recall*, (c) presisi, dan (d) G-mean.

Tabel 4.9 Hasil *multi-layer perceptron* persentase dataset 70:30, lr 0,1 neuron hidden 15, bobot pengabungan fitur $\alpha = 0,45$, $\beta = 0,45$, dan $\gamma = 0,1$

		<i>Predicted</i>	
		<i>Spammer</i>	<i>Legitimate</i>
<i>Actual</i>	<i>Spammer</i>	633	61
	<i>Legitimate</i>	21836	150226

yaitu #Kabar_Ciamis #Pembkab_Ciamis. Sedangkan, *account spam* dinyatakan sebagai *account legitimate* dikarenakan *similarity* rendah dengan cara melakukan *tweet* berbeda serta melakukan banyak *retweet* pada aktivitasnya.

B. Analisa bobot α , β , dan γ

Tujuan analisa bobot yaitu untuk mengetahui pengaruh bobot α , β , dan γ terhadap kelompok fitur (*account*, *tweet*, dan komunitas). Tabel 4.9 merupakan hasil dari nilai bobot α , β , dan γ , *multi-layer perceptron* menggunakan neuron *hidden* 15, *learning rate* 0,1, dan persentase *dataset* 70:30. Tabel 4.9 memperlihatkan pengaruh bobot γ terhadap fitur komunitas dimana semakin kecil nilainya maka keberhasilannya akan meningkat. Jika bobot γ lebih besar dari 0,5 maka hasil yang diperoleh semakin kecil. Hal ini menandakan fitur komunitas memiliki pengaruh untuk deteksi *spammer* akan tetapi paling kecil daripada fitur *account* dan *tweet*. Fitur paling berpengaruh berdasarkan bobot yang digunakan yaitu fitur *tweet*. Hal ini dikarenakan kegiatan *tweet* yang membuat *account* dinyatakan sebagai *spammer*, seperti penggunaan *hashtag*, *mention*, URL, perulangan *tweet*, dan lain sebagainya. Fitur *account* menjadi fitur berpengaruh kedua di antara fitur *tweet* dan fitur komunitas karena fitur *account* memberikan ciri-ciri *account* yang masih baru atau sengaja dibuat untuk keadaan tertentu. Fitur *account* mengalami perubahan sesuai keinginan pemegang *account* seperti jumlah *following* dan *followernya* sehingga akan mengakibatkan rasio keduanya akan berubah-ubah.

Tabel 4.10 Nilai bobot α , β , dan γ MLP *neuron hidden* 15 Lr 0,1 Persentase *dataset* 70:30

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	87,08%	89,19%	2,71%	15,54%
2	0,2	0,3	0,5	86,38%	90,06%	2,60%	15,29%
3	0,25	0,25	0,5	83,90%	90,78%	2,22%	14,20%
4	0,3	0,2	0,5	78,50%	91,79%	1,69%	12,45%
5	0,4	0,1	0,5	67,59%	91,74%	1,13%	10,17%
6	0,1	0,3	0,6	86,82%	88,90%	2,65%	15,34%
7	0,2	0,2	0,6	81,53%	90,35%	1,93%	13,21%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
8	0,3	0,1	0,6	68,66%	97,07%	1,17%	10,37%
9	0,1	0,2	0,7	81,09%	88,76%	1,86%	12,83%
10	0,15	0,15	0,7	76,81%	89,91%	1,54%	11,75%
11	0,2	0,1	0,7	69,13%	90,20%	1,16%	10,24%
12	0,1	0,1	0,8	70,37%	87,46%	1,17%	10,13%
13	0,05	0,05	0,9	61,66%	81,12%	0,84%	8,28%
14	0,1	0,5	0,4	87,08%	89,19%	2,71%	15,54%
15	0,2	0,4	0,4	87,15%	89,91%	2,74%	15,70%
16	0,3	0,3	0,4	85,57%	91,21%	2,48%	15,05%
17	0,4	0,2	0,4	75,27%	93,37%	1,50%	11,82%
18	0,5	0,1	0,4	66,66%	92,07%	1,10%	10,06%
19	0,1	0,6	0,3	87,10%	89,19%	2,71%	15,55%
20	0,2	0,5	0,3	87,14%	89,63%	2,73%	15,65%
21	0,3	0,4	0,3	87,26%	90,63%	2,79%	15,89%
22	0,35	0,35	0,3	86,58%	91,35%	2,67%	15,62%
23	0,4	0,3	0,3	83,16%	91,93%	2,15%	14,06%
24	0,5	0,2	0,3	72,64%	93,95%	1,36%	11,31%
25	0,6	0,1	0,3	65,88%	91,79%	1,07%	9,91%
26	0,1	0,7	0,2	87,15%	88,76%	2,71%	15,51%
27	0,2	0,6	0,2	87,16%	89,48%	2,73%	15,64%
28	0,3	0,5	0,2	87,02%	90,06%	2,76%	15,76%
29	0,4	0,4	0,2	87,05%	91,50%	2,77%	15,91%
30	0,5	0,3	0,2	79,65%	92,94%	1,80%	12,95%
31	0,6	0,2	0,2	70,71%	93,66%	1,27%	10,90%
32	0,7	0,1	0,2	65,39%	91,79%	1,06%	9,84%
33	0,1	0,8	0,1	87,17%	88,47%	2,71%	15,47%
34	0,2	0,7	0,1	87,18%	89,19%	2,73%	15,60%
35	0,3	0,6	0,1	87,19%	89,91%	2,75%	15,73%
36	0,4	0,5	0,1	87,37%	90,78%	2,81%	15,98%
37	0,45	0,45	0,1	87,32%	91,21%	2,82%	16,03%
38	0,5	0,4	0,1	85,71%	91,93%	2,52%	15,23%
39	0,6	0,3	0,1	76,73%	93,52%	1,59%	12,20%
40	0,7	0,2	0,1	69,41%	93,52%	1,21%	10,66%
41	0,8	0,1	0,1	65,18%	91,50%	1,05%	9,78%

[Halaman ini sengaja dikosongkan]

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Pengujian dan analisa yang telah dilakukan menghasilkan kesimpulan penelitian sebagai berikut:

1. Pengembangan ekstraksi fitur berdasarkan *hashtag* dan aktivitas komunitas berhasil untuk deteksi *account spammer* di *Twitter*.
2. Hasil deteksi *spammer* mendapatkan nilai akurasi, *recall*, presisi, dan *g-mean* yaitu 90,6%, 88,0%, 3,7%, dan 18%.
3. Pengaruh fitur *tweet* merupakan paling tinggi dibandingkan fitur *account* dan fitur komunitas. Fitur *tweet* di dalamnya terdapat *hashtag*, *mention*, dan URL.
4. Fitur komunitas dilihat berdasarkan bobot yang digunakan, dengan hasil yang bagus lebih kecil dari 0,3. Sehingga fitur komunitas memiliki pengaruh tetapi lebih kecil dibandingkan fitur *account* dan fitur *tweet*.
5. Hasil fitur seleksi pada fitur *tweet* setiap pembagian persentase *dataset* tidak mengalami perubahan yaitu 6 fitur. Pada fitur *tweet* terdapat fitur jumlah *hashtag* dan *unique hashtag*.

5.2.Saran

Berdasarkan penelitian yang telah dilakukan, saran yang diberikan untuk meningkatkan keberhasilan yaitu memasukkan fitur-fitur lainnya terkait *tweet*, *account*, dan komunitas yang belum dipergunakan pada penelitian ini.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- Aditya, C. S. K., Hani'ah, M., Fitrawan, A. A., Arifin, A. Z., & Purwitasari, D. (2016). Deteksi Bot Spammer pada Twitter Berbasis Sentiment Analysis dan Time Interval Entropy. *Jurnal Buana Informatika*, 7(3), 179–186. <https://doi.org/10.24002/jbi.v7i3.656>
- Bagić Babac, M., & Podobnik, V. (2016). A sentiment analysis of who participates, how and why, at social media sport websites. *Online Information Review*, 40(6), 814–833. <https://doi.org/10.1108/OIR-02-2016-0050>
- Benevenuto, F., Magno, G., Rodrigues, T., & Almeida, V. (2010). Detecting Spammers on Twitter. *Proc. Collaboration Electron. Messaging Anti-Abuse Spam Conf. (CEAS)*, 6, 3156–3164. <https://doi.org/10.1021/bi972148>
- Bhat, S. Y., & Abulaish, M. (2013). Community-based features for identifying spammers in online social networks. *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM '13*, 100–107. <https://doi.org/10.1145/2492517.2492567>
- Bindu, P. V., Mishra, R., & Thilagam, P. S. (2018). Discovering spammer communities in twitter. *Journal of Intelligent Information Systems*, 51(3), 503–527. <https://doi.org/10.1007/s10844-017-0494-z>
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008. <https://doi.org/10.1088/1742-5468/2008/10/P10008>
- Chapelle, O., Scholkopf, B., & Zien, A. (2006). *Deep Semi-Supervised Learning*. MIT Press.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002).

- SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>
- Chen, C., Zhang, J., Xiang, Y., Zhou, W., & Oliver, J. (2016, March). Spammers Are Becoming “Smarter” on Twitter. *IT Professional*, 18(2), 66–70. <https://doi.org/10.1109/MITP.2016.36>
- Chen, C., Zhang, J., Xie, Y., Xiang, Y., Zhou, W., Hassan, M. M., AlElaiwi, A., & Alrubaian, M. (2015). A Performance Evaluation of Machine Learning-Based Streaming Spam Tweets Detection. *IEEE Transactions on Computational Social Systems*, 2(3), 65–76. <https://doi.org/10.1109/TCSS.2016.2516039>
- Chien, K., Miils, E., & Airey, C. (2008). Perfect Match ? Generation Y as Change Agents for Information Communication Technology Implementation in Healthcare. *Stud. Health Technol. Inf*, 496–501.
- Contractor, D., Chawda, B., Mehta, S., Subramaniam, L. V., & Faruque, T. A. (2015). Tracking political elections on social media: Applications and experience. *IJCAI International Joint Conference on Artificial Intelligence*, 2320–2326.
- Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 297, 273–297. <https://doi.org/10.1023/A:1022627411411>
- Eshraqi, N., Jalali, M., & Moattar, M. H. (2015). Detecting spam tweets in Twitter using a data stream clustering algorithm. *2015 International Congress on Technology, Communication and Knowledge (ICTCK)*, 347–351. <https://doi.org/10.1109/ICTCK.2015.7582694>
- Fletcher, J. M., & Wennekers, T. (2018). From Structure to Activity: Using Centrality Measures to Predict Neuronal Activity. *International Journal of Neural Systems*, 28(02), 1750013. <https://doi.org/10.1142/S0129065717500137>

- Fortunato, S., & Castellano, C. (2012). Community Structure in Graphs. In R. A. Meyers (Ed.), *Computational Complexity* (pp. 490–512). Springer New York. https://doi.org/10.1007/978-1-4614-1800-9_33
- Guyon, I., & Weston, J. A. E. (2019). *Recursive feature elimination method using support vector machines*.
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning*, 46(1), 389–422. <https://doi.org/10.1023/A:1012487302797>
- Inuwa-Dutse, I., Liptrott, M., & Korkontzelos, I. (2018). Detection of spam-posting accounts on Twitter. *Neurocomputing*, 315(agustus), 496–511. <https://doi.org/10.1016/j.neucom.2018.07.044>
- Jain, G., Sharma, M., & Agarwal, B. (2019). Spam detection in social media using convolutional and long short term memory neural network. *Annals of Mathematics and Artificial Intelligence*, 85(1), 21–44. <https://doi.org/10.1007/s10472-018-9612-z>
- Japkowicz, N. (2013). Assessment Metrics for Imbalanced Learning. In *Imbalanced Learning* (pp. 187–206). John Wiley & Sons, Inc. <https://doi.org/10.1002/9781118646106.ch8>
- Liu, H., Morstatter, F., Tang, J., & Zafarani, R. (2016). The good, the bad, and the ugly: uncovering novel research opportunities in social media mining. *International Journal of Data Science and Analytics*, 1(3–4), 137–143. <https://doi.org/10.1007/s41060-016-0023-0>
- Martinez-Romo, J., & Araujo, L. (2013). Detecting malicious tweets in trending topics using a statistical analysis of language. *Expert Systems with Applications*, 40(8), 2992–3000. <https://doi.org/10.1016/j.eswa.2012.12.015>
- Masood, F., Ammad, G., Almogren, A., Abbas, A., Khattak, H. A., Ud Din, I.,

- Guizani, M., & Zuair, M. (2019). Spammer Detection and Fake User Identification on Social Networks. *IEEE Access*, 7(3), 68140–68152. <https://doi.org/10.1109/ACCESS.2019.2918196>
- Park, D., Lee, M., Park, S., Seong, J.-K., & Youn, I. (2018). Determination of Optimal Heart Rate Variability Features Based on SVM-Recursive Feature Elimination for Cumulative Stress Monitoring Using ECG Sensor. *Sensors*, 18(7), 2387. <https://doi.org/10.3390/s18072387>
- Perdana, R. S., Muliawati, T. H., & Alexandro, R. (2015). Bot Spammer Detection in Twitter Using Tweet Similarity and Time Interval Entropy. *Jurnal Ilmu Komputer Dan Informasi*, 8(1), 19. <https://doi.org/10.21609/jiki.v8i1.280>
- Rojas, E., Munoz-Gama, J., Sepúlveda, M., & Capurro, D. (2016). Process mining in healthcare: A literature review. *Journal of Biomedical Informatics*, 61(June), 224–236. <https://doi.org/10.1016/j.jbi.2016.04.007>
- Rosengren, E., Radev, D. R., Mei, Q., & Arbor, A. (2011). Rumor Has It- Identifying Misinformation in Microblogs. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1589–1599. <https://dl.acm.org/citation.cfm?id=2145602>
- Rozen, A. (2017). *Giving you more characters to express yourself*. Blog.Twitter.Com. https://blog.twitter.com/official/en_us/topics/product/2017/Giving-you-more-characters-to-express-yourself.html
- Sarlati, Y., Hashemi, S., & Mozaffari, N. (2015). Adopting Community Features to Detect Social Spammers. *2015 European Intelligence and Security Informatics Conference*, 153–156. <https://doi.org/10.1109/EISIC.2015.44>
- Siang, J. J. (2005). Jaringan Syaraf Tiruan Dan Pemrogramannya Dengan Matlab. In *andi yogyakarta*.

- Twitter. (2019). *Kebijakan terkait manipulasi platform dan spam*.
Help.Twitter.Com. <https://help.twitter.com/id/rules-and-policies/platform-manipulation>
- Yang, C., Harkreader, R. C., & Gu, G. (2011). Die Free or Live Hard? Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. *Springer Berlin Heidelberg, 11(I)*, 318–337.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.297.8133&rep=rep1&type=pdf>
- Yang, C., Harkreader, R., Zhang, J., Shin, S., & Gu, G. (2012). Analyzing spammers' social networks for fun and profit. *Proceedings of the 21st International Conference on World Wide Web - WWW '12*, 71.
<https://doi.org/10.1145/2187836.2187847>

LAMPIRAN

Lampiran 1 Nilai bobot, MLP *neuron* hidden 15 Lr 0,01 Persentase dataset 70:30

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	88,71%	85,99%	2,95%	15,91%
2	0,2	0,3	0,5	89,53%	86,17%	3,22%	16,65%
3	0,25	0,25	0,5	89,41%	85,59%	3,16%	16,45%
4	0,3	0,2	0,5	87,53%	85,59%	2,69%	15,19%
5	0,4	0,1	0,5	77,37%	83,14%	1,46%	11,01%
6	0,1	0,3	0,6	89,17%	85,30%	3,08%	16,22%
7	0,2	0,2	0,6	88,50%	84,01%	2,87%	15,52%
8	0,3	0,1	0,6	80,16%	82,28%	1,64%	11,63%
9	0,1	0,2	0,7	87,93%	83,00%	2,70%	14,98%
10	0,15	0,15	0,7	86,36%	81,41%	2,35%	13,84%
11	0,2	0,1	0,7	82,05%	78,96%	1,74%	11,74%
12	0,1	0,1	0,8	81,76%	70,32%	1,53%	10,39%
13	0,05	0,05	0,9	76,46%	56,63%	0,96%	7,39%
14	0,1	0,5	0,4	88,44%	86,31%	2,93%	15,89%
15	0,2	0,4	0,4	88,93%	86,74%	3,07%	16,31%
16	0,3	0,3	0,4	89,73%	86,60%	3,29%	16,89%
17	0,4	0,2	0,4	85,25%	85,73%	2,29%	14,01%
18	0,5	0,1	0,4	75,07%	83,86%	1,34%	10,59%
19	0,1	0,6	0,3	88,30%	86,60%	2,90%	15,85%
20	0,2	0,5	0,3	88,56%	87,18%	2,98%	16,13%
21	0,3	0,4	0,3	89,34%	87,75%	3,21%	16,79%
22	0,35	0,35	0,3	89,74%	87,46%	3,33%	17,06%
23	0,4	0,3	0,3	89,11%	86,89%	3,12%	16,46%
24	0,5	0,2	0,3	82,11%	86,60%	1,91%	12,87%
25	0,6	0,1	0,3	73,48%	84,73%	1,27%	10,37%
26	0,1	0,7	0,2	88,21%	86,60%	2,88%	15,79%
27	0,2	0,6	0,2	88,38%	87,32%	2,94%	16,03%
28	0,3	0,5	0,2	88,77%	87,90%	3,06%	16,41%
29	0,4	0,4	0,2	89,64%	87,61%	3,30%	17,00%
30	0,5	0,3	0,2	87,56%	87,46%	2,76%	15,53%
31	0,6	0,2	0,2	79,29%	86,74%	1,66%	11,99%
32	0,7	0,1	0,2	72,25%	85,30%	1,22%	10,21%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
33	0,1	0,8	0,1	88,16%	86,60%	2,87%	15,76%
34	0,2	0,7	0,1	88,29%	87,03%	2,91%	15,92%
35	0,3	0,6	0,1	88,50%	87,46%	2,98%	16,14%
36	0,4	0,5	0,1	89,19%	88,04%	3,18%	16,74%
37	0,45	0,45	0,1	90,55%	88,04%	3,67%	17,97%
38	0,5	0,4	0,1	89,17%	87,46%	3,16%	16,62%
39	0,6	0,3	0,1	84,97%	86,74%	2,27%	14,05%
40	0,7	0,2	0,1	76,88%	87,61%	1,50%	11,47%
41	0,8	0,1	0,1	71,33%	85,59%	1,19%	10,08%

Lampiran 2 Nilai bobot, MLP neuron hidden 15 Lr 0,001 Persentase dataset 70:30

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	87,72%	87,03%	2,78%	15,55%
2	0,2	0,3	0,5	88,38%	88,04%	2,97%	16,16%
3	0,25	0,25	0,5	87,94%	87,46%	2,84%	15,77%
4	0,3	0,2	0,5	85,39%	86,74%	2,34%	14,24%
5	0,4	0,1	0,5	76,32%	85,30%	1,43%	11,04%
6	0,1	0,3	0,6	87,94%	87,03%	2,83%	15,69%
7	0,2	0,2	0,6	86,55%	86,31%	2,52%	14,76%
8	0,3	0,1	0,6	77,43%	84,29%	1,48%	11,18%
9	0,1	0,2	0,7	85,42%	85,45%	2,31%	14,05%
10	0,15	0,15	0,7	83,12%	84,58%	1,98%	12,94%
11	0,2	0,1	0,7	78,11%	82,28%	1,49%	11,08%
12	0,1	0,1	0,8	76,88%	78,24%	1,35%	10,26%
13	0,05	0,05	0,9	68,95%	68,88%	0,89%	7,82%
14	0,1	0,5	0,4	87,55%	87,90%	2,77%	15,60%
15	0,2	0,4	0,4	88,00%	88,76%	2,90%	16,03%
16	0,3	0,3	0,4	88,77%	88,33%	3,07%	16,48%
17	0,4	0,2	0,4	82,83%	87,32%	2,01%	13,24%
18	0,5	0,1	0,4	75,16%	84,44%	1,35%	10,68%
19	0,1	0,6	0,3	87,47%	88,18%	2,76%	15,60%
20	0,2	0,5	0,3	87,71%	88,90%	2,83%	15,87%
21	0,3	0,4	0,3	88,54%	89,19%	3,04%	16,48%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
22	0,35	0,35	0,3	89,09%	89,05%	3,19%	16,85%
23	0,4	0,3	0,3	88,35%	87,90%	2,95%	16,11%
24	0,5	0,2	0,3	80,32%	87,18%	1,75%	12,36%
25	0,6	0,1	0,3	74,30%	84,01%	1,30%	10,45%
26	0,1	0,7	0,2	87,42%	88,04%	2,74%	15,54%
27	0,2	0,6	0,2	87,61%	88,18%	2,79%	15,69%
28	0,3	0,5	0,2	88,03%	89,19%	2,92%	16,13%
29	0,4	0,4	0,2	89,19%	89,19%	3,22%	16,95%
30	0,5	0,3	0,2	86,51%	88,18%	2,57%	15,05%
31	0,6	0,2	0,2	78,51%	87,32%	1,61%	11,86%
32	0,7	0,1	0,2	73,73%	83,43%	1,26%	10,26%
33	0,1	0,8	0,1	87,38%	88,04%	2,74%	15,52%
34	0,2	0,7	0,1	87,55%	88,62%	2,79%	15,72%
35	0,3	0,6	0,1	87,80%	89,05%	2,86%	15,96%
36	0,4	0,5	0,1	88,60%	89,48%	3,07%	16,57%
37	0,45	0,45	0,1	89,18%	89,19%	3,22%	16,94%
38	0,5	0,4	0,1	89,24%	88,76%	3,22%	16,91%
39	0,6	0,3	0,1	83,54%	87,90%	2,11%	13,61%
40	0,7	0,2	0,1	77,23%	87,32%	1,52%	11,52%
41	0,8	0,1	0,1	73,32%	83,72%	1,25%	10,22%

Lampiran 3 Nilai bobot, MLP *neuron* hidden 15 Lr 0,1 Persentase dataset 80:20

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,24%	85,06%	2,43%	14,38%
2	0,2	0,3	0,5	88,24%	83,98%	2,80%	15,34%
3	0,25	0,25	0,5	88,89%	82,90%	2,92%	15,57%
4	0,3	0,2	0,5	87,74%	79,65%	2,56%	14,27%
5	0,4	0,1	0,5	80,17%	78,57%	1,57%	11,11%
6	0,1	0,3	0,6	87,11%	83,33%	2,54%	14,55%
7	0,2	0,2	0,6	87,81%	80,95%	2,61%	14,54%
8	0,3	0,1	0,6	81,60%	76,41%	1,65%	11,22%
9	0,1	0,2	0,7	86,77%	77,92%	2,32%	13,45%
10	0,15	0,15	0,7	85,04%	76,19%	2,01%	12,39%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
11	0,2	0,1	0,7	81,58%	72,73%	1,57%	10,68%
12	0,1	0,1	0,8	77,57%	69,26%	1,23%	9,23%
13	0,05	0,05	0,9	70,52%	58,66%	0,80%	6,83%
14	0,1	0,5	0,4	85,92%	86,15%	2,41%	14,40%
15	0,2	0,4	0,4	86,98%	85,93%	2,59%	14,92%
16	0,3	0,3	0,4	89,31%	83,77%	3,07%	16,03%
17	0,4	0,2	0,4	86,57%	79,00%	2,32%	13,54%
18	0,5	0,1	0,4	78,64%	80,30%	1,49%	10,94%
19	0,1	0,6	0,3	85,72%	86,36%	2,38%	14,33%
20	0,2	0,5	0,3	86,30%	86,58%	2,48%	14,66%
21	0,3	0,4	0,3	88,21%	86,36%	2,87%	15,74%
22	0,35	0,35	0,3	89,41%	85,06%	3,14%	16,34%
23	0,4	0,3	0,3	89,57%	83,12%	3,12%	16,10%
24	0,5	0,2	0,3	84,74%	79,87%	2,07%	12,85%
25	0,6	0,1	0,3	77,41%	81,60%	1,43%	10,81%
26	0,1	0,7	0,2	85,60%	86,36%	2,36%	14,27%
27	0,2	0,6	0,2	86,04%	86,80%	2,44%	14,56%
28	0,3	0,5	0,2	87,09%	86,36%	2,62%	15,06%
29	0,4	0,4	0,2	89,38%	85,06%	3,13%	16,32%
30	0,5	0,3	0,2	88,82%	82,25%	2,88%	15,40%
31	0,6	0,2	0,2	82,72%	80,52%	1,84%	12,18%
32	0,7	0,1	0,2	76,50%	81,82%	1,38%	10,63%
33	0,1	0,8	0,1	85,54%	86,58%	2,35%	14,28%
34	0,2	0,7	0,1	85,85%	86,58%	2,40%	14,43%
35	0,3	0,6	0,1	86,47%	86,58%	2,51%	14,75%
36	0,4	0,5	0,1	88,30%	85,71%	2,87%	15,68%
37	0,45	0,45	0,1	89,35%	85,50%	3,14%	16,37%
38	0,5	0,4	0,1	89,90%	85,28%	3,29%	16,76%
39	0,6	0,3	0,1	87,41%	81,60%	2,55%	14,42%
40	0,7	0,2	0,1	81,02%	82,03%	1,71%	11,85%
41	0,8	0,1	0,1	75,89%	82,25%	1,35%	10,55%

Lampiran 4 Nilai bobot, MLP *neuron* hidden 15 Lr 0,01 Persentase dataset 80:20

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	85,81%	87,45%	2,42%	14,55%
2	0,2	0,3	0,5	87,17%	86,58%	2,65%	15,14%
3	0,25	0,25	0,5	87,37%	87,23%	2,71%	15,36%
4	0,3	0,2	0,5	85,01%	85,93%	2,26%	13,92%
5	0,4	0,1	0,5	76,38%	83,77%	1,41%	10,86%
6	0,1	0,3	0,6	86,10%	87,01%	2,46%	14,62%
7	0,2	0,2	0,6	85,48%	85,50%	2,32%	14,07%
8	0,3	0,1	0,6	77,30%	82,47%	1,44%	10,90%
9	0,1	0,2	0,7	84,28%	85,50%	2,14%	13,53%
10	0,15	0,15	0,7	81,66%	83,77%	1,81%	12,30%
11	0,2	0,1	0,7	77,37%	82,90%	1,45%	10,97%
12	0,1	0,1	0,8	73,60%	77,49%	1,17%	9,51%
13	0,05	0,05	0,9	62,23%	73,81%	0,78%	7,59%
14	0,1	0,5	0,4	85,64%	87,88%	2,40%	14,53%
15	0,2	0,4	0,4	86,40%	88,10%	2,54%	14,96%
16	0,3	0,3	0,4	88,20%	87,45%	2,90%	15,92%
17	0,4	0,2	0,4	83,46%	85,28%	2,03%	13,17%
18	0,5	0,1	0,4	75,38%	83,55%	1,35%	10,61%
19	0,1	0,6	0,3	85,57%	87,88%	2,39%	14,50%
20	0,2	0,5	0,3	86,04%	88,31%	2,48%	14,81%
21	0,3	0,4	0,3	87,55%	87,66%	2,76%	15,55%
22	0,35	0,35	0,3	88,55%	87,66%	2,99%	16,20%
23	0,4	0,3	0,3	88,27%	85,71%	2,86%	15,66%
24	0,5	0,2	0,3	81,53%	85,50%	1,83%	12,50%
25	0,6	0,1	0,3	74,66%	83,98%	1,32%	10,51%
26	0,1	0,7	0,2	85,52%	87,66%	2,38%	14,44%
27	0,2	0,6	0,2	85,85%	88,31%	2,45%	14,71%
28	0,3	0,5	0,2	86,69%	88,10%	2,59%	15,12%
29	0,4	0,4	0,2	88,73%	87,23%	3,02%	16,24%
30	0,5	0,3	0,2	87,25%	85,93%	2,64%	15,08%
31	0,6	0,2	0,2	79,81%	84,85%	1,66%	11,88%
32	0,7	0,1	0,2	74,12%	83,98%	1,29%	10,40%
33	0,1	0,8	0,1	85,48%	87,66%	2,37%	14,42%
34	0,2	0,7	0,1	85,76%	88,10%	2,43%	14,63%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
35	0,3	0,6	0,1	86,22%	87,88%	2,50%	14,83%
36	0,4	0,5	0,1	87,82%	87,88%	2,82%	15,75%
37	0,45	0,45	0,1	88,81%	87,23%	3,05%	16,30%
38	0,5	0,4	0,1	89,09%	86,58%	3,10%	16,38%
39	0,6	0,3	0,1	85,33%	86,15%	2,31%	14,11%
40	0,7	0,2	0,1	78,39%	85,06%	1,56%	11,51%
41	0,8	0,1	0,1	73,74%	83,77%	1,27%	10,30%

Lampiran 5 Nilai bobot, MLP neuron hidden 15 Lr 0,001 Persentase dataset 80:20

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	85,92%	87,23%	2,43%	14,57%
2	0,2	0,3	0,5	87,07%	86,58%	2,63%	15,08%
3	0,25	0,25	0,5	86,68%	87,01%	2,56%	14,94%
4	0,3	0,2	0,5	84,00%	85,87%	2,07%	13,35%
5	0,4	0,1	0,5	73,65%	85,71%	1,29%	10,52%
6	0,1	0,3	0,6	86,43%	85,71%	2,48%	14,58%
7	0,2	0,2	0,6	85,26%	86,15%	2,30%	14,07%
8	0,3	0,1	0,6	75,70%	84,20%	1,37%	10,76%
9	0,1	0,2	0,7	84,99%	84,85%	2,23%	13,74%
10	0,15	0,15	0,7	82,34%	82,68%	1,85%	12,37%
11	0,2	0,1	0,7	77,64%	80,95%	1,44%	10,78%
12	0,1	0,1	0,8	75,45%	74,03%	1,20%	9,43%
13	0,05	0,05	0,9	67,54%	66,67%	0,82%	7,40%
14	0,1	0,5	0,4	85,64%	87,88%	2,40%	14,54%
15	0,2	0,4	0,4	86,24%	88,31%	2,52%	14,91%
16	0,3	0,3	0,4	87,49%	87,88%	2,75%	15,55%
17	0,4	0,2	0,4	81,40%	87,01%	1,85%	12,68%
18	0,5	0,1	0,4	71,91%	87,45%	1,24%	10,39%
19	0,1	0,6	0,3	85,50%	87,66%	2,38%	14,43%
20	0,2	0,5	0,3	85,83%	88,10%	2,44%	14,67%
21	0,3	0,4	0,3	86,94%	88,31%	2,65%	15,30%
22	0,35	0,35	0,3	87,76%	88,31%	2,82%	15,79%
23	0,4	0,3	0,3	86,93%	87,45%	2,62%	15,15%
24	0,5	0,2	0,3	78,11%	87,45%	1,58%	11,76%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
25	0,6	0,1	0,3	70,66%	88,31%	1,19%	10,27%
26	0,1	0,7	0,2	85,43%	87,88%	2,37%	14,43%
27	0,2	0,6	0,2	85,68%	88,31%	2,42%	14,62%
28	0,3	0,5	0,2	86,16%	88,53%	2,51%	14,91%
29	0,4	0,4	0,2	87,78%	88,10%	2,82%	15,77%
30	0,5	0,3	0,2	84,96%	87,66%	2,29%	14,17%
31	0,6	0,2	0,2	75,57%	88,53%	1,44%	11,27%
32	0,7	0,1	0,2	69,87%	88,10%	1,16%	10,11%
33	0,1	0,8	0,1	85,39%	87,88%	2,36%	14,41%
34	0,2	0,7	0,1	85,56%	88,10%	2,40%	14,53%
35	0,3	0,6	0,1	85,82%	88,31%	2,45%	14,70%
36	0,4	0,5	0,1	86,89%	88,74%	2,65%	15,34%
37	0,45	0,45	0,1	87,77%	88,96%	2,85%	15,91%
38	0,5	0,4	0,1	87,82%	88,31%	2,84%	15,83%
39	0,6	0,3	0,1	81,76%	87,66%	1,90%	12,89%
40	0,7	0,2	0,1	73,96%	90,26%	1,37%	11,13%
41	0,8	0,1	0,1	69,29%	88,53%	1,15%	10,07%

Lampiran 6 Nilai bobot, MLP *neuron* hidden 15 Lr 0,1 Persentase dataset 90:10

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,64%	82,68%	2,43%	14,19%
2	0,2	0,3	0,5	88,38%	83,98%	2,83%	15,43%
3	0,25	0,25	0,5	88,85%	83,12%	2,92%	15,58%
4	0,3	0,2	0,5	87,21%	82,68%	2,54%	14,49%
5	0,4	0,1	0,5	77,31%	87,45%	1,53%	11,55%
6	0,1	0,3	0,6	87,38%	82,68%	2,58%	14,59%
7	0,2	0,2	0,6	87,79%	82,25%	2,65%	14,75%
8	0,3	0,1	0,6	79,14%	83,98%	1,59%	11,57%
9	0,1	0,2	0,7	87,30%	80,09%	2,48%	14,10%
10	0,15	0,15	0,7	85,72%	80,09%	2,21%	13,31%
11	0,2	0,1	0,7	80,98%	80,95%	1,69%	11,68%
12	0,1	0,1	0,8	78,72%	73,16%	1,37%	10,00%
13	0,05	0,05	0,9	70,75%	60,61%	0,83%	7,09%
14	0,1	0,5	0,4	86,33%	82,68%	2,38%	14,03%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
15	0,2	0,4	0,4	87,13%	84,42%	2,58%	14,75%
16	0,3	0,3	0,4	89,29%	83,98%	3,07%	16,05%
17	0,4	0,2	0,4	84,78%	85,28%	2,21%	13,72%
18	0,5	0,1	0,4	75,40%	87,88%	1,42%	11,15%
19	0,1	0,6	0,3	86,12%	82,25%	2,33%	13,85%
20	0,2	0,5	0,3	86,59%	84,42%	2,47%	14,45%
21	0,3	0,4	0,3	88,13%	84,85%	2,80%	15,41%
22	0,35	0,35	0,3	89,34%	83,98%	3,08%	16,09%
23	0,4	0,3	0,3	89,43%	83,12%	3,08%	16,00%
24	0,5	0,2	0,3	81,63%	87,45%	1,88%	12,82%
25	0,6	0,1	0,3	74,14%	88,74%	1,36%	10,99%
26	0,1	0,7	0,2	86,06%	81,82%	2,31%	13,75%
27	0,2	0,6	0,2	86,34%	83,55%	2,41%	14,18%
28	0,3	0,5	0,2	87,09%	84,85%	2,58%	14,80%
29	0,4	0,4	0,2	89,30%	83,55%	3,06%	15,98%
30	0,5	0,3	0,2	88,05%	83,55%	2,74%	15,14%
31	0,6	0,2	0,2	79,18%	90,04%	1,71%	12,40%
32	0,7	0,1	0,2	73,26%	89,18%	1,32%	10,86%
33	0,1	0,8	0,1	86,00%	81,82%	2,30%	13,72%
34	0,2	0,7	0,1	86,21%	83,55%	2,38%	14,11%
35	0,3	0,6	0,1	86,57%	84,85%	2,48%	14,52%
36	0,4	0,5	0,1	88,14%	85,28%	2,82%	15,50%
37	0,45	0,45	0,1	89,24%	84,85%	3,08%	16,17%
38	0,5	0,4	0,1	89,90%	84,42%	3,26%	16,60%
39	0,6	0,3	0,1	85,37%	86,15%	2,32%	14,12%
40	0,7	0,2	0,1	77,55%	90,48%	1,59%	12,01%
41	0,8	0,1	0,1	72,55%	89,18%	1,29%	10,72%

Lampiran 7 Nilai bobot, MLP *neuron* hidden 15 Lr 0,01 Persentase dataset 90:10

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,78%	80,95%	2,41%	13,97%
2	0,2	0,3	0,5	88,00%	83,12%	2,72%	15,03%
3	0,25	0,25	0,5	87,35%	84,85%	2,63%	14,95%
4	0,3	0,2	0,5	83,45%	87,45%	2,08%	13,49%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
5	0,4	0,1	0,5	73,66%	90,04%	1,35%	11,04%
6	0,1	0,3	0,6	87,36%	81,82%	2,55%	14,43%
7	0,2	0,2	0,6	85,82%	83,12%	2,31%	13,85%
8	0,3	0,1	0,6	75,89%	87,01%	1,43%	11,16%
9	0,1	0,2	0,7	86,95%	80,52%	2,43%	13,99%
10	0,15	0,15	0,7	83,27%	81,39%	1,92%	12,51%
11	0,2	0,1	0,7	78,25%	83,55%	1,52%	11,28%
12	0,1	0,1	0,8	77,36%	72,73%	1,28%	9,64%
13	0,05	0,05	0,9	68,35%	60,61%	0,77%	6,81%
14	0,1	0,5	0,4	86,50%	81,82%	2,39%	13,97%
15	0,2	0,4	0,4	87,11%	83,55%	2,55%	14,59%
16	0,3	0,3	0,4	88,02%	85,28%	2,79%	15,42%
17	0,4	0,2	0,4	80,07%	90,04%	1,78%	12,67%
18	0,5	0,1	0,4	71,81%	90,04%	1,27%	10,68%
19	0,1	0,6	0,3	86,32%	81,82%	2,35%	13,88%
20	0,2	0,5	0,3	86,64%	82,68%	2,44%	14,19%
21	0,3	0,4	0,3	87,76%	84,42%	2,70%	15,11%
22	0,35	0,35	0,3	88,34%	86,15%	2,89%	15,78%
23	0,4	0,3	0,3	86,38%	87,01%	2,51%	14,77%
24	0,5	0,2	0,3	77,01%	91,34%	1,57%	11,98%
25	0,6	0,1	0,3	70,56%	90,04%	1,21%	10,45%
26	0,1	0,7	0,2	86,23%	81,82%	2,34%	13,84%
27	0,2	0,6	0,2	86,40%	82,68%	2,39%	14,06%
28	0,3	0,5	0,2	86,94%	83,98%	2,53%	14,57%
29	0,4	0,4	0,2	88,35%	85,71%	2,88%	15,72%
30	0,5	0,3	0,2	83,14%	89,61%	2,09%	13,69%
31	0,6	0,2	0,2	74,88%	91,34%	1,44%	11,47%
32	0,7	0,1	0,2	69,61%	90,48%	1,18%	10,34%
33	0,1	0,8	0,1	86,18%	81,39%	2,32%	13,74%
34	0,2	0,7	0,1	86,28%	82,68%	2,37%	14,01%
35	0,3	0,6	0,1	86,58%	83,55%	2,45%	14,30%
36	0,4	0,5	0,1	87,62%	85,71%	2,71%	15,25%
37	0,45	0,45	0,1	88,39%	86,15%	2,90%	15,82%
38	0,5	0,4	0,1	87,66%	87,01%	2,76%	15,50%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
39	0,6	0,3	0,1	79,87%	92,64%	1,82%	12,97%
40	0,7	0,2	0,1	73,35%	90,91%	1,35%	11,08%
41	0,8	0,1	0,1	69,03%	90,48%	1,16%	10,24%

Lampiran 8 Nilai bobot, MLP *neuron* hidden 15 Lr 0,001 Persentase dataset 90:10

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,19%	83,55%	2,38%	14,10%
2	0,2	0,3	0,5	87,25%	83,98%	2,59%	14,74%
3	0,25	0,25	0,5	87,38%	84,42%	2,63%	14,89%
4	0,3	0,2	0,5	85,78%	86,15%	2,38%	14,32%
5	0,4	0,1	0,5	77,07%	86,58%	1,50%	11,38%
6	0,1	0,3	0,6	86,57%	83,12%	2,44%	14,23%
7	0,2	0,2	0,6	86,02%	83,98%	2,36%	14,09%
8	0,3	0,1	0,6	78,29%	85,71%	1,56%	11,58%
9	0,1	0,2	0,7	84,94%	83,12%	2,18%	13,45%
10	0,15	0,15	0,7	82,95%	83,12%	1,93%	12,65%
11	0,2	0,1	0,7	78,69%	83,12%	1,55%	11,33%
12	0,1	0,1	0,8	75,30%	76,62%	1,23%	9,72%
13	0,05	0,05	0,9	65,41%	69,26%	0,80%	7,44%
14	0,1	0,5	0,4	85,96%	83,55%	2,34%	13,99%
15	0,2	0,4	0,4	86,59%	83,98%	2,46%	14,38%
16	0,3	0,3	0,4	87,99%	84,42%	2,76%	15,25%
17	0,4	0,2	0,4	84,42%	86,15%	2,18%	13,70%
18	0,5	0,1	0,4	75,85%	89,18%	1,46%	11,42%
19	0,1	0,6	0,3	85,85%	83,55%	2,32%	13,94%
20	0,2	0,5	0,3	86,23%	83,98%	2,40%	14,19%
21	0,3	0,4	0,3	87,26%	84,85%	2,61%	14,89%
22	0,35	0,35	0,3	88,22%	84,85%	2,82%	15,47%
23	0,4	0,3	0,3	88,11%	84,42%	2,78%	15,33%
24	0,5	0,2	0,3	82,49%	85,28%	1,92%	12,81%
25	0,6	0,1	0,3	75,07%	89,18%	1,42%	11,24%
26	0,1	0,7	0,2	85,83%	83,55%	2,32%	13,92%
27	0,2	0,6	0,2	86,10%	83,98%	2,38%	14,13%
28	0,3	0,5	0,2	86,54%	84,85%	2,48%	14,50%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
29	0,4	0,4	0,2	88,33%	84,85%	2,85%	15,55%
30	0,5	0,3	0,2	87,17%	84,85%	2,60%	14,84%
31	0,6	0,2	0,2	80,59%	88,31%	1,80%	12,60%
32	0,7	0,1	0,2	74,42%	89,61%	1,39%	11,15%
33	0,1	0,8	0,1	85,73%	83,55%	2,30%	13,87%
34	0,2	0,7	0,1	86,02%	83,55%	2,35%	14,02%
35	0,3	0,6	0,1	86,30%	84,85%	2,43%	14,37%
36	0,4	0,5	0,1	87,30%	84,85%	2,62%	14,91%
37	0,45	0,45	0,1	88,38%	85,28%	2,87%	15,66%
38	0,5	0,4	0,1	88,70%	84,42%	2,93%	15,71%
39	0,6	0,3	0,1	85,65%	84,42%	2,32%	13,98%
40	0,7	0,2	0,1	79,10%	88,31%	1,67%	12,14%
41	0,8	0,1	0,1	73,94%	89,61%	1,36%	11,05%

Lampiran 9 Nilai bobot, MLP *neuron hidden* 20 Lr 0,1 Persentase dataset 70:30

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	88,31%	86,31%	2,89%	15,80%
2	0,2	0,3	0,5	89,28%	84,87%	3,10%	16,22%
3	0,25	0,25	0,5	89,23%	84,44%	3,07%	16,10%
4	0,3	0,2	0,5	87,43%	82,85%	2,59%	14,66%
5	0,4	0,1	0,5	77,79%	82,56%	1,48%	11,04%
6	0,1	0,3	0,6	88,84%	85,45%	3,00%	16,01%
7	0,2	0,2	0,6	88,27%	82,85%	2,77%	15,16%
8	0,3	0,1	0,6	80,47%	81,12%	1,65%	11,56%
9	0,1	0,2	0,7	87,58%	82,28%	2,61%	14,64%
10	0,15	0,15	0,7	86,49%	80,26%	2,34%	13,72%
11	0,2	0,1	0,7	82,64%	77,09%	1,76%	11,65%
12	0,1	0,1	0,8	82,18%	70,61%	1,58%	10,55%
13	0,05	0,05	0,9	55,70%	86,74%	0,78%	8,23%
14	0,1	0,5	0,4	88,02%	87,03%	2,85%	15,74%
15	0,2	0,4	0,4	88,59%	86,74%	2,98%	16,07%
16	0,3	0,3	0,4	89,65%	85,45%	3,23%	16,60%
17	0,4	0,2	0,4	84,74%	83,72%	2,17%	13,47%
18	0,5	0,1	0,4	75,61%	83,29%	1,36%	10,63%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
19	0,1	0,6	0,3	87,87%	87,03%	2,81%	15,65%
20	0,2	0,5	0,3	88,21%	87,75%	2,92%	15,99%
21	0,3	0,4	0,3	89,06%	87,18%	3,12%	16,48%
22	0,35	0,35	0,3	89,74%	85,59%	3,26%	16,70%
23	0,4	0,3	0,3	89,28%	84,73%	3,09%	16,19%
24	0,5	0,2	0,3	81,58%	85,01%	1,83%	12,46%
25	0,6	0,1	0,3	73,93%	83,00%	1,27%	10,25%
26	0,1	0,7	0,2	87,78%	87,32%	2,80%	15,64%
27	0,2	0,6	0,2	88,01%	87,61%	2,86%	15,84%
28	0,3	0,5	0,2	88,49%	87,90%	2,99%	16,21%
29	0,4	0,4	0,2	89,66%	87,03%	3,29%	16,91%
30	0,5	0,3	0,2	87,46%	85,01%	5,16%	15,05%
31	0,6	0,2	0,2	78,97%	84,58%	1,59%	11,61%
32	0,7	0,1	0,2	72,77%	83,57%	1,22%	10,10%
33	0,1	0,8	0,1	87,71%	87,32%	2,79%	15,60%
34	0,2	0,7	0,1	87,89%	87,61%	2,84%	15,76%
35	0,3	0,6	0,1	88,19%	88,04%	2,92%	16,03%
36	0,4	0,5	0,1	88,90%	87,61%	3,09%	16,45%
37	0,45	0,45	0,1	89,58%	87,03%	3,26%	16,85%
38	0,5	0,4	0,1	89,60%	85,73%	3,22%	16,62%
39	0,6	0,3	0,1	84,32%	85,59%	2,15%	13,57%
40	0,7	0,2	0,1	76,94%	84,87%	1,46%	11,13%
41	0,8	0,1	0,1	71,88%	83,14%	1,18%	9,89%

Lampiran 10 Nilai bobot, MLP *neuron hidden* 20 Lr 0,01 Persentase dataset 70:30

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	87,90%	88,18%	2,86%	15,87%
2	0,2	0,3	0,5	88,73%	88,47%	3,07%	16,48%
3	0,25	0,25	0,5	87,97%	87,18%	2,84%	15,73%
4	0,3	0,2	0,5	85,23%	87,18%	2,32%	14,23%
5	0,4	0,1	0,5	75,04%	85,16%	1,36%	10,74%
6	0,1	0,3	0,6	88,15%	87,75%	2,90%	15,96%
7	0,2	0,2	0,6	86,18%	86,74%	2,47%	14,64%
8	0,3	0,1	0,6	76,29%	85,88%	1,44%	11,11%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
9	0,1	0,2	0,7	85,46%	86,60%	2,34%	14,25%
10	0,15	0,15	0,7	82,59%	85,01%	1,93%	12,81%
11	0,2	0,1	0,7	77,25%	84,29%	1,47%	11,13%
12	0,1	0,1	0,8	74,66%	79,97%	1,26%	10,02%
13	0,05	0,05	0,9	66,04%	72,91%	0,86%	7,91%
14	0,1	0,5	0,4	87,75%	88,76%	2,84%	15,87%
15	0,2	0,4	0,4	88,34%	89,19%	2,99%	16,33%
16	0,3	0,3	0,4	89,05%	87,61%	3,13%	16,56%
17	0,4	0,2	0,4	82,75%	86,74%	1,99%	13,12%
18	0,5	0,1	0,4	73,99%	85,01%	1,30%	10,51%
19	0,1	0,6	0,3	87,68%	89,19%	2,84%	15,91%
20	0,2	0,5	0,3	88,03%	89,05%	2,91%	16,11%
21	0,3	0,4	0,3	89,12%	89,19%	3,20%	16,89%
22	0,35	0,35	0,3	89,62%	88,04%	3,31%	17,07%
23	0,4	0,3	0,3	88,67%	87,90%	3,04%	16,33%
24	0,5	0,2	0,3	79,78%	86,74%	1,70%	12,14%
25	0,6	0,1	0,3	73,18%	84,87%	1,26%	10,33%
26	0,1	0,7	0,2	87,65%	88,92%	2,79%	15,75%
27	0,2	0,6	0,2	87,88%	89,19%	2,88%	16,03%
28	0,3	0,5	0,2	88,52%	89,48%	3,05%	16,51%
29	0,4	0,4	0,2	89,85%	88,04%	3,38%	17,26%
30	0,5	0,3	0,2	86,86%	87,46%	2,61%	15,12%
31	0,6	0,2	0,2	77,72%	86,46%	1,54%	11,53%
32	0,7	0,1	0,2	72,41%	84,58%	1,22%	10,16%
33	0,1	0,8	0,1	87,62%	88,90%	2,81%	15,82%
34	0,2	0,7	0,1	87,81%	89,34%	2,87%	16,01%
35	0,3	0,6	0,1	88,19%	89,34%	2,96%	16,26%
36	0,4	0,5	0,1	89,28%	89,05%	3,24%	16,99%
37	0,45	0,45	0,1	89,91%	88,18%	3,41%	17,33%
38	0,5	0,4	0,1	89,81%	88,18%	3,37%	17,25%
39	0,6	0,3	0,1	84,28%	87,18%	2,19%	13,80%
40	0,7	0,2	0,1	76,41%	86,74%	1,46%	11,25%
41	0,8	0,1	0,1	71,90%	84,44%	1,20%	10,05%

Lampiran 11 Nilai bobot, MLP neuron hidden 20 Lr 0,001 Persentase dataset 70:30

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	88,22%	86,46%	2,88%	15,77%
2	0,2	0,3	0,5	89,00%	86,60%	3,08%	16,33%
3	0,25	0,25	0,5	88,83%	85,45%	3,00%	16,00%
4	0,3	0,2	0,5	86,88%	83,72%	2,51%	14,50%
5	0,4	0,1	0,5	77,15%	84,01%	1,46%	11,07%
6	0,1	0,3	0,6	88,41%	86,02%	2,91%	15,82%
7	0,2	0,2	0,6	87,21%	85,01%	2,61%	14,90%
8	0,3	0,1	0,6	78,22%	84,29%	1,54%	11,38%
9	0,1	0,2	0,7	86,32%	85,01%	2,45%	14,42%
10	0,15	0,15	0,7	83,80%	83,43%	2,03%	13,03%
11	0,2	0,1	0,7	78,30%	82,85%	1,52%	11,20%
12	0,1	0,1	0,8	74,74%	80,55%	1,27%	10,11%
13	0,05	0,05	0,9	64,55%	74,93%	0,84%	7,95%
14	0,1	0,5	0,4	88,16%	87,03%	2,88%	15,84%
15	0,2	0,4	0,4	88,67%	87,61%	3,02%	16,28%
16	0,3	0,3	0,4	89,52%	86,17%	3,21%	16,64%
17	0,4	0,2	0,4	85,57%	84,58%	2,31%	13,98%
18	0,5	0,1	0,4	76,01%	84,01%	1,39%	10,81%
19	0,1	0,6	0,3	88,12%	86,89%	2,87%	15,78%
20	0,2	0,5	0,3	88,43%	87,46%	2,96%	16,09%
21	0,3	0,4	0,3	89,36%	86,89%	3,19%	16,65%
22	0,35	0,35	0,3	89,84%	86,46%	3,32%	16,95%
23	0,4	0,3	0,3	89,32%	85,16%	3,12%	16,30%
24	0,5	0,2	0,3	83,78%	84,44%	2,06%	13,18%
25	0,6	0,1	0,3	75,17%	83,14%	1,33%	10,52%
26	0,1	0,7	0,2	88,12%	86,74%	2,86%	15,76%
27	0,2	0,6	0,2	88,32%	87,03%	2,92%	15,94%
28	0,3	0,5	0,2	88,90%	87,61%	3,09%	16,44%
29	0,4	0,4	0,2	90,00%	86,31%	3,37%	17,05%
30	0,5	0,3	0,2	88,46%	85,01%	2,89%	15,67%
31	0,6	0,2	0,2	82,04%	84,58%	1,86%	12,55%
32	0,7	0,1	0,2	74,44%	82,71%	1,29%	10,32%
33	0,1	0,8	0,1	88,11%	86,60%	2,85%	15,72%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
34	0,2	0,7	0,1	88,27%	87,03%	2,91%	15,91%
35	0,3	0,6	0,1	88,62%	87,18%	3,00%	16,17%
36	0,4	0,5	0,1	89,65%	87,18%	3,29%	16,93%
37	0,45	0,45	0,1	90,09%	86,17%	3,39%	17,10%
38	0,5	0,4	0,1	90,02%	85,16%	3,33%	16,85%
39	0,6	0,3	0,1	87,27%	85,01%	2,63%	14,94%
40	0,7	0,2	0,1	80,48%	84,44%	1,71%	12,03%
41	0,8	0,1	0,1	73,87%	82,85%	1,26%	10,22%

Lampiran 12 Nilai bobot, MLP *neuron hidden* 20 Lr 0,1 Persentase dataset 80:20

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,33%	85,50%	2,46%	14,50%
2	0,2	0,3	0,5	88,39%	83,98%	2,83%	15,43%
3	0,25	0,25	0,5	88,81%	82,25%	2,88%	15,40%
4	0,3	0,2	0,5	87,59%	80,95%	2,57%	14,41%
5	0,4	0,1	0,5	79,09%	81,17%	1,54%	11,18%
6	0,1	0,3	0,6	87,48%	83,77%	2,63%	14,83%
7	0,2	0,2	0,6	87,97%	80,52%	2,63%	14,55%
8	0,3	0,1	0,6	81,28%	78,35%	1,66%	11,40%
9	0,1	0,2	0,7	87,25%	77,27%	2,39%	13,59%
10	0,15	0,15	0,7	86,17%	75,76%	2,16%	12,81%
11	0,2	0,1	0,7	82,45%	73,16%	1,65%	11,00%
12	0,1	0,1	0,8	79,72%	69,26%	1,36%	9,70%
13	0,05	0,05	0,9	73,39%	53,68%	0,81%	6,59%
14	0,1	0,5	0,4	85,87%	86,58%	2,41%	14,44%
15	0,2	0,4	0,4	86,95%	86,36%	2,60%	14,97%
16	0,3	0,3	0,4	89,21%	83,55%	3,03%	15,91%
17	0,4	0,2	0,4	85,75%	82,03%	2,27%	13,64%
18	0,5	0,1	0,4	76,96%	83,55%	1,44%	10,96%
19	0,1	0,6	0,3	85,58%	86,58%	2,36%	14,30%
20	0,2	0,5	0,3	86,17%	87,01%	2,47%	14,66%
21	0,3	0,4	0,3	87,99%	86,15%	2,81%	15,56%
22	0,35	0,35	0,3	89,18%	84,85%	3,07%	16,13%
23	0,4	0,3	0,3	89,13%	83,98%	3,02%	15,93%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
24	0,5	0,2	0,3	83,04%	83,33%	1,94%	12,72%
25	0,6	0,1	0,3	75,46%	84,42%	1,36%	10,73%
26	0,1	0,7	0,2	85,44%	87,23%	2,35%	14,33%
27	0,2	0,6	0,2	85,83%	87,23%	2,42%	14,52%
28	0,3	0,5	0,2	86,79%	86,80%	2,58%	14,96%
29	0,4	0,4	0,2	89,05%	85,71%	3,06%	16,20%
30	0,5	0,3	0,2	87,97%	83,98%	2,74%	15,16%
31	0,6	0,2	0,2	80,40%	84,42%	1,70%	11,99%
32	0,7	0,1	0,2	74,28%	85,71%	1,32%	10,65%
33	0,1	0,8	0,1	85,31%	87,23%	2,33%	14,27%
34	0,2	0,7	0,1	85,62%	87,66%	2,40%	14,49%
35	0,3	0,6	0,1	86,14%	87,66%	2,48%	14,75%
36	0,4	0,5	0,1	87,84%	86,58%	2,79%	15,54%
37	0,45	0,45	0,1	88,93%	86,58%	3,06%	16,26%
38	0,5	0,4	0,1	89,43%	85,71%	3,17%	16,47%
39	0,6	0,3	0,1	85,62%	84,42%	2,31%	13,97%
40	0,7	0,2	0,1	78,36%	85,06%	1,56%	11,51%
41	0,8	0,1	0,1	73,46%	85,93%	1,28%	10,51%

Lampiran 13 Nilai bobot, MLP *neuron hidden* 20 Lr 0,01 Persentase dataset 80:20

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,90%	84,20%	2,52%	14,58%
2	0,2	0,3	0,5	88,23%	83,33%	2,78%	15,21%
3	0,25	0,25	0,5	88,26%	83,12%	2,78%	15,19%
4	0,3	0,2	0,5	86,37%	81,39%	2,35%	13,84%
5	0,4	0,1	0,5	77,85%	79,44%	1,42%	10,63%
6	0,1	0,3	0,6	87,32%	82,03%	2,54%	14,45%
7	0,2	0,2	0,6	86,57%	82,03%	2,40%	14,04%
8	0,3	0,1	0,6	78,75%	79,00%	1,48%	10,80%
9	0,1	0,2	0,7	85,26%	80,30%	2,15%	13,14%
10	0,15	0,15	0,7	82,58%	79,22%	1,80%	11,94%
11	0,2	0,1	0,7	78,60%	76,84%	1,43%	10,47%
12	0,1	0,1	0,8	74,09%	71,65%	1,10%	8,89%
13	0,05	0,05	0,9	64,09%	67,75%	0,75%	7,15%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
14	0,1	0,5	0,4	86,74%	84,42%	2,50%	14,53%
15	0,2	0,4	0,4	87,43%	83,77%	2,62%	14,81%
16	0,3	0,3	0,4	89,14%	84,20%	3,03%	15,98%
17	0,4	0,2	0,4	85,19%	81,82%	2,18%	13,35%
18	0,5	0,1	0,4	76,71%	80,95%	1,38%	10,57%
19	0,1	0,6	0,3	86,66%	84,20%	2,48%	14,45%
20	0,2	0,5	0,3	87,05%	84,20%	2,55%	14,67%
21	0,3	0,4	0,3	88,39%	84,42%	2,85%	15,51%
22	0,35	0,35	0,3	89,42%	84,20%	3,11%	16,18%
23	0,4	0,3	0,3	89,24%	83,12%	3,02%	15,85%
24	0,5	0,2	0,3	83,41%	80,95%	1,93%	12,50%
25	0,6	0,1	0,3	75,83%	81,82%	1,34%	10,48%
26	0,1	0,7	0,2	86,59%	84,20%	2,47%	14,42%
27	0,2	0,6	0,2	86,93%	83,98%	2,52%	14,56%
28	0,3	0,5	0,2	87,55%	84,20%	2,66%	14,95%
29	0,4	0,4	0,2	89,53%	84,63%	3,16%	16,35%
30	0,5	0,3	0,2	88,34%	82,90%	2,79%	15,21%
31	0,6	0,2	0,2	81,67%	81,82%	1,77%	12,02%
32	0,7	0,1	0,2	75,04%	83,12%	1,32%	10,48%
33	0,1	0,8	0,1	86,57%	83,98%	2,46%	14,37%
34	0,2	0,7	0,1	86,83%	84,20%	2,51%	14,54%
35	0,3	0,6	0,1	87,19%	85,06%	2,61%	14,89%
36	0,4	0,5	0,1	88,56%	83,98%	2,88%	15,54%
37	0,45	0,45	0,1	89,59%	84,20%	3,16%	16,31%
38	0,5	0,4	0,1	89,99%	83,33%	3,25%	16,47%
39	0,6	0,3	0,1	86,55%	81,60%	2,39%	13,96%
40	0,7	0,2	0,1	80,25%	81,82%	1,64%	11,59%
41	0,8	0,1	0,1	74,42%	82,90%	1,29%	10,33%

Lampiran 14 Nilai bobot, MLP *neuron hidden* 20 Lr 0,001 Persentase dataset 80:20

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	85,88%	87,23%	2,43%	14,55%
2	0,2	0,3	0,5	87,05%	86,80%	2,63%	15,11%
3	0,25	0,25	0,5	86,52%	86,15%	2,51%	14,71%

4	0,3	0,2	0,5	83,99%	84,85%	2,09%	13,32%
5	0,4	0,1	0,5	75,56%	85,06%	1,38%	10,84%
6	0,1	0,3	0,6	86,56%	85,71%	2,50%	14,65%
7	0,2	0,2	0,6	85,13%	84,20%	2,23%	13,71%
8	0,3	0,1	0,6	77,34%	82,47%	1,44%	10,91%
9	0,1	0,2	0,7	84,25%	80,52%	2,02%	12,75%
10	0,15	0,15	0,7	82,61%	80,09%	1,82%	12,06%
11	0,2	0,1	0,7	78,77%	79,00%	1,48%	10,80%
12	0,1	0,1	0,8	75,78%	72,73%	1,20%	9,33%
13	0,05	0,05	0,9	70,95%	61,47%	0,85%	7,21%
14	0,1	0,5	0,4	85,60%	87,45%	2,39%	14,44%
15	0,2	0,4	0,4	86,26%	88,53%	2,53%	14,96%
16	0,3	0,3	0,4	87,46%	87,23%	2,73%	15,42%
17	0,4	0,2	0,4	81,58%	85,50%	1,83%	12,52%
18	0,5	0,1	0,4	73,82%	87,23%	1,32%	10,74%
19	0,1	0,6	0,3	85,46%	87,45%	2,36%	14,38%
20	0,2	0,5	0,3	85,77%	88,31%	2,44%	14,67%
21	0,3	0,4	0,3	86,93%	88,10%	2,64%	15,26%
22	0,35	0,35	0,3	87,76%	87,88%	2,81%	15,72%
23	0,4	0,3	0,3	86,77%	86,80%	2,57%	14,95%
24	0,5	0,2	0,3	78,78%	87,01%	1,62%	11,88%
25	0,6	0,1	0,3	72,52%	87,88%	1,27%	10,56%
26	0,1	0,7	0,2	85,33%	87,23%	2,34%	14,28%
27	0,2	0,6	0,2	85,57%	88,31%	2,40%	14,57%
28	0,3	0,5	0,2	86,14%	89,18%	2,52%	15,00%
29	0,4	0,4	0,2	87,81%	87,88%	2,82%	15,75%
30	0,5	0,3	0,2	84,39%	87,66%	2,21%	13,92%
31	0,6	0,2	0,2	76,78%	88,53%	1,51%	11,56%
32	0,7	0,1	0,2	71,55%	87,66%	1,22%	10,36%
33	0,1	0,8	0,1	85,26%	87,45%	2,33%	14,28%
34	0,2	0,7	0,1	85,46%	88,10%	2,38%	14,48%
35	0,3	0,6	0,1	85,77%	88,96%	2,45%	14,78%
36	0,4	0,5	0,1	86,88%	88,10%	2,63%	15,23%
37	0,45	0,45	0,1	87,76%	87,88%	2,81%	15,72%
38	0,5	0,4	0,1	87,70%	87,23%	2,78%	15,57%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
39	0,6	0,3	0,1	81,30%	88,31%	1,86%	12,83%
40	0,7	0,2	0,1	75,16%	88,96%	1,42%	11,24%
41	0,8	0,1	0,1	70,98%	87,66%	1,20%	10,25%

Lampiran 15 Nilai bobot, MLP neuron hidden 20 Lr 0,1 Persentase dataset 90:10

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,51%	80,95%	2,36%	13,83%
2	0,2	0,3	0,5	87,69%	82,68%	2,64%	14,77%
3	0,25	0,25	0,5	87,82%	83,55%	2,69%	15,00%
4	0,3	0,2	0,5	85,81%	82,68%	2,30%	13,78%
5	0,4	0,1	0,5	77,41%	85,28%	1,50%	11,29%
6	0,1	0,3	0,6	86,92%	80,95%	2,44%	14,05%
7	0,2	0,2	0,6	86,45%	82,25%	2,39%	14,02%
8	0,3	0,1	0,6	78,77%	83,98%	1,57%	11,47%
9	0,1	0,2	0,7	85,23%	82,25%	2,20%	13,44%
10	0,15	0,15	0,7	83,55%	79,65%	1,91%	12,35%
11	0,2	0,1	0,7	79,48%	78,79%	1,52%	10,95%
12	0,1	0,1	0,8	77,03%	74,89%	1,30%	9,85%
13	0,05	0,05	0,9	66,53%	65,37%	0,78%	7,14%
14	0,1	0,5	0,4	86,26%	81,82%	2,34%	13,85%
15	0,2	0,4	0,4	86,94%	83,12%	2,50%	14,42%
16	0,3	0,3	0,4	88,44%	82,25%	2,79%	15,16%
17	0,4	0,2	0,4	84,23%	80,95%	2,03%	12,81%
18	0,5	0,1	0,4	76,13%	86,15%	1,43%	11,10%
19	0,1	0,6	0,3	86,15%	81,82%	2,33%	13,80%
20	0,2	0,5	0,3	86,53%	82,68%	2,42%	14,13%
21	0,3	0,4	0,3	87,77%	82,68%	2,65%	14,82%
22	0,35	0,35	0,3	88,68%	82,68%	2,86%	15,39%
23	0,4	0,3	0,3	88,27%	83,55%	2,79%	15,27%
24	0,5	0,2	0,3	82,22%	83,55%	1,86%	12,45%
25	0,6	0,1	0,3	75,15%	86,58%	1,38%	10,94%
26	0,1	0,7	0,2	86,11%	81,39%	2,31%	13,71%
27	0,2	0,6	0,2	86,39%	81,82%	2,37%	13,92%
28	0,3	0,5	0,2	86,95%	83,12%	2,50%	14,43%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
29	0,4	0,4	0,2	88,75%	82,68%	2,88%	15,43%
30	0,5	0,3	0,2	86,97%	81,39%	4,77%	14,14%
31	0,6	0,2	0,2	80,44%	85,71%	1,73%	12,19%
32	0,7	0,1	0,2	74,58%	87,88%	1,37%	10,97%
33	0,1	0,8	0,1	86,09%	81,39%	2,31%	13,70%
34	0,2	0,7	0,1	86,27%	81,82%	2,35%	13,86%
35	0,3	0,6	0,1	86,61%	82,25%	2,42%	14,10%
36	0,4	0,5	0,1	87,77%	82,25%	2,64%	14,74%
37	0,45	0,45	0,1	88,70%	82,68%	2,87%	15,40%
38	0,5	0,4	0,1	88,91%	82,68%	2,92%	15,54%
39	0,6	0,3	0,1	85,18%	81,82%	2,18%	13,35%
40	0,7	0,2	0,1	78,93%	86,58%	1,63%	11,87%
41	0,8	0,1	0,1	74,20%	87,88%	1,35%	10,89%

Lampiran 16 Nilai bobot, MLP neuron hidden 20 Lr 0,01 Persentase dataset 90:10

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	85,73%	84,42%	2,33%	14,02%
2	0,2	0,3	0,5	86,61%	85,71%	2,51%	14,68%
3	0,25	0,25	0,5	85,93%	85,71%	2,40%	14,33%
4	0,3	0,2	0,5	82,66%	85,71%	1,95%	12,93%
5	0,4	0,1	0,5	74,48%	88,74%	1,38%	11,06%
6	0,1	0,3	0,6	86,18%	84,42%	2,40%	14,24%
7	0,2	0,2	0,6	83,95%	85,28%	2,10%	13,37%
8	0,3	0,1	0,6	75,44%	88,74%	1,43%	11,27%
9	0,1	0,2	0,7	83,17%	83,98%	1,97%	12,86%
10	0,15	0,15	0,7	80,33%	83,98%	1,69%	11,91%
11	0,2	0,1	0,7	75,49%	87,88%	1,42%	11,17%
12	0,1	0,1	0,8	72,95%	80,09%	1,18%	9,71%
13	0,05	0,05	0,9	38,91%	93,51%	0,61%	7,56%
14	0,1	0,5	0,4	85,52%	84,42%	2,29%	13,92%
15	0,2	0,4	0,4	86,01%	85,71%	2,41%	14,37%
16	0,3	0,3	0,4	87,05%	86,15%	2,61%	15,00%
17	0,4	0,2	0,4	80,05%	90,04%	1,78%	12,67%
18	0,5	0,1	0,4	73,28%	88,74%	1,32%	10,81%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
19	0,1	0,6	0,3	85,41%	84,42%	2,28%	13,87%
20	0,2	0,5	0,3	85,68%	85,28%	2,34%	14,14%
21	0,3	0,4	0,3	86,73%	86,58%	2,56%	14,89%
22	0,35	0,35	0,3	87,54%	85,71%	2,70%	15,21%
23	0,4	0,3	0,3	86,52%	85,28%	2,48%	14,56%
24	0,5	0,2	0,3	78,15%	91,34%	1,65%	12,28%
25	0,6	0,1	0,3	72,44%	88,31%	1,27%	10,60%
26	0,1	0,7	0,2	85,34%	84,42%	2,27%	13,83%
27	0,2	0,6	0,2	85,55%	84,85%	2,31%	14,00%
28	0,3	0,5	0,2	86,04%	86,15%	2,43%	14,45%
29	0,4	0,4	0,2	87,80%	86,15%	2,77%	15,44%
30	0,5	0,3	0,2	83,97%	86,58%	2,13%	13,57%
31	0,6	0,2	0,2	76,44%	90,48%	2,99%	11,72%
32	0,7	0,1	0,2	71,96%	88,31%	1,25%	10,51%
33	0,1	0,8	0,1	85,31%	84,41%	2,26%	13,82%
34	0,2	0,7	0,1	85,50%	84,42%	2,29%	13,91%
35	0,3	0,6	0,1	85,72%	85,71%	2,36%	14,23%
36	0,4	0,5	0,1	86,76%	86,58%	2,57%	14,91%
37	0,45	0,45	0,1	87,81%	86,58%	2,78%	15,52%
38	0,5	0,4	0,1	87,87%	85,28%	2,76%	15,33%
39	0,6	0,3	0,1	80,72%	89,61%	1,83%	12,82%
40	0,7	0,2	0,1	75,39%	90,04%	1,45%	11,42%
41	0,8	0,1	0,1	71,55%	87,88%	1,23%	10,38%

Lampiran 17 Nilai bobot, MLP *neuron hidden* 20 Lr 0,001 Persentase dataset 90:10

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,53	82,25%	2,40%	14,06%
2	0,2	0,3	0,5	88,58%	81,82%	2,81%	15,16%
3	0,25	0,25	0,5	89,24%	81,39%	2,97%	15,53%
4	0,3	0,2	0,5	88,31%	79,65%	2,68%	14,61%
5	0,4	0,1	0,5	80,28%	83,12%	1,67%	11,78%
6	0,1	0,3	0,6	87,54%	81,39%	2,57%	14,46%
7	0,2	0,2	0,6	88,60%	79,22%	2,73%	14,71%
8	0,3	0,1	0,6	82,07%	79,22%	1,75%	11,77%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
9	0,1	0,2	0,7	87,73%	77,49%	2,49%	13,89%
10	0,15	0,15	0,7	86,78%	75,32%	2,25%	13,02%
11	0,2	0,1	0,7	83,45%	71,86%	1,72%	11,13%
12	0,1	0,1	0,8	80,73%	66,67%	1,38%	9,58%
13	0,05	0,05	0,9	74,50%	56,28%	0,88%	7,05%
14	0,1	0,5	0,4	85,95%	82,68%	2,32%	13,84%
15	0,2	0,4	0,4	87,18%	82,68%	2,54%	14,48%
16	0,3	0,3	0,4	89,57%	82,25%	3,09%	15,93%
17	0,4	0,2	0,4	86,50%	81,39%	2,37%	13,90%
18	0,5	0,1	0,4	78,73%	85,71%	1,59%	11,69%
19	0,1	0,6	0,3	85,69%	82,68%	2,28%	13,72%
20	0,2	0,5	0,3	86,38%	83,12%	2,40%	14,13%
21	0,3	0,4	0,3	88,18%	84,42%	2,80%	15,38%
22	0,35	0,35	0,3	89,51%	83,55%	3,11%	16,13%
23	0,4	0,3	0,3	89,69%	81,39%	3,09%	15,86%
24	0,5	0,2	0,3	83,68%	85,28%	2,06%	13,25%
25	0,6	0,1	0,3	77,63%	85,28%	1,51%	11,35%
26	0,1	0,7	0,2	85,54%	82,68%	2,25%	13,65%
27	0,2	0,6	0,2	85,96%	83,12%	2,33%	13,92%
28	0,3	0,5	0,2	87,00%	84,42%	2,55%	14,67%
29	0,4	0,4	0,2	89,37%	83,98%	3,09%	16,11%
30	0,5	0,3	0,2	88,62%	81,39%	2,81%	15,11%
31	0,6	0,2	0,2	81,19%	87,01%	1,83%	12,61%
32	0,7	0,1	0,2	76,86%	84,85%	1,45%	11,10%
33	0,1	0,8	0,1	85,46%	82,68%	2,24%	13,61%
34	0,2	0,7	0,1	85,79%	82,68%	2,29%	13,77%
35	0,3	0,6	0,1	86,32%	83,98%	2,41%	14,24%
36	0,4	0,5	0,1	88,03%	84,85%	2,78%	15,36%
37	0,45	0,45	0,1	89,23%	84,42%	3,07%	16,09%
38	0,5	0,4	0,1	89,96%	82,68%	3,22%	16,32%
39	0,6	0,3	0,1	86,38%	84,85%	2,45%	14,42%
40	0,7	0,2	0,1	79,68%	87,88%	1,71%	12,26%
41	0,8	0,1	0,1	76,32%	85,28%	1,43%	11,03%

Lampiran 18 Nilai bobot, MLP *neuron hidden* 25 Lr 0,1 Persentase dataset 70:30

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	88,42%	85,88%	2,91%	15,80%
2	0,2	0,3	0,5	90,05%	84,01%	3,30%	16,66%
3	0,25	0,25	0,5	90,78%	81,41%	3,45%	16,77%
4	0,3	0,2	0,5	90,24%	79,68%	3,20%	15,97%
5	0,4	0,1	0,5	83,94%	73,63%	1,82%	11,58%
6	0,1	0,3	0,6	88,89%	85,30%	3,01%	16,02%
7	0,2	0,2	0,6	89,54%	80,89%	3,04%	15,68%
8	0,3	0,1	0,6	84,75%	74,93%	1,95%	12,08%
9	0,1	0,2	0,7	87,63%	81,84%	2,60%	14,60%
10	0,15	0,15	0,7	86,97%	79,25%	2,40%	13,79%
11	0,2	0,1	0,7	84,58%	74,06%	1,91%	11,88%
12	0,1	0,1	0,8	80,27%	72,77%	1,47%	10,34%
13	0,05	0,05	0,9	72,66%	64,41%	0,94%	7,79%
14	0,1	0,5	0,4	88,12%	86,31%	2,85%	15,68%
15	0,2	0,4	0,4	89,19%	86,02%	3,11%	16,37%
16	0,3	0,3	0,4	91,26%	82,13%	3,66%	17,35%
17	0,4	0,2	0,4	89,77%	78,24%	3,01%	15,34%
18	0,5	0,1	0,4	83,10%	73,49%	1,73%	11,27%
19	0,1	0,6	0,3	87,96%	86,74%	2,83%	15,66%
20	0,2	0,5	0,3	88,62%	86,74%	2,98%	16,09%
21	0,3	0,4	0,3	90,35%	84,73%	3,43%	17,04%
22	0,35	0,35	0,3	91,45%	82,71%	3,77%	17,66%
23	0,4	0,3	0,3	91,82%	80,55%	3,84%	17,59%
24	0,5	0,2	0,3	88,64%	77,52%	2,69%	14,44%
25	0,6	0,1	0,3	82,34%	72,91%	1,64%	10,94%
26	0,1	0,7	0,2	87,88%	86,60%	2,80%	15,58%
27	0,2	0,6	0,2	88,32%	86,74%	2,91%	15,89%
28	0,3	0,5	0,2	89,46%	85,45%	3,17%	16,46%
29	0,4	0,4	0,2	91,52%	82,56%	3,79%	17,70%
30	0,5	0,3	0,2	91,52%	79,54%	3,66%	17,07%
31	0,6	0,2	0,2	87,28%	75,79%	2,36%	13,36%
32	0,7	0,1	0,2	81,69%	73,05%	1,59%	10,77%
33	0,1	0,8	0,1	87,83%	86,46%	2,79%	15,52%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
34	0,2	0,7	0,1	2,85%	86,46%	2,86%	15,73%
35	0,3	0,6	0,1	88,89%	86,46%	3,05%	16,23%
36	0,4	0,5	0,1	90,56%	84,29%	3,49%	17,14%
37	0,45	0,45	0,1	91,53%	82,71%	3,80%	17,74%
38	0,5	0,4	0,1	92,05%	80,98%	3,97%	17,93%
39	0,6	0,3	0,1	90,72%	78,10%	3,30%	16,06%
40	0,7	0,2	0,1	86,05%	75,22%	2,14%	12,67%
41	0,8	0,1	0,1	81,10%	73,05%	1,54%	10,60%

Lampiran 19 Nilai bobot, MLP *neuron hidden* 25 Lr 0,01 Persentase dataset 70:30

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	89,02%	84,44%	3,01%	15,95%
2	0,2	0,3	0,5	89,87%	84,01%	3,25%	16,51%
3	0,25	0,25	0,5	89,33%	84,15%	3,09%	16,12%
4	0,3	0,2	0,5	86,16%	84,01%	2,39%	14,17%
5	0,4	0,1	0,5	75,34%	83,00%	1,34%	10,54%
6	0,1	0,3	0,6	89,13%	83,86%	3,02%	15,92%
7	0,2	0,2	0,6	87,33%	83,72%	2,60%	14,75%
8	0,3	0,1	0,6	76,66%	84,15%	1,43%	10,98%
9	0,1	0,2	0,7	86,90%	83,43%	2,51%	14,46%
10	0,15	0,15	0,7	83,81%	83,00%	2,03%	12,97%
11	0,2	0,1	0,7	77,83%	82,13%	1,47%	10,99%
12	0,1	0,1	0,8	76,19%	80,69%	1,35%	10,43%
13	0,05	0,05	0,9	66,06%	74,93%	0,88%	8,13%
14	0,1	0,5	0,4	88,92%	84,29%	2,98%	15,86%
15	0,2	0,4	0,4	89,55%	84,73%	3,17%	16,39%
16	0,3	0,3	0,4	90,34%	84,29%	3,41%	16,95%
17	0,4	0,2	0,4	83,72%	84,01%	2,04%	13,09%
18	0,5	0,1	0,4	74,17%	83,29%	1,28%	10,33%
19	0,1	0,6	0,3	88,88%	84,44%	2,98%	15,85%
20	0,2	0,5	0,3	89,29%	84,44%	3,09%	16,15%
21	0,3	0,4	0,3	90,38%	84,29%	3,42%	16,98%
22	0,35	0,35	0,3	90,79%	83,86%	3,55%	17,26%
23	0,4	0,3	0,3	89,83%	83,86%	3,23%	16,45%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
24	0,5	0,2	0,3	81,20%	84,15%	1,77%	12,21%
25	0,6	0,1	0,3	73,02%	82,71%	1,22%	10,04%
26	0,1	0,7	0,2	2,96%	84,29%	2,97%	15,83%
27	0,2	0,6	0,2	89,14%	84,58%	3,05%	16,07%
28	0,3	0,5	0,2	89,96%	84,73%	3,30%	16,72%
29	0,4	0,4	0,2	91,05%	84,58%	3,68%	17,65%
30	0,5	0,3	0,2	87,84%	83,72%	2,71%	15,05%
31	0,6	0,2	0,2	79,35%	83,00%	1,59%	11,50%
32	0,7	0,1	0,2	72,10%	82,56%	1,18%	9,86%
33	0,1	0,8	0,1	88,88%	84,01%	2,96%	15,78%
34	0,2	0,7	0,1	89,08%	84,29%	3,02%	15,97%
35	0,3	0,6	0,1	89,65%	84,29%	3,19%	16,39%
36	0,4	0,5	0,1	90,69%	84,44%	3,54%	17,28%
37	0,45	0,45	0,1	91,21%	84,58%	3,75%	17,80%
38	0,5	0,4	0,1	90,92%	83,57%	3,59%	17,32%
39	0,6	0,3	0,1	85,31%	83,29%	2,24%	13,65%
40	0,7	0,2	0,1	77,85%	83,72%	1,50%	11,20%
41	0,8	0,1	0,1	71,49%	82,13%	1,15%	9,71%

Lampiran 20 Nilai bobot, MLP *neuron hidden* 25 Lr 0,001 Persentase dataset 70:30

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	88,12%	86,60%	2,86%	15,73%
2	0,2	0,3	0,5	88,95%	87,18%	3,09%	16,40%
3	0,25	0,25	0,5	88,77%	86,17%	3,01%	16,09%
4	0,3	0,2	0,5	86,13%	85,16%	2,42%	14,35%
5	0,4	0,1	0,5	76,37%	84,29%	1,42%	10,93%
6	0,1	0,3	0,6	88,57%	85,73%	2,94%	15,87%
7	0,2	0,2	0,6	88,06%	85,01%	2,79%	15,41%
8	0,3	0,1	0,6	78,82%	83,00%	1,55%	11,36%
9	0,1	0,2	0,7	88,41%	83,57%	2,83%	15,38%
10	0,15	0,15	0,7	85,95%	82,28%	2,31%	13,79%
11	0,2	0,1	0,7	81,17%	78,96%	1,66%	11,46%
12	0,1	0,1	0,8	80,26%	71,90%	1,45%	10,21%
13	0,05	0,05	0,9	73,84%	60,23%	0,92%	7,45%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
14	0,1	0,5	0,4	87,83%	87,03%	2,81%	15,63%
15	0,2	0,4	0,4	88,29%	88,04%	2,94%	16,10%
16	0,3	0,3	0,4	89,06%	87,18%	3,12%	16,48%
17	0,4	0,2	0,4	83,03%	86,89%	2,02%	13,25%
18	0,5	0,1	0,4	74,57%	85,30%	1,33%	10,66%
19	0,1	0,6	0,3	87,68%	87,61%	2,79%	15,63%
20	0,2	0,5	0,3	87,92%	88,04%	2,86%	15,85%
21	0,3	0,4	0,3	88,64%	89,19%	3,07%	16,55%
22	0,35	0,35	0,3	89,01%	87,90%	3,13%	16,58%
23	0,4	0,3	0,3	87,97%	87,75%	2,86%	15,83%
24	0,5	0,2	0,3	79,97%	87,32%	1,73%	12,28%
25	0,6	0,1	0,3	73,15%	85,30%	1,26%	10,38%
26	0,1	0,7	0,2	87,59%	87,75%	2,77%	15,60%
27	0,2	0,6	0,2	87,77%	88,04%	2,82%	15,76%
28	0,3	0,5	0,2	88,11%	88,76%	2,92%	16,10%
29	0,4	0,4	0,2	88,89%	88,62%	3,12%	16,62%
30	0,5	0,3	0,2	85,36%	88,90%	2,39%	14,57%
31	0,6	0,2	0,2	77,60%	87,90%	1,56%	11,69%
32	0,7	0,1	0,2	71,90%	85,73%	1,21%	10,20%
33	0,1	0,8	0,1	87,53%	87,90%	2,76%	15,59%
34	0,2	0,7	0,1	87,67%	87,90%	2,80%	15,68%
35	0,3	0,6	0,1	87,87%	88,47%	2,86%	15,90%
36	0,4	0,5	0,1	88,42%	88,90%	3,00%	16,34%
37	0,45	0,45	0,1	88,77%	88,76%	3,09%	16,56%
38	0,5	0,4	0,1	88,29%	88,62%	2,96%	16,20%
39	0,6	0,3	0,1	82,29%	88,62%	1,98%	13,23%
40	0,7	0,2	0,1	75,92%	88,33%	1,46%	11,34%
41	0,8	0,1	0,1	70,95%	86,02%	1,18%	10,07%

Lampiran 21 Nilai bobot, MLP *neuron hidden* 25 Lr 0,1 Persentase dataset 80:20

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	85,99%	86,80%	2,43%	14,54%
2	0,2	0,3	0,5	86,95%	86,80%	2,61%	15,05%
3	0,25	0,25	0,5	86,64%	86,15%	2,53%	14,77%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
4	0,3	0,2	0,5	84,07%	84,85%	2,10%	13,35%
5	0,4	0,1	0,5	74,36%	84,63%	1,31%	10,53%
6	0,1	0,3	0,6	86,53%	85,71%	2,50%	14,64%
7	0,2	0,2	0,6	85,49%	84,42%	2,29%	13,90%
8	0,3	0,1	0,6	76,77%	81,17%	1,39%	10,61%
9	0,1	0,2	0,7	85,20%	82,25%	2,19%	13,42%
10	0,15	0,15	0,7	83,23%	82,03%	1,93%	12,59%
11	0,2	0,1	0,7	78,70%	79,00%	1,47%	10,78%
12	0,1	0,1	0,8	77,32%	72,51%	1,27%	9,61%
13	0,05	0,05	0,9	71,07%	61,69%	0,85%	7,25%
14	0,1	0,5	0,4	85,76%	87,23%	2,41%	14,49%
15	0,2	0,4	0,4	86,26%	88,31%	2,52%	14,93%
16	0,3	0,3	0,4	87,17%	87,66%	2,68%	15,32%
17	0,4	0,2	0,4	81,38%	85,93%	1,82%	12,52%
18	0,5	0,1	0,4	72,62%	87,01%	1,26%	10,48%
19	0,1	0,6	0,3	85,62%	87,23%	2,38%	14,42%
20	0,2	0,5	0,3	85,91%	88,31%	2,46%	14,74%
21	0,3	0,4	0,3	86,80%	88,74%	2,63%	15,29%
22	0,35	0,35	0,3	87,39%	87,88%	2,73%	15,49%
23	0,4	0,3	0,3	86,11%	86,15%	2,44%	14,49%
24	0,5	0,2	0,3	78,70%	85,93%	1,60%	11,71%
25	0,6	0,1	0,3	71,30%	87,66%	1,21%	10,31%
26	0,1	0,7	0,2	85,54%	87,01%	2,36%	14,34%
27	0,2	0,6	0,2	85,73%	88,10%	2,42%	14,62%
28	0,3	0,5	0,2	86,12%	88,53%	2,50%	14,89%
29	0,4	0,4	0,2	87,46%	87,88%	2,75%	15,53%
30	0,5	0,3	0,2	83,97%	87,45%	2,15%	13,71%
31	0,6	0,2	0,2	76,37%	87,45%	1,47%	11,32%
32	0,7	0,1	0,2	70,24%	87,88%	1,17%	10,15%
33	0,1	0,8	0,1	85,47%	87,01%	2,35%	14,31%
34	0,2	0,7	0,1	85,60%	88,10%	2,40%	14,55%
35	0,3	0,6	0,1	85,84%	88,53%	2,45%	14,74%
36	0,4	0,5	0,1	86,71%	88,53%	2,61%	15,21%
37	0,45	0,45	0,1	87,53%	87,66%	2,75%	15,54%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
38	0,5	0,4	0,1	86,77%	87,01%	2,58%	14,98%
39	0,6	0,3	0,1	81,45%	87,66%	1,87%	12,79%
40	0,7	0,2	0,1	74,71%	87,45%	1,37%	10,95%
41	0,8	0,1	0,1	69,40%	88,31%	1,15%	10,06%

Lampiran 22 Nilai bobot, MLP *neuron hidden* 25 Lr 0,01 Persentase dataset 80:20

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,77%	84,63%	2,51%	14,59%
2	0,2	0,3	0,5	88,53%	83,77%	2,86%	15,48%
3	0,25	0,25	0,5	88,60%	81,17%	2,80%	15,06%
4	0,3	0,2	0,5	87,08%	80,09%	2,44%	13,98%
5	0,4	0,1	0,5	78,50%	80,30%	1,48%	10,91%
6	0,1	0,3	0,6	87,23%	83,98%	2,58%	14,73%
7	0,2	0,2	0,6	86,85%	80,52%	2,41%	13,93%
8	0,3	0,1	0,6	79,64%	80,52%	1,57%	11,24%
9	0,1	0,2	0,7	84,97%	81,82%	2,15%	13,25%
10	0,15	0,15	0,7	83,44%	79,44%	1,90%	12,28%
11	0,2	0,1	0,7	80,01%	78,79%	1,56%	11,10%
12	0,1	0,1	0,8	75,31%	74,89%	1,21%	9,51%
13	0,05	0,05	0,9	66,64%	67,53%	0,81%	7,39%
14	0,1	0,5	0,4	86,53%	84,63%	2,47%	14,46%
15	0,2	0,4	0,4	87,58%	84,20%	2,66%	14,97%
16	0,3	0,3	0,4	89,60%	81,60%	3,07%	15,83%
17	0,4	0,2	0,4	86,09%	79,44%	2,25%	13,38%
18	0,5	0,1	0,4	77,35%	81,17%	1,42%	10,74%
19	0,1	0,6	0,3	86,39%	84,63%	2,44%	14,38%
20	0,2	0,5	0,3	87,02%	85,28%	2,58%	14,83%
21	0,3	0,4	0,3	88,93%	83,98%	2,97%	15,80%
22	0,35	0,35	0,3	90,00%	82,47%	3,22%	16,30%
23	0,4	0,3	0,3	90,08%	81,17%	3,20%	16,12%
24	0,5	0,2	0,3	83,98%	79,65%	1,96%	12,51%
25	0,6	0,1	0,3	76,32%	82,03%	1,37%	10,62%
26	0,1	0,7	0,2	86,32%	84,85%	2,44%	14,39%
27	0,2	0,6	0,2	86,75%	85,50%	2,53%	14,72%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
28	0,3	0,5	0,2	87,92%	85,06%	2,76%	15,32%
29	0,4	0,4	0,2	90,14%	83,33%	3,30%	16,58%
30	0,5	0,3	0,2	89,62%	80,74%	3,05%	15,69%
31	0,6	0,2	0,2	81,80%	81,82%	1,78%	12,06%
32	0,7	0,1	0,2	75,62%	82,03%	1,34%	10,47%
33	0,1	0,8	0,1	86,28%	84,63%	2,42%	14,33%
34	0,2	0,7	0,1	86,62%	85,28%	2,50%	14,61%
35	0,3	0,6	0,1	87,32%	85,71%	2,65%	15,08%
36	0,4	0,5	0,1	89,32%	84,63%	3,10%	16,20%
37	0,45	0,45	0,1	90,17%	83,33%	3,31%	16,61%
38	0,5	0,4	0,1	90,67%	82,25%	3,44%	16,82%
39	0,6	0,3	0,1	88,29%	79,87%	2,68%	14,63%
40	0,7	0,2	0,1	80,22%	82,03%	1,64%	11,61%
41	0,8	0,1	0,1	75,20%	82,03%	1,31%	10,38%

Lampiran 23 Nilai bobot, MLP *neuron hidden* 25 Lr 0,001 Persentase dataset 80:20

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,00%	86,58%	2,43%	14,50%
2	0,2	0,3	0,5	86,67%	86,58%	2,55%	14,86%
3	0,25	0,25	0,5	85,56%	87,66%	2,39%	14,46%
4	0,3	0,2	0,5	81,93%	87,66%	1,91%	12,95%
5	0,4	0,1	0,5	73,87%	86,36%	1,31%	10,64%
6	0,1	0,3	0,6	86,01%	85,71%	2,41%	14,37%
7	0,2	0,2	0,6	82,81%	87,23%	2,00%	13,21%
8	0,3	0,1	0,6	73,83%	87,23%	1,32%	10,74%
9	0,1	0,2	0,7	81,70%	86,36%	1,86%	12,69%
10	0,15	0,15	0,7	78,12%	86,58%	1,57%	11,65%
11	0,2	0,1	0,7	72,74%	87,23%	1,27%	10,52%
12	0,1	0,1	0,8	69,88%	84,42%	1,11%	9,70%
13	0,05	0,05	0,9	57,91%	80,74%	0,76%	7,86%
14	0,1	0,5	0,4	85,94%	86,80%	2,43%	14,51%
15	0,2	0,4	0,4	86,49%	87,23%	2,53%	14,87%
16	0,3	0,3	0,4	87,30%	87,88%	2,71%	15,44%
17	0,4	0,2	0,4	80,28%	86,15%	1,73%	12,20%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
18	0,5	0,1	0,4	73,31%	87,23%	1,30%	10,63%
19	0,1	0,6	0,3	85,91%	86,36%	2,41%	14,42%
20	0,2	0,5	0,3	86,30%	87,23%	2,50%	14,76%
21	0,3	0,4	0,3	87,58%	87,45%	2,76%	15,53%
22	0,35	0,35	0,3	88,19%	87,45%	2,90%	15,92%
23	0,4	0,3	0,3	86,49%	87,01%	2,53%	14,83%
24	0,5	0,2	0,3	78,87%	86,15%	1,61%	11,79%
25	0,6	0,1	0,3	72,97%	87,23%	1,28%	10,57%
26	0,1	0,7	0,2	85,94%	85,93%	2,40%	14,37%
27	0,2	0,6	0,2	86,17%	87,01%	2,47%	14,66%
28	0,3	0,5	0,2	86,87%	87,66%	2,62%	15,15%
29	0,4	0,4	0,2	88,69%	87,66%	3,03%	16,29%
30	0,5	0,3	0,2	84,51%	87,01%	2,21%	13,87%
31	0,6	0,2	0,2	77,72%	87,23%	1,55%	11,63%
32	0,7	0,1	0,2	72,64%	86,80%	1,26%	10,45%
33	0,1	0,8	0,1	85,96%	85,50%	2,39%	14,31%
34	0,2	0,7	0,1	86,15%	86,36%	2,45%	14,54%
35	0,3	0,6	0,1	86,53%	87,45%	2,55%	14,92%
36	0,4	0,5	0,1	88,04%	87,45%	2,86%	15,82%
37	0,45	0,45	0,1	88,92%	87,66%	3,09%	16,46%
38	0,5	0,4	0,1	88,50%	86,36%	2,94%	15,93%
39	0,6	0,3	0,1	82,51%	86,58%	1,95%	13,00%
40	0,7	0,2	0,1	76,81%	87,66%	1,50%	11,46%
41	0,8	0,1	0,1	72,36%	86,15%	1,24%	10,32%

Lampiran 24 Nilai bobot, MLP *neuron hidden* 25 Lr 0,1 Persentase dataset 90:10

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	87,20%	82,25%	2,53%	14,42%
2	0,2	0,3	0,5	89,58%	81,39%	3,06%	15,78%
3	0,25	0,25	0,5	90,33%	79,65%	3,22%	16,03%
4	0,3	0,2	0,5	89,32%	76,19%	2,81%	14,62%
5	0,4	0,1	0,5	82,46%	74,46%	1,68%	11,20%
6	0,1	0,3	0,6	88,07%	82,25%	2,71%	14,92%
7	0,2	0,2	0,6	89,23%	77,06%	2,81%	14,72%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
8	0,3	0,1	0,6	83,98%	72,73%	1,81%	11,46%
9	0,1	0,2	0,7	87,66%	77,49%	2,47%	13,85%
10	0,15	0,15	0,7	86,79%	75,32%	2,25%	13,03%
11	0,2	0,1	0,7	83,98%	72,73%	1,80%	11,44%
12	0,1	0,1	0,8	79,54%	67,53%	1,31%	9,42%
13	0,05	0,05	0,9	70,78%	58,44%	0,80%	6,84%
14	0,1	0,5	0,4	86,79%	82,25%	2,45%	14,19%
15	0,2	0,4	0,4	88,16%	82,68%	2,74%	15,05%
16	0,3	0,3	0,4	90,84%	80,09%	3,42%	16,54%
17	0,4	0,2	0,4	88,26%	75,32%	2,53%	13,80%
18	0,5	0,1	0,4	81,26%	75,76%	1,60%	11,02%
19	0,1	0,6	0,3	86,58%	81,82%	2,40%	14,01%
20	0,2	0,5	0,3	87,30%	82,25%	2,55%	14,47%
21	0,3	0,4	0,3	89,69%	81,82%	3,11%	15,95%
22	0,35	0,35	0,3	90,98%	80,52%	3,48%	16,75%
23	0,4	0,3	0,3	91,14%	78,35%	3,46%	16,46%
24	0,5	0,2	0,3	86,47%	75,76%	2,21%	12,94%
25	0,6	0,1	0,3	80,38%	76,19%	1,54%	10,84%
26	0,1	0,7	0,2	86,48%	81,82%	2,38%	13,96%
27	0,2	0,6	0,2	86,93%	81,82%	2,46%	14,19%
28	0,3	0,5	0,2	88,29%	81,39%	2,73%	14,90%
29	0,4	0,4	0,2	90,98%	80,95%	3,50%	16,84%
30	0,5	0,3	0,2	90,45%	77,06%	3,17%	15,62%
31	0,6	0,2	0,2	84,63%	77,49%	1,99%	12,43%
32	0,7	0,1	0,2	79,72%	75,76%	1,48%	10,60%
33	0,1	0,8	0,1	86,43%	81,39%	2,36%	13,87%
34	0,2	0,7	0,1	86,72%	81,82%	2,43%	14,09%
35	0,3	0,6	0,1	87,47%	81,39%	2,55%	14,42%
36	0,4	0,5	0,1	89,84%	80,95%	3,12%	15,89%
37	0,45	0,45	0,1	90,90%	80,95%	3,48%	16,77%
38	0,5	0,4	0,1	91,41%	78,79%	3,58%	16,80%
39	0,6	0,3	0,1	88,96%	75,76%	2,70%	14,31%
40	0,7	0,2	0,1	83,12%	76,62%	1,80%	11,74%
41	0,8	0,1	0,1	79,21%	76,62%	1,46%	10,59%

Lampiran 25 Nilai bobot, MLP *neuron hidden* 25 Lr 0,01 Persentase dataset 90:10

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	87,18%	82,68%	2,54%	14,48%
2	0,2	0,3	0,5	89,07%	83,98%	3,01%	15,89%
3	0,25	0,25	0,5	88,91%	83,12%	2,94%	15,62%
4	0,3	0,2	0,5	86,89%	81,82%	2,45%	14,17%
5	0,4	0,1	0,5	78,79%	83,12%	1,55%	11,36%
6	0,1	0,3	0,6	87,87%	81,82%	2,65%	14,73%
7	0,2	0,2	0,6	87,36%	80,95%	2,52%	14,28%
8	0,3	0,1	0,6	79,63%	82,68%	1,61%	11,53%
9	0,1	0,2	0,7	86,06%	79,65%	2,25%	13,40%
10	0,15	0,15	0,7	83,76%	80,09%	1,95%	12,50%
11	0,2	0,1	0,7	79,73%	80,52%	1,57%	11,26%
12	0,1	0,1	0,8	75,67%	73,59%	1,20%	9,41%
13	0,05	0,05	0,9	66,81%	68,83%	0,83%	7,55%
14	0,1	0,5	0,4	86,86%	82,68%	2,47%	14,30%
15	0,2	0,4	0,4	88,02%	83,12%	2,72%	15,04%
16	0,3	0,3	0,4	89,86%	81,82%	3,16%	16,08%
17	0,4	0,2	0,4	85,37%	82,25%	2,22%	13,50%
18	0,5	0,1	0,4	77,94%	83,55%	1,50%	11,20%
19	0,1	0,6	0,3	86,71%	81,82%	2,42%	14,08%
20	0,2	0,5	0,3	87,36%	83,12%	2,58%	14,65%
21	0,3	0,4	0,3	89,32%	84,42%	3,09%	16,16%
22	0,35	0,35	0,3	90,23%	81,82%	3,27%	16,37%
23	0,4	0,3	0,3	89,86%	82,25%	3,17%	16,16%
24	0,5	0,2	0,3	83,39%	83,55%	1,99%	12,88%
25	0,6	0,1	0,3	77,24%	83,55%	1,46%	11,03%
26	0,1	0,7	0,2	86,64%	81,39%	2,40%	13,97%
27	0,2	0,6	0,2	87,09%	82,68%	2,52%	14,43%
28	0,3	0,5	0,2	88,26%	83,12%	2,78%	15,19%
29	0,4	0,4	0,2	90,38%	82,25%	3,34%	16,58%
30	0,5	0,3	0,2	88,54%	81,39%	2,79%	15,06%
31	0,6	0,2	0,2	81,66%	84,42%	1,82%	12,39%
32	0,7	0,1	0,2	76,83%	83,98%	1,44%	10,99%
33	0,1	0,8	0,1	86,58%	80,95%	2,38%	13,87%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
34	0,2	0,7	0,1	86,93%	82,25%	2,48%	14,27%
35	0,3	0,6	0,1	87,59%	82,68%	2,62%	14,71%
36	0,4	0,5	0,1	89,54%	83,98%	3,14%	16,24%
37	0,45	0,45	0,1	90,50%	83,12%	3,41%	16,85%
38	0,5	0,4	0,1	90,74%	81,82%	3,45%	16,80%
39	0,6	0,3	0,1	86,49%	83,12%	2,42%	14,19%
40	0,7	0,2	0,1	80,45%	85,28%	1,72%	12,13%
41	0,8	0,1	0,1	76,39%	84,42%	1,42%	10,94%

Lampiran 26 Nilai bobot, MLP *neuron hidden* 25 Lr 0,001 Persentase dataset 90:10

No	α	β	γ	Akursi	Recall	Presisi	G-mean
1	0,1	0,4	0,5	86,84%	83,12%	2,48%	14,37%
2	0,2	0,3	0,5	88,47%	83,98%	2,85%	15,48%
3	0,25	0,25	0,5	88,29%	81,82%	2,74%	14,98%
4	0,3	0,2	0,5	85,92%	80,95%	4,41%	13,55%
5	0,4	0,1	0,5	77,55%	83,12%	1,47%	11,04%
6	0,1	0,3	0,6	87,66%	81,82%	2,61%	14,60%
7	0,2	0,2	0,6	86,99%	80,52%	2,44%	14,01%
8	0,3	0,1	0,6	79,57%	81,39%	1,58%	11,33%
9	0,1	0,2	0,7	86,54%	77,49%	2,27%	13,27%
10	0,15	0,15	0,7	84,71%	77,49%	2,00%	12,46%
11	0,2	0,1	0,7	81,24%	75,76%	1,60%	11,02%
12	0,1	0,1	0,8	78,36%	69,70%	1,28%	9,45%
13	0,05	0,05	0,9	72,78%	60,61%	0,89%	7,35%
14	0,1	0,5	0,4	86,45%	82,68%	2,40%	14,09%
15	0,2	0,4	0,4	87,35%	83,98%	2,61%	14,80%
16	0,3	0,3	0,4	88,99%	83,55%	2,97%	15,76%
17	0,4	0,2	0,4	83,42%	83,55%	1,99%	12,89%
18	0,5	0,1	0,4	75,89%	83,55%	1,37%	10,72%
19	0,1	0,6	0,3	86,20%	81,82%	2,34%	13,82%
20	0,2	0,5	0,3	86,73%	83,55%	2,47%	14,38%
21	0,3	0,4	0,3	88,31%	84,85%	2,84%	15,53%
22	0,35	0,35	0,3	89,35%	84,85%	3,12%	16,26%
23	0,4	0,3	0,3	88,36%	82,25%	2,77%	15,10%

No	α	β	γ	Akursi	Recall	Presisi	G-mean
24	0,5	0,2	0,3	80,81%	84,85%	1,75%	12,18%
25	0,6	0,1	0,3	74,67%	84,85%	1,33%	10,62%
26	0,1	0,7	0,2	86,10%	81,82%	2,32%	13,77%
27	0,2	0,6	0,2	86,44%	83,12%	2,41%	14,16%
28	0,3	0,5	0,2	87,24%	85,28%	2,62%	14,96%
29	0,4	0,4	0,2	89,39%	84,85%	3,12%	16,28%
30	0,5	0,3	0,2	85,89%	84,42%	2,35%	14,10%
31	0,6	0,2	0,2	78,83%	85,28%	1,59%	11,66%
32	0,7	0,1	0,2	73,68%	84,42%	1,27%	10,37%
33	0,1	0,8	0,1	86,03%	81,82%	2,31%	13,74%
34	0,2	0,7	0,1	86,28%	82,68%	2,37%	14,00%
35	0,3	0,6	0,1	86,70%	84,42%	2,50%	14,51%
36	0,4	0,5	0,1	88,24%	85,28%	2,84%	15,56%
37	0,45	0,45	0,1	89,37%	84,85%	3,12%	16,27%
38	0,5	0,4	0,1	89,16%	83,55%	3,02%	15,88%
39	0,6	0,3	0,1	83,30%	83,98%	1,99%	12,91%
40	0,7	0,2	0,1	77,38%	85,71%	1,50%	11,34%
41	0,8	0,1	0,1	73,02%	84,42%	1,24%	10,24%

BIOGRAFI PENULIS



Arif Mudi Priyatno lahir di Pasir Emas pada 23 Mei 1995. Penulis menempuh Pendidikan dari Sekolah Dasar Negeri 012 Pasir Emas, Sekolah Menengah Pertama Negeri 04 Singingi, Sekolah Menengah Atas Negeri 2 Singingi. Pada tahun 2013 penulis melanjutkan Pendidikan Strata-1 jurusan Teknik Informatika di Universitas Islam Negeri Sultan Syarif Kasim Riau. Kemudian pada tahun 2018 penulis diterima di Magister Teknik Informatika Institut Teknologi Sepuluh Nopember. Penulis dapat dihubungi melalui email: arifmudi11@gmail.com.