

Detty Purnamasari
Jonathan Henharta
Yoga Perdana Sasmita
Fuji Ihsani
I Wayan Simri Wicaksana



**Get Easy Using
WEKA**



Dapur Buku

GET EASY USING WEKA

Oleh :

Detty Purnamasari

Jonathan Henharta

Yoga Perdana Sasmita

Fuji Ihsani

I Wayan Simri Wicaksana

Penyunting: Lintang Yuniar Banowosari

Desain Sampul: Yoga Perdana Sasmita

Diterbitkan pertama kali oleh Dapur Buku



Jln. Kerja Bakti RT 001/02 No.9

Kel. Makasar, Kec. Makasar

Jakarta Timur 13570, Indonesia

Telp: 021-8098208

e-mail: info@DapurBuku.com

www.DapurBuku.com

Cetakan Pertama, Maret 2013

ISBN: 978-602-7749-94-8

Kata Pengantar

Segala puji bagi Tuhan Yang Maha Esa yang telah memberikan kemudahan bagi penyusun dalam menyelesaikan buku yang berjudul “GET EASY USING WEKA”. Buku ini disusun agar dapat membantu pembaca memperluas pengetahuan tentang aplikasi WEKA untuk kepentingan Data Mining. Buku disajikan berdasarkan pengamatan dan percobaan langsung terhadap aplikasi WEKA serta melihat dari berbagai sumber mengenai materi terkait dimana terdapat juga satu sumber buku manual WEKA berbahasa Inggris.

Buku ini memuat banyak penjelasan mengenai fitur-fitur yang terdapat di dalam aplikasi WEKA serta cara penggunaan aplikasi. Informasi detail mengenai aplikasi dan *tools* terkait dipaparkan dengan cukup jelas bagi pembaca.

Penyusun juga mengucapkan terimakasih kepada semua pihak yang telah membantu dalam proses penyusunan buku ini, dan saran yang membangun untuk kelancaran penyusunan buku ini.

Semoga buku ini dapat memberikan wawasan yang lebih luas dan manfaat bagi pembaca. Pada buku ini mungkin terdapat kesalahan dalam tata bahasa maupun dalam melakukan penjelasan mengenai aplikasi WEKA, maka dari itu diharapkan pembaca dapat memaklumi kekurangan dari buku ini dan penyusun sangat terbuka terhadap saran dan kritik dari pembaca.

Penyusun
Maret 2013

Daftar Isi

Kata Pengantar.....	i
Daftar Isi.....	ii
Bab I. Pendahuluan.....	1
1.1. Artificial Intelligence.....	1
1.2. Machine Learning.....	4
1.3. Data Mining.....	9
Bab II. Tentang Weka.....	17
Bab III. Instalasi.....	19
3.1. System Requirement.....	19
3.2. Cara Instalasi.....	27
Bab IV. Command Line.....	27
4.1. Pendahuluan.....	27
4.2. Konsep Dasar.....	28
Bab V. GUI Weka.....	38
5.1. Menjalankan WEKA.....	38
5.2. Explorer.....	42
5.3. Simple CLI.....	101
5.4. Experimenter.....	106
5.5. Knowledge Flow.....	148
5.6. Arff Viewer.....	160
5.7. Bayes Network Editor.....	165
Bab VI. Data.....	184
Bab VII. Glosarium.....	217

Bab VIII. Lampiran.....	221
8.1. Daftar Tabel dan Gambar.....	221
Daftar Pustaka.....	230



Bab I

Pendahuluan

Pada bagian ini dijelaskan mengenai dasar teori yang berhubungan dengan *Weka Machine Learning*.

1.1.

Artificial Intelligence (Kecerdasan Buatan)

Pengertian

Artificial Intelligence (AI) atau dalam Bahasa Indonesia berarti Kecerdasan Buatan merupakan teknologi yang sudah sangat populer saat ini.

Berikut ini adalah pengertian *Artificial Intelligence* menurut beberapa ahli.

Menurut H.A.Simon (1987)

”Kecerdasan buatan (*artificial intelligence*) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan hal yang dalam pandangan manusia adalah cerdas”.

Menurut Rich and knight (1991)

“Kecerdasan buatan (*artificial intelligence*) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia”.

Menurut Encyclopedia Britannica

“Kecerdasan Buatan (AI) merupakan cabang dari ilmu komputer yang dalam merepresentasi pengetahuan lebih banyak menggunakan bentuk simbol-simbol daripada bilangan, dan memproses informasi berdasarkan metode heuristik atau dengan berdasarkan sejumlah aturan”.

Banyak hal yang kelihatannya sulit untuk kecerdasan manusia, tetapi untuk informatika relatif tidak bermasalah, seperti contoh: mentransformasikan persamaan, menyelesaikan persamaan integral, membuat permainan catur atau *Backgammon*. Di sisi lain, hal yang bagi manusia kelihatannya menuntut sedikit kecerdasan, sampai sekarang masih sulit untuk direalisasikan dalam Informatika, seperti contoh: pengenalan obyek/muka, bermain sepak bola.

Walaupun AI memiliki konotasi fiksi ilmiah yang kuat, AI membentuk cabang yang sangat penting pada ilmu komputer, berhubungan dengan perilaku, pembelajaran dan adaptasi yang cerdas dalam sebuah mesin. Penelitian dalam AI menyangkut pembuatan mesin untuk mengotomatisasikan tugas-tugas yang membutuhkan perilaku cerdas, termasuk contohnya adalah pengendalian, perencanaan dan penjadwalan, kemampuan untuk menjawab diagnosa dan pertanyaan pelanggan, serta pengenalan tulisan tangan, suara dan wajah. Hal-hal seperti itu telah menjadi disiplin ilmu tersendiri, yang memusatkan perhatian pada penyediaan solusi masalah kehidupan yang nyata. Sistem AI

sekarang ini sering digunakan dalam bidang ekonomi, obat-obatan, teknik dan militer, seperti yang telah dibangun dalam beberapa aplikasi perangkat lunak komputer rumah dan video *game*.

Paham Pemikiran

Secara garis besar, AI terbagi ke dalam dua paham pemikiran yaitu: AI Konvensional dan Kecerdasan Komputasional (CI, *Computational Intelligence*). AI konvensional kebanyakan melibatkan metode yang sekarang diklasifikasikan sebagai pembelajaran mesin, yang ditandai dengan formalisme dan analisis statistik, dikenal juga sebagai AI simbolis, AI logis, AI murni, dan AI cara lama (GOFAI, *Good Old Fashioned Artificial Intelligence*). Metodenya meliputi:

1. Sistem pakar: menerapkan kapabilitas pertimbangan untuk mencapai kesimpulan. Sebuah sistem pakar dapat memproses sejumlah besar informasi yang diketahui dan menyediakan kesimpulan berdasarkan pada informasi tersebut
2. Petimbangan berdasar kasus
3. Jaringan Bayesian
4. AI berdasar tingkah laku: metoda modular pada pembentukan sistem AI secara manual

Kecerdasan komputasional melibatkan pengembangan atau pembelajaran iteratif. Pembelajaran ini berdasarkan pada data

empiris dan diasosiasikan dengan AI non-simbolis, AI yang tak teratur dan perhitungan lunak. Metode pokok meliputi [9]:

1. Jaringan Syaraf: sistem dengan kemampuan pengenalan pola yang sangat kuat
2. Sistem Fuzzy: teknik-teknik untuk pertimbangan di bawah ketidakpastian, telah digunakan secara meluas dalam industri modern dan sistem kendali produk konsumen.
3. Komputasi Evolusioner: menerapkan konsep-konsep yang terinspirasi secara biologis seperti populasi, mutasi dan “*survival of the fittest*” untuk menghasilkan pemecahan masalah yang lebih baik.

Untuk melakukan aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan, yaitu [7] :

- a. Basis Pengetahuan (*Knowledge Base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
- b. Motor Inferensi (*Inference Engine*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.

1.2. Machine Learning (Mesin Pembelajaran)

Pendahuluan

Saat ini *Machine Learning* sudah banyak digunakan di dalam kehidupan, bahkan pada tempat maupun bidang yang belum

diketahui. Banyak penerapan sistem *Machine Learning* pada kehidupan, seperti pada *browser* yang dapat memunculkan hasil pencarian yang sering lakukan jika akan mencari hal yang sejenis pada waktu tertentu, dan contoh lainnya adalah *face recognition* serta *hand writing recognition*.

Pengertian

“*Machine Learning* adalah cabang dari kecerdasan buatan, merupakan disiplin ilmu yang mencakup perancangan dan pengembangan algoritma yang memungkinkan komputer untuk mengembangkan perilaku yang didasarkan kepada data empiris, seperti dari sensor data pada basis data. Sistem pembelajaran dapat memanfaatkan contoh (data) untuk menangkap ciri yang diperlukan dari probabilitas yang mendasarinya (yang tidak diketahui). Data dapat dilihat sebagai contoh yang menggambarkan hubungan antara variabel yang diamati. Fokus besar penelitian *Machine Learning* adalah bagaimana mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data. Kesukarannya terjadi karena himpunan semua perilaku yang mungkin, dari semua masukan yang dimungkinkan, terlalu besar untuk diliput oleh himpunan contoh pengamatan (data pelatihan). Karena itu *Machine Learning* harus merampatkan (generalisasi) perilaku dari contoh yang ada untuk menghasilkan keluaran yang berguna dalam kasus-kasus baru.”[10]

Terminologi

Sebelum masuk ke algoritma *Machine Learning*, maka harus diketahui terlebih dahulu mengenai beberapa terminologi dari *Machine Learning*. Misalkan akan membuat sebuah sistem klasifikasi yang merupakan sistem yang berhubungan dengan *Machine Learning*, disebut dengan *Expert System*.

Pada sistem klasifikasi, pembuatan tabel dengan kolom yang terdiri dari data untuk pengukuran yang akan dilakukan. Nama-nama kolom yang diukur disebut dengan fitur, atau dapat juga disebut atribut. Baris-baris isi dari setiap kolom merupakan *Instance* dari fitur.

Salah satu tugas dari *Machine Learning* adalah klasifikasi. Misalkan ingin menentukan suatu hal dengan sistem pakar, maka hal-hal yang bisa dilakukan, antara lain seperti menimbang dengan timbangan jika perlu tahu mengenai berat, atau menggunakan *computer vision* jika ingin mengenali suatu bentuk, dan hal-hal lainnya untuk mengumpulkan informasi. Jika semua informasi penting sudah terkumpul, hal terakhir yang akan dilakukan pastinya adalah proses klasifikasi, yang nantinya akan menghasilkan *output* berupa jenis klasifikasi yang kita inginkan. Klasifikasi pada *Machine Learning* biasanya dilakukan dengan menggunakan algoritma yang sudah sangat baik untuk digunakan dalam proses tersebut.

Jika telah menggunakan algoritma *Machine Learning* untuk klasifikasi, maka selanjutnya harus melatih algoritma yang digunakan tersebut, dan untuk melatih algoritma, maka hal yang

dilakukan adalah dengan memberi algoritma tersebut data yang berkualitas (dikenal dengan istilah *training set*). Setiap *training* mempunyai fitur dan variabel target. Target variabel adalah apa yang akan diprediksikan oleh algoritma *Machine Learning*, yang selanjutnya *Machine Learning* akan mempelajari hubungan antara fitur dan variabel target.

Tugas Utama *Machine Learning*

Pada bagian sebelumnya telah dijelaskan mengenai tugas dari klasifikasi. Pada bagian klasifikasi proses yang terjadi adalah menentukan pada *class* apa sebuah *instance* itu berada. Tugas lain dari *Machine Learning* adalah regresi. Regresi adalah prediksi dari nilai numerik. Regresi ini termasuk ke dalam tipe algoritma pembelajaran terarah yang akan dibahas pada bab selanjutnya.

Tipe Algoritma

Algoritma dalam *Machine Learning* dapat dikelompokkan berdasarkan keluaran yang diharapkan dari algoritma [10].

1. Pembelajaran terarah (*supervised learning*) membuat fungsi yang memetakan masukan ke keluaran yang dikehendaki. Misalnya pengelompokan (klasifikasi). Contoh: *k-Nearest Neighbors*, *Naive Bayes*, *Support vector machines*, *Decision trees*.

2. Pembelajaran tak terarah (*unsupervised learning*) memodelkan himpunan masukan, seperti penggolongan (*clustering*). Contoh: k-Means, DBSCAN.

Langkah-langkah Dalam Mengembangkan Aplikasi *Machine Learning* [5]

1. Mengumpulkan Data

Proses pengumpulan data dilakukan dengan mengambil contoh dari berbagai sumber informasi, seperti di Internet dan media cetak. Data yang dikumpulkan adalah data yang disebarluaskan secara bebas ke publik.

2. Mempersiapkan Data Masukan

Pada hal ini data masukan yang disiapkan adalah data masukan yang sesuai dengan format yang dibutuhkan untuk analisis.

3. Menganalisis Data Masukan

Setelah proses pertama dan kedua dilakukan, maka hal selanjutnya yang harus dilakukan adalah menganalisis data masukan dan untuk menganalisis dapat dilakukan dengan melihat pola data dan juga dengan memisahkan data berdasarkan dimensi masing-masing data.

4. Mengikutsertakan Keterlibatan Manusia

5. Melatih Algoritma

Pada langkah ini pengguna “memberi makan” algoritma dengan data yang berkualitas, dan nantinya algoritma akan mengolah data tersebut menjadi informasi serta menyimpannya.

6.Menguji Algoritma

Pada langkah ini hal yang dilakukan adalah melihat seberapa baik kualitas algoritma yang telah dilatih pada tahap sebelumnya.

7.Menggunakannya

Langkah ini merupakan langkah akhir untuk algoritma yang diterapkan dalam suatu program, sehingga dapat melakukan suatu hal. Kemudian dilakukan pengecekan ulang terhadap langkah-langkah sebelumnya.

1.3. Data Mining (Penggalian Data)

Latar Belakang

Saat ini data dapat disimpan diberbagai basis data maupun *repository*. Di tempat penyimpanan itu terdapat data yang semakin hari terus melimpah, tetapi data itu hanya menjadi tumpukan data saja yang kita tidak tahu banyak informasi di dalam data itu, maka munculah istilah “*We are Data Rich, but Information Poor*”. Dari istilah tersebut, maka muncul kebutuhan akan *tools* untuk menggali dan mengekstraksi informasi dari tumpukan data itu.

Pengertian

Berikut ini adalah pengertian data mining dari beberapa ahli:

Definisi 1.

“Data Mining adalah serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual.”(Pramudiono, 2006).

“Data Mining adalah analisis otomatis dari data yang berjumlah besar atau kompleks dengan tujuan untuk menemukan pola atau kecenderungan yang penting yang biasanya tidak disadari keberadaannya.” (Pramudiono, 2006)

Definisi 2.

“Data Mining merupakan analisis dari peninjauan kumpulan data untuk menemukan hubungan yang tidak diduga dan meringkas data dengan cara yang berbeda dengan sebelumnya, yang dapat dipahami dan bermanfaat bagi pemilik data.”(Larose, 2005).

“Data Mining merupakan bidang dari beberapa bidang keilmuan yang menyatukan teknik dari pembelajaran mesin, pengenalan pola, statistik, basis data, dan visualisasi untuk penanganan permasalahan pengambilan informasi dari basis data yang besar.” (Larose, 2005)

Data Mining memiliki beberapa nama alternatif, meskipun definisi eksaknya berbeda, seperti KDD (*Knowledge Discovery in Database*), analisis pola, arkeologi data, pemanenan informasi, dan intelegensia bisnis. Penggalian data diperlukan

saat data yang tersedia terlalu banyak (misalnya data yang diperoleh dari sistem basis data perusahaan, *e-commerce*, data saham, dan data bio informatika), tapi tidak tahu pola apa yang bisa didapatkan.

Proses Pencarian Pola

Penggalian data adalah salah satu bagian dari proses pencarian pola. Berikut ini urutan proses pencarian pola [12]:

1. Pembersihan Data, yaitu: menghapus data pengganggu (*noise*) dan mengisi data yang hilang.
2. Integrasi Data, yaitu: menggabungkan berbagai sumber data.
3. Pemilihan Data, yaitu: memilih data yang relevan.
4. Transformasi Data, yaitu: mentransformasi data ke dalam format untuk diproses dalam penggalian data.
5. Penggalian Data, yaitu: menerapkan metode cerdas untuk ekstraksi pola.
6. Evaluasi pola: yaitu mengenali pola-pola yang menarik saja.
7. Penyajian pola, yaitu: memvisualisasi pola ke pengguna.

Teknik Web Mining

Pada dasarnya penggalian data dibedakan menjadi dua fungsi, yaitu deskripsi dan prediksi. Berikut ini beberapa fungsionalitas penggalian data yang sering digunakan [12]:

1. Karakterisasi dan Diskriminasi, yaitu: menggeneralisasi, merangkum, dan mengkontraskan karakteristik data.

2. Penggalian pola berulang, yaitu: pencarian pola asosiasi (*association rule*) atau pola intra-transaksi, atau pola pembelian yang terjadi dalam satu kali transaksi.
3. Klasifikasi, yaitu: membangun suatu model yang bisa mengklasifikasikan suatu objek berdasar atribut-atributnya. Kelas target sudah tersedia dalam data sebelumnya, sehingga fokusnya adalah bagaimana mempelajari data yang ada agar klasifikator bisa mengklasifikasikan sendiri.
4. Prediksi, yaitu: memprediksi nilai yang tidak diketahui atau nilai yang hilang, menggunakan model dari klasifikasi.
5. Penggugusan/*Cluster analysis*, yaitu: mengelompokkan sekumpulan objek data berdasarkan kemiripannya. Kelas target tidak tersedia dalam data sebelumnya, sehingga fokusnya adalah memaksimalkan kemiripan intrakelas dan meminimalkan kemiripan antarkelas.
6. Analisis *outlier*, yaitu: proses pengenalan data yang tidak sesuai dengan perilaku umum dari data lainnya. Contoh: mengenali *noise* dan pengecualian dalam data.
7. Analisis trend dan evolusi: meliputi analisis regresi, penggalian pola sekuensial, analisis periode, dan analisis berbasis kemiripan.

Komponen Utama Data Mining

Arsitektur Data Mining secara umum memiliki komponen utama seperti berikut ini [4]:

1. *Database, data warehouse, WorldWideWeb*, atau *repository* informasi lain, adalah satu atau sekumpulan *database*, gudang data, *spreadsheet*, atau jenis lain dari *repository* informasi. Pembersihan data dan data teknik integrasi dapat dilakukan pada data.
2. *Database* atau *data warehouse server*, adalah *database* atau *data warehouse server* yang bertanggung jawab untuk mengambil data yang relevan berdasarkan permintaan data mining dari pengguna.
3. Basis pengetahuan, adalah pengetahuan domain yang digunakan untuk memandu pencarian atau mengevaluasi daya tarik pola yang dihasilkan. Pengetahuan tersebut dapat mencakup hirarki konsep, digunakan untuk mengatur atribut atau nilai atribut ke dalam berbagai tingkat abstraksi. Pengetahuan seperti kepercayaan pengguna, yang dapat digunakan untuk menilai daya tarik pola yang didasarkan pada ketidakterdugaan. Contoh lain dari pengetahuan domain merupakan kendala tambahan atau ambang batas, dan metadata (misalnya, menggambarkan data dari sumber heterogen).
4. Data mining *engine*, adalah penting untuk sistem data mining dan idealnya terdiri dari satu set modul fungsional untuk tugas-tugas seperti karakterisasi, asosiasi dan analisis korelasi, klasifikasi, prediksi, analisis claster, analisis outlier, dan analisis evolusi.

5. Modul pola evaluasi, merupakan komponen yang biasanya menggunakan ukuran daya tarik dan berinteraksi dengan modul data mining sehingga berfokus pencarian terhadap pola yang menarik. Modul ini dapat menggunakan ambang daya tarik untuk menyaring pola yang ditemukan, sebagai alternatif, modul pola evaluasi dapat diintegrasikan dengan modul mining, tergantung pada pelaksanaan metode data mining yang digunakan. Data mining yang efisien, sangat dianjurkan untuk mendorong evaluasi pola daya tarik sedalam mungkin ke dalam proses penambangan sehingga dapat membatasi pencarian hanya pola yang menarik.
6. *User interface*, merupakan modul yang berkomunikasi antara pengguna dan sistem data mining, yang memungkinkan pengguna untuk berinteraksi dengan sistem untuk menentukan permintaan data mining atau tugas, memberikan informasi untuk membantu memfokuskan pencarian, dan melakukan eksplorasi data mining berdasarkan pada hasil perantara data mining. Selain itu, komponen ini memungkinkan pengguna untuk menelusuri skema basis data dan gudang data atau struktur data, mengevaluasi pola yang digali, dan memvisualisasikan pola dalam bentuk yang berbeda.

Teknik Data Mining

Ada beberapa teknik data mining, yaitu: *Line Analytical Processing* (OLAP), *Classification*, *Clustering*, *Association Rule Mining*, *Temporal Data Mining*, *Time Series Analysis*, *Spatial Mining*, *Web Mining*, dll. Semua teknik itu dapat dijalankan dengan menggunakan berbagai jenis algoritma data mining.

Berikut ini adalah penjelasan untuk beberapa teknik data mining [6]:

1. Concept/Class Description

Merupakan metode untuk pembagian *class*, dengan data diasosiasikan dalam berbagai jenis, seperti pembagian jenis makanan yang dijual di pasar.

2. Association Analysis

Merupakan metode untuk mencari aturan asosiasi untuk menentukan nilai kondisi atribut. Metode ini berhubungan dengan analisis “*market basket*” dan analisis data transaksi.

3. Klasifikasi dan Prediksi

Merupakan metode untuk mengklasifikasi dan memberi prediksi terhadap suatu data, sehingga atribut yang tidak diketahui sebelumnya dapat diidentifikasi.

4. Cluster Analysis

Merupakan metode analisis tanpa label kelas data yang nantinya dapat menghasilkan label-label.

5.Outlier Analysis

Merupakan metode untuk menganalisis data yang di luar sebaran data normal. Pengujian yang dilakukan adalah pengujian statistik.

6.Evolution Analysis

Merupakan metode untuk menganalisis perubahan karakteristik data yang terjadi setiap waktunya.

Contoh Algoritma Data Mining

1. Algoritma C4.5, untuk klasifikasi data dan membuat pohon keputusan
2. Algoritma *nearest neighbor*, untuk melakukan klasifikasi data.
3. Algoritma *a priori*, untuk melakukan asosiasi data.
4. Fuzzy C Means, untuk mengklusterkan data.
5. Bayesian *Classification*, untuk pengklasifikasian statistik dan prediksi keanggotaan suatu *class*.



Bab II Tentang WEKA

“WEKA adalah sebuah paket *tools machine learning* praktis. “WEKA” merupakan singkatan dari “Waikato *Environment for Knowledge Analysis*”, yang dibuat di Universitas Waikato, New Zealand untuk penelitian, pendidikan dan berbagai aplikasi. WEKA mampu menyelesaikan masalah-masalah data mining di dunia nyata, khususnya klasifikasi yang mendasari pendekatan *machine learning*. Perangkat lunak ini ditulis dalam hirarki *class* Java dengan metode berorientasi objek dan dapat berjalan hampir di semua *platform*.

WEKA mudah digunakan dan diterapkan pada beberapa tingkatan yang berbeda. Tersedia implementasi algoritma pembelajaran *state of the art* yang dapat diterapkan pada dataset dari *command line*. WEKA mengandung *tools* untuk *pre-processing* data, klasifikasi, regresi, *clustering*, aturan asosiasi, dan visualisasi. Pengguna dapat melakukan *preprocess* pada data, memasukkannya dalam sebuah skema pembelajaran, dan menganalisis *classifier* yang dihasilkan dan performanya, semua itu tanpa menulis kode program sama sekali. Contoh penggunaan WEKA adalah dengan menerapkan sebuah metode pembelajaran ke dataset dan menganalisis hasilnya untuk memperoleh informasi tentang data, atau menerapkan beberapa metode dan membandingkan performanya untuk dipilih.

Tools yang dapat digunakan untuk *preprocessing* dataset membuat *user* dapat berfokus pada algoritma yang digunakan tanpa terlalu memperhatikan detail seperti pembacaan data dari file, implementasi algoritma *filtering*, dan penyediaan kode untuk evaluasi hasil.”[Dm Crews.2004. Modul dan Jurnal Praktek Data Mining T.A 2004/2005. STT Telkom: Bandung]



Bab III Instalasi

3.1. *System Requirement*

Berikut ini adalah spesifikasi *Personal Computer/Laptop* yang akan digunakan untuk menjalankan *Weka Machine Learning*:

- [1] Sistem Operasi: Linux, Mac OS X, Windows
- [2] Java, dengan matriks ketentuan seperti pada Tabel 3.1

Tabel 3.1. *System Requirement* untuk versi Java yang digunakan menjalankan *Weka Machine Learning* 3.6.0

		Java		
		1.4	1.5	1.6
WEKA	<3.4.0	X	X	X
	3.4.x	X	X	X
	3.5.x	3.5.0- 3.5.2	>3.5.2 r2892, 20/02/2006	X
	3.6.x		X	X
	3.7.x		3.7.0	>3.7.0 r5678, 25/06/2009

Catatan:

untuk Java 5.0 dan versi terbarunya pada Linux/GNOME mempunyai masalah pada *Look'n'Feel defaultnya*. Dari Weka, pengguna Mac OS X akan

membutuhkan Java untuk Mac OS X 10.6 *Update 3* (java 1.6.0_22) atau lebih baru.

Untuk mengunduh *installer* Weka, dapat dilakukan dari situs: <http://www.cs.waikato.ac.nz/ml/weka/unduhing.html>.

3.2. Cara Instalasi

Pada buku ini, Weka yang digunakan adalah versi 3.6.0 yang dapat diunduh dari:

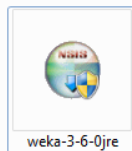
<http://sourceforge.net/projects/weka/files/weka-3-6-windows-jre/3.6.0/>,

Catatan:

tahap penginstalan untuk versi lain dari Weka kurang lebih sama.

Langkah instalasi adalah sebagai berikut:

1. Klik dua kali *file executable* dari Weka 3.6.0 (.exe) hasil proses pengunduhan



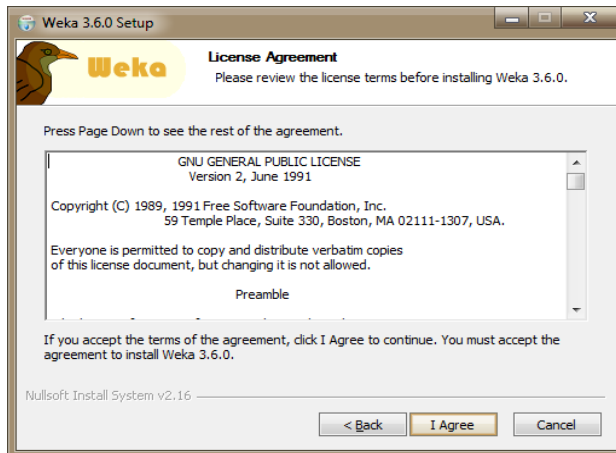
Gambar 3.1. *Installer* Weka 3.6.0

2. Pada jendela seperti tampak Gambar 3.2., pilih *Next*



Gambar 3.2. Jendela Awal *Setup* Weka 3.6.0

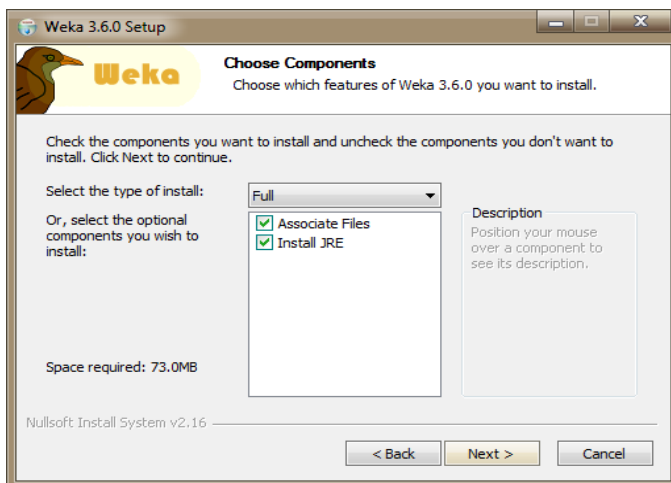
3. Pada bagian *License Agreement* (Gambar 3.3), pilih *I Agree*



Gambar 3.3. Jendela *License Agreement* Proses Instalasi Weka 3.6.0

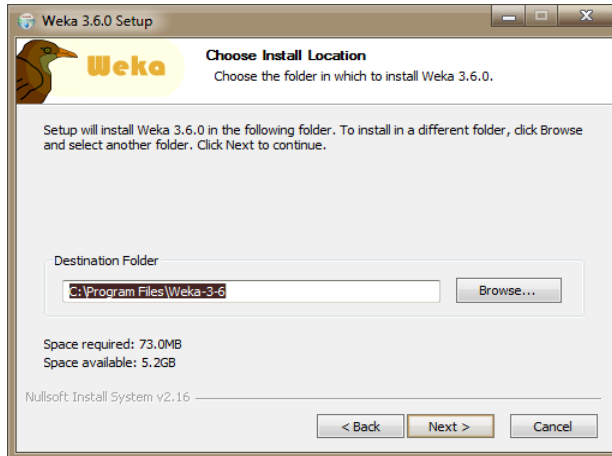
4. Jendela selanjutnya (Gambar 3.4), pada bagian *select the type of install* pilih *full*, untuk menginstall seluruh komponen yang

diperlukan untuk menjalankan aplikasi (jika JRE yang sesuai sudah *terinstall* maka hilangkan *ceklist* pada bagian *Install JRE*), lalu pilih *next*

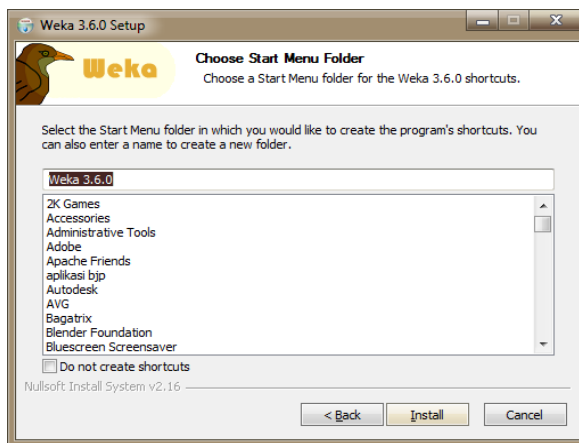


Gambar 3.4. Jendela *Choose Components*

5. Pada bagian selanjutnya (Gambar 3.5), tentukan di mana/di direktori mana Anda ingin menyimpan *file* hasil proses instalasinya (perlu diingat aplikasi ini memerlukan memori sebesar 73.0 MB, jadi pilih direktori yang Anda anggap dapat untuk menampungnya), setelah selesai menentukan direktorinya pilih *next*
6. Pada jendela selanjutnya (Gambar 3.6.), tentukan apakah Anda ingin membuat *shortcut* untuk menjalankan aplikasinya pada *start* menu atau tidak dan tentukan nama dari *shortcut*nya, selanjutnya pilih *Install*

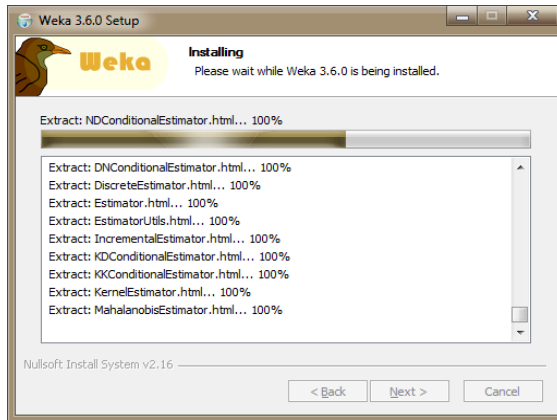


Gambar 3.5. Jendela *Choose Install Location*



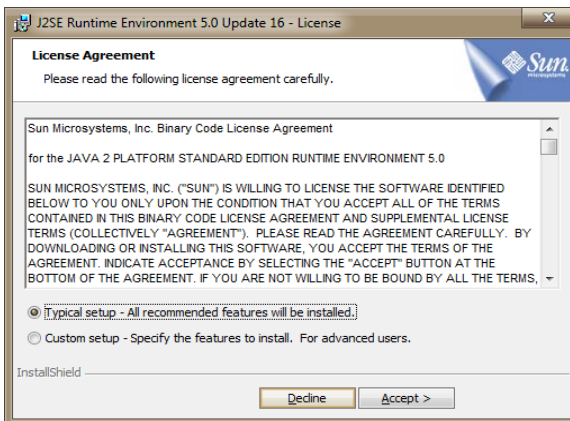
Gambar 3.6. Jendela *Choose Start Menu Folder*

7. Maka proses instalasi akan dilakukan (Gambar 3.7)



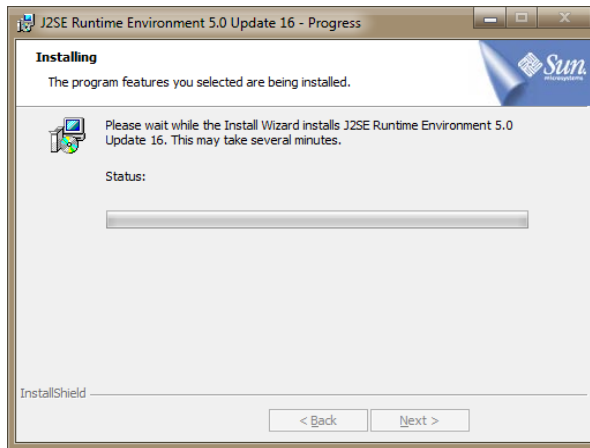
Gambar 3.7. Jendela Proses Penginstallan Weka

8. Tetapi sebelum proses selesai, anda akan diminta verifikasi untuk menginstall Java Runtime Environment untuk keperluan aplikasinya nanti, pilih *Accept* (Gambar 3.8)



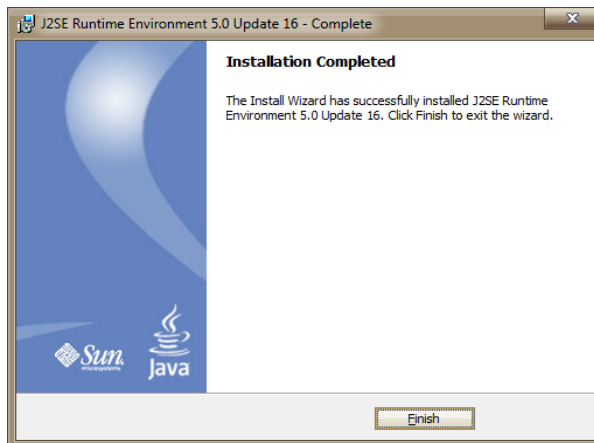
Gambar 3.8. Jendela *License Agreement* Instalasi Java Runtime Environment

9. Tunggu proses instalasi *Java Runtime Environment*nya
(Gambar 3.9)



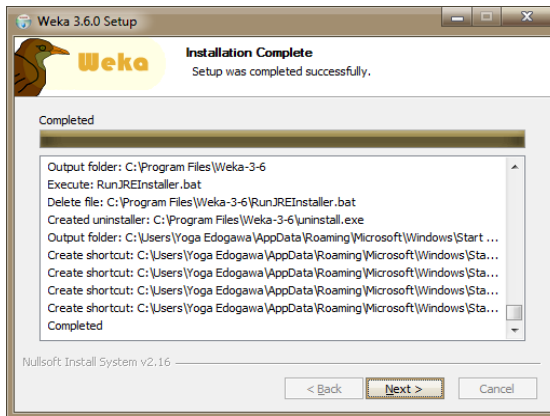
Gambar 3.9. Jendela Proses Penginstallan *Java Runtime Environment*

10. Kemudian klik *Finish* (Gambar 3.10)



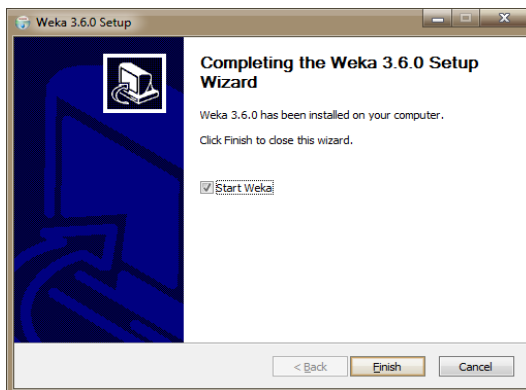
Gambar 3.10. Jendela Proses instalasi JRE Selesai

11. Setelah itu maka proses instalasi aplikasi Weka sudah selesai, klik *Next* (Gambar 3.11)



Gambar 3.11. Jendela Instalasi Weka 3.6.0 (JRE selesai diinstall)

12. Selanjutnya klik *Finish*, apabila Anda ingin langsung menjalankan aplikasi Weka maka tandai pilihan *Start Weka* (Gambar 3.12)



Gambar 3.12. Jendela Pemberitahuan Instalasi Weka 3.6.0 Selesai



Bab IV

Command Line

4.1.

Pendahuluan

Pada bagian ini dijelaskan konsep dasar dan ide, serta paket `weka.filters` digunakan untuk mengubah *input* data, misalnya untuk *preprocessing*, transformasi, fitur generasi, dan sebagainya. Kemudian fokus pada mesin belajar algoritma itu sendiri, yang disebut sebagai *Classifier* di WEKA. Pengaturan umum akan dibatasi untuk semua pengklasifikasi dan mencatat perwakilan untuk semua pendekatan utama dalam *machine learning*, serta ada contoh praktis yang diberikan.

Pada bagian akhir, di direktori dokumen WEKA dapat ditemukan dokumentasi dari semua kelas-kelas Java dalam WEKA. Jika ingin tahu persis apa yang sedang terjadi, dapat melihat pada sebagian besar kode sumber yang terdokumentasi dengan baik, yang dapat ditemukan di `weka-src.jar` dan dapat diekstraksi.

4.2. Konsep Dasar

Dataset

Satu set *item* data/dataset, adalah konsep yang sangat dasar dari pembelajaran mesin. Suatu dataset kira-kira setara dengan *spreadsheet* dua dimensi atau tabel basis data. Di WEKA, hal ini diterapkan oleh kelas *weka.core.instances*. Dataset merupakan kumpulan contoh, masing-masing dari kelas *weka.core.instance*. Setiap *instance* terdiri dari sejumlah atribut, apapun yang dapat berupa nominal (salah satu dari standar daftar nilai), numerik (bilangan real atau integer) atau string (suatu kumpulan karakter, tertutup dalam "tanda kutip ganda"). Jenis tambahan adalah *date* dan relasional, yang dibahas dalam bab ARFF. Representasi eksternal dari kelas *instance* adalah *file* ARFF, yang terdiri dari *header* yang menggambarkan jenis atribut dan data sebagai daftar yang dipisahkan koma. Berikut ini adalah contoh yang disertai keterangan. Penjelasan lengkap dari *file* ARFF Format dapat ditemukan di sini.

Komentar baris di awal dataset harus memberikan indikasi dari sumbernya, konteks dan makna.

```
% This is a toy example, the UCI weather dataset.  
% Any relation to real weather is purely coincidental
```

Di sini dicantumkan nama internal dataset.

```
@relation golfWeatherMichigan_1988/02/10_14days
```

Didefinisikan dua atribut nominal, *outlook* dan *windy*. Atribut *outlook* memiliki tiga nilai: *sunny*, *overcast* dan *rainy*, sedangkan atribut *windy* memiliki dua nilai: *TRUE* dan *FALSE*. Nilai nominal dengan karakter khusus, koma atau spasi tertutup di 'tanda kutip tunggal'.

```
@attribute outlook {sunny, overcast rainy}  
@attribute windy {TRUE, FALSE}
```

Baris ini menentukan dua atribut numerik. Integer atau numerik juga dapat digunakan. Sementara nilai *double floating point* disimpan secara internal, hanya tujuh angka desimal yang diproses.

```
@attribute temperature real  
@attribute humidity real
```

Atribut terakhir adalah target *default* atau variabel kelas yang digunakan untuk prediksi. Dalam kasus itu terdapat nominal atribut dengan dua nilai, membuat ini sebagai masalah klasifikasi biner.

```
@attribute play {yes, no}
```

Sisa dari dataset terdiri dari token @ data, diikuti oleh nilai *comma-separated* untuk atribut satu baris per contoh. Terdapat lima contoh pada kasus dibawah ini.

```
@data
sunny,FALSE,85,85,no
sunny,TRUE,80,90,no
overcast,FALSE,83,86,yes
rainy,FALSE,70,96,yes
rainy,FALSE,68,80,yes
```

Pada contoh belum disebutkan atribut jenis string, yang mendefinisikan atribut string "ganda dikutip" untuk penggalian teks (*text mining*). Pada versi terakhir WEKA, tipe atribut tanggal/waktu juga didukung. Secara *default*, atribut terakhir dianggap sebagai variabel kelas/target, yaitu atribut yang harus diprediksi sebagai fungsi dari semua atribut lainnya. Jika ini tidak terjadi, tentukan variabel target melalui -c. Nilai atribut berbasis satu indeks, yaitu c-1 menentukan atribut pertama. Beberapa statistik dasar dan validasi *file* ARFF yang diberikan dapat diperoleh melalui routine main () dari *weka.core.instances*, yaitu:

```
java weka.core.Instances data/soybean.arff
```

weka.core menawarkan beberapa *routine* berguna lainnya, misalnya *converters*. C45Loader dan *converters.CSVLoader*, yang dapat digunakan untuk mengimpor dataset C45 dan dataset koma / *tabseparated* secara masing-masing, misalnya:

```
java weka.core.converters.CSVLoader data.csv > data.arff  
java weka.core.converters.C45Loader c45_filestem > data.arff
```

Classifier

Setiap algoritma pembelajaran di WEKA berasal dari abstrak *weka.classifiers.classifierclass*. Anehnya sedikit yang diperlukan untuk *classifier* dasar: *routine* yang menghasilkan model *classifier* dari dataset pelatihan (= *buildClassifier*) dan *routine* lain yang mengevaluasi model yang dihasilkan pada dataset uji yang tidak terlihat (= *ClassifyInstance*), atau menghasilkan distribusi probabilitas untuk semua kelas (= *DistributionForInstance*).

Sebuah model *classifier* adalah pemetaan kompleks sembarang dari atribut dataset semua-tapi-satu untuk atribut kelas. Bentuk spesifik dan penciptaan pemetaan ini, atau model, berbeda dari *classifier* ke *classifier* lain, dan sebagai contoh, model ZeroR's (*weka.classifiers.rules.ZeroR*) hanya terdiri dari nilai tunggal: kelas yang paling umum, atau median dari semua nilai numerik dalam kasus memprediksi nilai numerik (*regression learning*). ZeroR adalah *classifier trivial*, tetapi memberikan *lower bound* pada kinerja dataset tertentu yang harusnya signifikan meningkat dengan pengklasifikasi lebih kompleks, karena itu adalah tes yang wajar tentang bagaimana baik kelas dapat diprediksi tanpa mempertimbangkan atribut lainnya.

Kemudian, akan dijelaskan bagaimana menginterpretasikan output dari pengklasifikasi secara rinci, dan untuk saat ini hanya fokus pada *Correctly Classified Instances* di bagian *Stratified cross-validation* dan perhatikan bagaimana mengganti dari ZeroR ke J48:

```
java weka.classifiers.rules.ZeroR -t weather.arff  
java weka.classifiers.trees.J48 -t weather.arff
```

Ada berbagai pendekatan untuk menentukan kinerja pengklasifikasi. Kinerja itu hanya dapat diukur dengan menghitung proporsi dari memprediksi contoh dalam dataset tes yang tak terlihat dengan benar. Nilai ini adalah akurasi, yang juga 1-ErrorRate. Kedua istilah ini digunakan dalam literatur. Kasus yang paling sederhana adalah menggunakan *training* set dan *test* set yang saling berdiri sendiri. Hal ini disebut sebagai estimasi *hold-out*, dan untuk memperkirakan varians dalam perkiraan kinerja, perkiraan *hold-out* dapat dihitung dengan berulang kali *resampling* dataset yang sama, yaitu penataan secara acak dan kemudian membaginya dalam *training* dan *test* set dengan proporsi tertentu dari contoh, mengumpulkan semua perkiraan pada data uji dan deviasi komputasi rata-rata dan standar akurasi.

Metode yang lebih rumit adalah lintas-validasi dengan menentukan sejumlah lipatan n . Dataset secara acak mengatur kembali dan kemudian dipecah menjadi lipatan n yang sama ukuran. Setiap iterasi, satu kawanan digunakan untuk pengujian

dan lipatan $n-1$ lainnya digunakan untuk pelatihan *classifier*. Hasil pengujian dikumpulkan dan dirata-ratakan dari semua lipatan yang akan memberikan perkiraan akurasi lintas-validasi. Lipatan dapat menjadi murni acak atau sedikit dimodifikasi untuk menciptakan distribusi kelas yang sama di setiap kali lipat seperti pada dataset lengkap. Pada kasus terakhir lintas-validasi disebut bertingkat. Validasi silang (*Cross Validation*) *Leave-one-out* menandakan bahwa n adalah sama dengan jumlah contoh. Keluar dari kebutuhan, *cv* (*croos validation*) *Leave-one-out* harus non-hirarki, yaitu distribusi kelas pada set tes tidak berhubungan dengan pelatihan data, oleh karena itu *cv* (*cross validation*) *Leave-one-out* cenderung memberikan hasil yang kurang dapat diandalkan.

Namun itu masih sangat berguna dalam berurusan dengan dataset kecil karena menggunakan jumlah terbesar data pelatihan dari dataset.

weka.filters

Paket `weka.filters` berkaitan dengan kelas yang mengubah dataset dengan menghapus atau menambahkan atribut, resampling dataset, menghapus contoh dan sebagainya.

weka.filters.supervised

Kelas dibawah `weka.filters.supervised` dalam hirarki kelas adalah untuk penyaringan *supervised*, yaitu: mengambil keuntungan dari informasi kelas.

weka.filters.supervised.attribute

Discretize digunakan untuk *discretize* atribut numerik menjadi yang nominal, berdasarkan pada informasi kelas, melalui metode MDL Fayyad & Irani, atau opsional dengan metode MDL Kononeko.

weka.filters.supervised.instance

Resample menciptakan *subsample* bertingkat dari dataset yang diberikan dan berarti bahwa distribusi kelas keseluruhan yang kurang dipertahankan dalam sampel. A terhadap distribusi kelas seragam bisa ditetapkan melalui -B.

weka.filters.unsupervised

Kelas dibawah *weka.filters.unsupervised* dalam hirarki kelas adalah untuk penyaringan *unsupervised*, misalnya versi non-bertingkat dari *Resample*. Kelas A seharusnya tidak diberikan di sini.

weka.filters.unsupervised.attribute

StringToWordVector mengubah atribut string ke vektor kata, yaitu menciptakan satu atribut untuk setiap kata yang baik mengkode keberadaan atau jumlah kata (-C) dalam string. -W dapat digunakan untuk menetapkan batas perkiraan pada jumlah kata. Ketika kelas diberikan, batas berlaku untuk masing-masing

kelas secara terpisah. *Filter* ini berguna untuk pertambangan teks (*texts mining*).

weka.filters.unsupervised.instance

Resample menciptakan *subsample* non-stratified dari dataset yang diberikan, yaitu random *sampling* tanpa memperhatikan informasi kelas. Jika tidak, setara dengan varian *supervised*.

```
java weka.filters.unsupervised.Instance.Resample -i data/soybean.arff \
-o soybean-5%.arff -Z 5
```

RemoveFolds membuat lipatan validasi silang (*Cross Validation*) dari dataset yang diberikan. Distribusi Kelas tidak dipertahankan. Contoh berikut ini membagi soybean.arff ke dataset pelatihan dan pengujian, yang terakhir terdiri dari 25% (= 1/4) dari data. *RemoveWithValues* menyaring contoh sesuai dengan nilai atribut.

weka.classifiers

Pengklasifikasi adalah inti dari WEKA. Ada banyak pilihan umum untuk pengklasifikasi, yang sebagian besar berkaitan dengan tujuan evaluasi, dan difokuskan pada hal terpenting. Semua fungsi dari classifier-parameter termasuk classifier-parameter khusus dapat ditemukan melalui perintah `-h` :

Tabel 4.1. Fungsi *Classifier*-Parameter

Fungsi	Keterangan
t	menentukan file pelatihan (ARFF format)
T	menentukan file tes dalam (ARFF format). Jika parameter ini hilang, crossvalidation akan dilakukan (default: sepuluh kali lipat cv)
X	Parameter ini menentukan jumlah lipatan untuk crossvalidation tersebut. Sebuah cv hanya akan dilakukan jika T-hilang.
C	Seperti yang sudah kita ketahui dari bagian weka.filters, parameter ini menetapkan variabel kelas dengan indeks berbasis satu.
D	Model setelah pelatihan dapat disimpan melalui parameter ini.
L	Memuat model yang disimpan sebelumnya, biasanya untuk pengujian pada baru, data yang tak terlihat sebelumnya.
P#	Jika file tes ditentukan, parameter ini berisi prediksi dan satu atribut (0 untuk tidak ada) untuk semua kasus uji.
I	Penjelasan lebih rinci kinerja melalui precision, recall, tingkat positif benar dan salah adalah tambahan output dengan parameter ini.
O	Parameter ini mengubah/mengganti output yang terbaca manusia dari model deskripsi menjadi off.

Berikut ini pada Tabel 4.2 adalah daftar singkat dari pengklasifikasi yang dipilih di WEKA. Pengklasifikasi dibawah weka.classifiers lainnya juga dapat digunakan, dan lebih mudah untuk melihat pada GUI Explorer.

Tabel 4.2. Daftar Singkat Klasifikasi WEKA

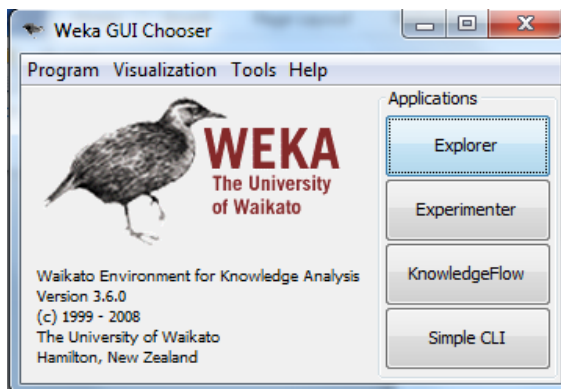
Klasifikasi	Keterangan
trees.J48 A	Sebuah tiruan pohon keputusan C4.5 untuk pembelajar/pengguna
bayes.NaiveBayes	Pembelajaran Bayesian naive. -K mengubah pada estimasi densitas kernel untuk atribut numerik yang sering meningkatkan kinerja.
meta.Classification ViaRegression-W functions.LinearReg ression	Regresi linier Multi-respon.
functions.Logistic	Regresi Logistik
functions.SMO Support Vector Machine (kernel linear, polinomial dan RBF) dengan Algoritma Optimasi Sequential Minimal]	Standar untuk SVM dengan kernel linier, -E 5 -C 10 memberikan sebuah SVM dengan kernel polinomial derajat 5 dan lambda dari 10.
lazy.KStar	Pembelajaran <i>Instance</i> -Based. -E menetapkan entropi campuran secara otomatis, yang biasanya lebih disukai.
lazy.IBk	Pembelajaran <i>Instance</i> -Based dengan neighborhood yang tetap. -K menetapkan jumlah neighbors yang digunakan. IB1 setara dengan IBK-K 1
rules.JRip	Sebuah tiruan dari pembelajaran aturan RIPPER.



Bab V GUI WEKA

5.1. Menjalankan WEKA

Berikut ini pada Gambar 5.1 adalah tampilan awal dari WEKA.



Gambar 5.1. Tampilan Awal WEKA

Launcher

Pada tampilan awal ketika aplikasi WEKA dibuka terlihat seperti pada Gambar 5.1, yaitu WEKA memiliki empat menu utama dan empat tombol. Empat menu utama tersebut adalah *program*, *visualisation*, *tools*, dan *help*.

Program

Pada menu program, terdapat tiga sub menu, yaitu:

1. *LogWindow* (Shortcut Ctrl+L)

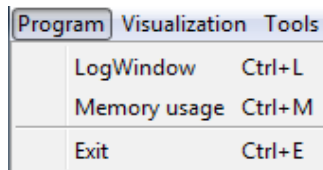
Sub menu ini berfungsi untuk menampilkan jendela Log yang merekap semua yang tercetak untuk stdout dan stderr.

2. *Memory usage* (Shortcut Ctrl+M)

Menampilkan penggunaan memori pada saat aplikasi WEKA digunakan.

3. *Exit* (Shortcut Ctrl+E)

Untuk keluar dari aplikasi WEKA.



Gambar 5.2. *Drop Down Menu Program*

Visualization

Menu ini merupakan sarana untuk memvisualisasikan data dengan aplikasi WEKA. Pada menu ini terdapat lima sub menu, yaitu:

1. *Plot* (Shortcut Ctrl+P)

Untuk menampilkan plot 2D dari sebuah dataset.

2. *ROC* (Shortcut Ctrl+R)

Untuk menampilkan kurva ROC yang telah disimpan sebelumnya.

3. *TreeVisualizer* (Shortcut Ctrl+T)

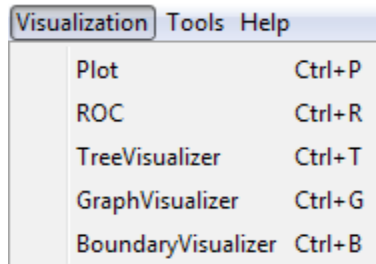
Untuk menampilkan graf berarah, contohnya: sebuah *decision tree*.

4. *Graph Visualizer* (Shortcut Ctrl+G)

Memvisualisasikan format grafik XML BIF atau DOT, contohnya sebuah jaringan Bayesian.

5. *Boundary Visualizer* (Shortcut Ctrl+B)

Mengizinkan visualisasi dari batas keputusan *classifier* dalam plot 2D.



Gambar 5.3. Drop Down Menu *Visualization*

Tools

Menu ini menampilkan aplikasi lainnya yang berguna bagi pengguna. Pada menu ini terdapat tiga sub menu, yaitu:

1. *ArffViewer* (Shortcut Ctrl+A)

Sebuah aplikasi MDI yang menampilkan file Arff dalam format *spreadsheet*.

2. *SqlViewer* (Shortcut Ctrl+S)

Merepresentasikan sebuah lembar kerja SQL, untuk melakukan *query database* via JDBC.

3. *Bayes net editor* (Shortcut Ctrl+N)

Sebuah aplikasi untuk mengedit, memvisualisasikan dan mempelajari bayes net.

Daftar Pustaka

- [1] Bouckaert, Remco R.; Frank, Eibe; dkk. 2008. *WEKA Manual for Version 3-6-0*. New Zealand: University of Waikato.
- [2] Dm Crews.2004. *Modul dan Jurnal Praktek Data Mining T.A 2004/2005*. STT Telkom : Bandung
- [3] Fadli, Ari. 2011. *Konsep Data Minning*. Ilmukomputer.com
Harrington, Peter. 2012. *Machine Learning in Action*. Manning Publications : New York.
- [4] Han, Jiawei dan Kamber, Micheline. 2006. *Data Mining: Concepts and Techniques Second Edition*. Morgan Kaufmann Publisher.
- [5] Harrington, Peter. 2012. *Machine Learning in Action*. Manning Publications: New York.
- [6] Kusrini dan Luthfi, Emha Taufiq. 2009. *Algoritma Data Mining*. Penerbit Andi : Yogyakarta.
- [7] Kusumadewi, Sri. 2003. *Artificial Inteligence (Teknik dan Aplikasinya)*. Graha Ilmu : Yogyakarta.
- [8] Mr. Deshpande, S. P. dan Dr. Thakare, V. M. *Data Mining System And Applications: A Review*. Department of MCA, D.C.P.E, H.V.P.Mandal Amravati : India.
- [9] http://id.wikipedia.org/wiki/Kecerdasan_buatan
- [10] http://id.wikipedia.org/wiki/Pembelajaran_mesin
- [11] http://id.wikipedia.org/wiki/Penggalian_data
- [12] <http://www.youtube.com/watch?v=qWj4ZxaIVF4>
- [13] <http://www.youtube.com/watch?v=r9hxW3I-XEM>



THE UNIVERSITY OF
WAIKATO

Te Whare Wānanga o Waikato

Waikato Environment for Knowledge Analysis, menyelesaikan masalah-masalah data mining, khususnya klasifikasi yang mendasar pendekatan-pendekatan machine learning. WEKA mengandung tools untuk pre-processing, klasifikasi, regresi, clustering, aturan asosiasi dan visualisasi melakukan preprocess pada data, memisukkannya dalam sebuah skema pembelajaran dan menganalisa classifier yang dihasilkan dan performansinya tanpa menulis kode program sama sekali.



Detty Purnamasari adalah staf pengajar Universitas Gunadarma, menyelesaikan Sarjana dan Magister pada bidang Sistem Informasi. Penelitian utama adalah terkait Sistem Informasi dengan titik berat pada pengumpulan data di media Internet. Kegiatan eksternal adalah bahasan terkait e-Government.



Jonathan Hindharta adalah seorang mahasiswa Strata 1 Teknik Informatika Universitas Gunadarma. Aktif dalam kegiatan kuliah serta pembuatan penulisan ilmiah berupa paper, buku, dan lainnya yang berkaitan dengan perancangan dan pengembangan segala hal pada lingkup Teknologi Informasi. Selain itu juga aktif mengikuti berbagai seminar dan workshop di bidang informatika. Saat ini sedang menyelesaikan program Strata 1 untuk selanjutnya meneruskan ke program Strata 2 sesuai dengan program Sarjana Magister Universitas Gunadarma yang sedang dijalani serta mempersiapkan diri untuk melanjutkan ke dunia kerja.



Yoga Perdana Sasmita adalah seorang mahasiswa Strata 1 Teknik Informatika Universitas Gunadarma. Aktif dalam kegiatan kuliah serta pembuatan penulisan ilmiah berupa paper, buku dan lainnya yang berkaitan dengan perancangan dan pengembangan segala hal pada lingkup Teknologi Informasi. Saat ini sedang menyelesaikan program Strata 1 untuk selanjutnya meneruskan ke program Strata 2 sesuai dengan program Sarjana Magister Universitas Gunadarma yang sedang dijalani serta mempersiapkan diri untuk melanjutkan ke dunia kerja.



Fuji Ihsani adalah seorang mahasiswa Strata 1 Teknik Informatika Universitas Gunadarma. Aktif dalam kegiatan kuliah serta pembuatan penulisan ilmiah berupa paper, buku, dan lainnya yang berkaitan dengan perancangan dan pengembangan segala hal pada lingkup Teknologi Informasi dan memiliki minat dan kecintaan terhadap dunia Fotografi. Saat ini sedang menyelesaikan program Strata 1 untuk selanjutnya meneruskan ke program Strata 2 sesuai dengan program Sarjana Magister Universitas Gunadarma yang sedang dijalani serta mempersiapkan diri untuk melanjutkan ke dunia kerja.



I Wayan Simri Wicaksana adalah Guru Besar bidang Teknologi INformasi dan telah banyak menulis karya ilmiah bidang Teknologi Informasi terutama bidang Interoperabilitas. Aktif dalam pemanfaatan solusi berbasis Open Source. Selain sebagai akademisi, dia terlihat pada berbagai proyek Sistem Informasi pada sektor pemerintah dan swasta Indonesia.