TESIS

PENENTUAN REKOMENDASI PELATIHAN PENGEMBANGAN DIRI BAGI PEGAWAI NEGERI SIPIL MENGGUNAKAN ALGORITMA C4.5 DENGAN PRINCIPAL COMPONENT ANALYSIS DAN DISKRITISASI

DETERMINING RECOMMENDATION OF SELF DEVELOPMENT TRAINING FOR CIVIL SERVANTS USING C4.5 ALGORITHM WITH PRINCIPAL COMPONENT ANALYSIS AND DISCRITIZATION

Diajukan untuk memenuhi salah satu syarat memperoleh derajat
Master of Science Ilmu Komputer

HANIF RAHMAWAN
15 / 388476 / PPA / 04915

PROGRAM STUDI S2 ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA

2017

TESIS

PENENTUAN REKOMENDASI PELATIHAN PENGEMBANGAN DIRI BAGI PEGAWAI NEGERI SIPIL MENGGUNAKAN ALGORITMA C4.5 DENGAN PRINCIPAL COMPONENT ANALYSIS DAN DISKRITISASI

Telah dipersiapkan dan disusun oleh

HANIF RAHMAWAN
15 / 388476 / PPA / 04915

Telah dipertahankan di depan Tim Penguji
Pada tanggal XX Agutus 2017

Tim Penguji

Dr. Azhari SN., M.T.

(Dosen Pembimbing) (Ketua Tim Penguji)

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tesis ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Master di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, __ Agustus 2017

Hanif Rahmawan

Karya ini kupersembahkan untuk keluargaku tercinta ^_^

Allah akan meninggikan derajat orang-orang yang berilmu

PRAKATA

Alhamdulillah. Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan nikmatnya sehingga tugas akhir ini bisa selesai. Shalawat dan salam semoga senantiasa dilimpahkan kepada Rasulullah Muhammad Shallallahu ‘alaihi wa sallam.

Selesainya penulisan laporan ini tentunya tidak lepas dari bantuan berbagai pihak. Oleh karena itu, saya mengucapkan terima kasih kepada

1. Istriku Hayati Setiyaningsih dan anak-anakku tersayang Farras Al Izzi dan Aisha El Mufida atas semua keceriaan, doa dan dukungannya
2. Bapak dan ibu tercinta serta adik-adikku tersayang atas segala dukungannya selama ini.
3. Bapak Dr. Azhari SN.,M.T. selaku dosen pembimbing yang telah membimbing, mengarahkan, memotivasi sehingga laporan ini dapat diselesaikan.
4. Teman-teman seperjuangan di Pascasarjana Ilmu Komputer UGM Angkatan Tahun 2015.
5. Rekan-rekan kerja di BKN Kantor Regional I Yogyakarta, terkhusus Assessor Kanreg I, Mas Ridlowi dan Mbak Tin, yang banyak memberikan ilmu terkait Assessment Center.
6. Sahabat SMA-ku, Hendy di BPS Bandung, yang sudah berbagi ilmu Statistiknya.
7. Semua pihak yang sudah membantu penulisan laporan ini.

Saya menyadari bahwa Tesis ini masih memiliki banyak kekurangan sehingga saran dan kritik yang membangun senantiasa saya harapkan. Namun saya tetap berharap, laporan ini tetap bisa memberi manfaat bagi para pembaca.

Yogyakarta, ____ Agustus 2017

Penulis,

Hanif Rahmawan

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN

AMT	*Achievement Motivation Training*
ARFF	*Attribute-Relation File Format*
AVF	*Atribute Value Frequency*
BKN	Badan Kepegawaian Negara
CIP	*Class Imbalance Problem*
CSV	*Comma Separated Value*
DAD	Diagram Alir Data
DBMS	*Database Management System*
EBD	*Entropy Based Discretization*
GUI	*Graphical User Interface*
IR	*Imbalance Ratio*
KDD	*Knowledge Discovery in Database*
MDLP	*Minimum Description Length Principle*
PCA	*Principal Component Analysis*
PNS	Pegawai Negeri Sipil
SDM	Sumber Daya Manusia
SMOTE	*Synthetic Minority Over-sampling Technique*
SMOTE-N	*Synthetic Minority Over-sampling Technique – Nominal*
VDM	*Value Difference Metric*
WAVF	*Weighted Atribute Value Frequency*

INTISARI

PENENTUAN REKOMENDASI PELATIHAN PENGEMBANGAN DIRI BAGI PEGAWAI NEGERI SIPIL MENGGUNAKAN ALGORITMA C4.5 DENGAN PRINCIPAL COMPONENT ANALYSIS DAN DISKRITISASI

Oleh

Hanif Rahmawan
15 / 388476 / PPA / 04915

Setiap institusi memiliki kebutuhan untuk terus meningkatkan pelayanan dan melakukan inovasi yang perlu mendapatkan dukungan dari SDM yang berkualitas. Pelatihan menjadi salah satu cara untuk mewujudkan SDM yang berkualitas. Namun terkadang penentuan pelatihan yang sesuai untuk seorang pegawai tidak mudah dan berpeluang menimbulkan ketidakkonsistenan. Masalah tersebut dapat diatasi dengan melakukan data mining terhadap data pemetaan pegawai yang sudah ada sehingga didapatkan aturan-aturan untuk penentuan rekomendasi pelatihan pengembangan diri. Data pemetaan terdiri dari nilai aspek psikologis pegawai dan rekomendasi pelatihan yang diberikan oleh assessor. Data tersebut kemudian dipecah menjadi 6 data pelatihan karena ada 6 jenis pelatihan yang digunakan.

Pada penelitian ini digunakan tiga metode, yaitu algoritma C4.5, kombinasi PCA, dan C4.5, serta kombinasi PCA, diskritisasi, dan C4.5 untuk melakukan penambangan pada data. Diskritisasi yang digunakan adalah diskritisasi berbasis entropi dengan dua macam kriteria pemberhentian yaitu berdasar jumlah interval dan MDLP. Pada tahap pra-pemrosesan digunakan teknik over-sampling SMOTE untuk menangani 4 data pelatihan yang mengalami ketidakseimbangan kelas. Pada penerapan kombinasi algoritma PCA, diskritisasi, dan C4.5 dilakukan ekstraksi fitur dengan menggunakan algoritma PCA setelah proses over-sampling dilakukan. Data hasil reduksi didiskritisasi kemudian diklasifikasi dengan algoritma C4.5.

Hasil pengujian menunjukkan bahwa kombinasi PCA, diskritisasi, dan C4.5 memberikan performa yang lebih baik daripada kedua metode yang lain. Keenam jenis pelatihan menunjukan performa terbaik ketika diproses dengan metode ini. Metode ini dapat menjadi cara alternatif untuk melakukan *pruning* terhadap pohon keputusan. Penentuan rekomendasi pelatihan pengembangan diri bagi pegawai dapat dilakukan dengan metode ini dengan rerata nilai akurasi 86,61% dan rerata nilai *F-measure* 82,23%.

Kata kunci : pohon keputusan c4.5, pelatihan pengembangan diri, principal component analysis, diskritisasi berbasis entropi

ABSTRACT

DETERMINING RECOMMENDATION OF SELF DEVELOPMENT TRAINING FOR CIVIL SERVANTS USING C4.5 ALGORITHM WITH PRINCIPAL COMPONENT ANALYSIS AND DISCRITIZATION

Oleh

Hanif Rahmawan
15 / 388476 / PPA / 04915

The need to continuously improve service and innovation becomes the need of every institution that needs to get support from qualified human resources. Training becomes one way to realize qualified human resources. But sometimes determining appropriate training for an employee is not easy. The problem can be solved by performing data mining on existing employee mapping data so that the rules for determining training recommendations are obtained. Mapping data consists of the psychological aspects of the employee and the training recommendations provided by the assessor. The data is then divided into 6 training data as there are 6 types of training used.

There are three methods used in this thesis to perform data mining, namely C4.5 algorithm, combination of PCA, and C4.5, and combination of PCA, discritization, and C4.5. The discretization used is entropy-based discretization with two kinds of stopping criteria based on the number of intervals and MDLP. SMOTE over-sampling technique is used to handle 4 training data that encountered problems of class imbalance at the pre-processing step. In the application of PCA, discrete, and C4.5 methods, features of data are extracted using PCA algorithm after over-sampling is done. The extraction result is discretized and then classified by C4.5 algorithm.

Test results show that PCA, discretization, and C4.5 methods provide better performance than the other two methods. The six types of training shows the best performance when processed with the method. This method can be an alternative way to pruning decision trees. The determination of self-development training recommendations for employees can be done with this method with an average accuracy of 86,61% and an average F-measure of 82,23%.

Kata kunci : C4.5 decision tree, self development training, principal component analysis, entropy-based discretization

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Sekarang ini, institusi baik swasta maupun pemerintah dituntut untuk lebih meningkatkan pelayanan dan senantiasa mengembangkan inovasi-inovasi baru. Dukungan dari sumber daya manusia (SDM) yang berkualitas dibutuhkan untuk dapat melakukan hal tersebut (Jantan dkk., 2011). Dengan menimbang pentingnya SDM yang berkualitas, Pemerintah Indonesia, yang dalam hal ini diwakili oleh Badan Kepegawaian Negara sebagai instansi yang ditujuk menjadi pembina kepegawaian, mendirikan *Assessment Center*. *Assessment Center* berperan dalam meningkatkan kualitas pegawai negeri sipil (PNS) sebagai SDM milik pemerintah. Beberapa bentuk kegiatan dari *Assessment Center* diantaranya adalah penilaian kompetensi dan pemetaan PNS.

Pelatihan merupakan salah satu cara untuk mewujudkan SDM yang berkualitas. Dengan memberikan pengetahuan dan keterampilan kepada pegawai, pegawai akan dapat meningkatkan pelayanan yang diberikan kepada masyarakat yang dilayani (Noe, 2009). Pelatihan juga berfungsi untuk menyelaraskan pengetahuan, sikap, dan keterampilan pegawai dengan kebutuhan organisasi (Munandar, 2006).

Pelatihan pegawai merupakan salah satu bentuk aktivitas manajemen talenta yang merupakan tugas manajemen sumber daya manusia (*human resource management*). Pengambilan keputusan dalam rangka aktivitas manajemen talenta terkadang sulit dilakukan. Di samping itu, keputusan yang dibuat juga bergantung pada berbagai macam faktor di antaranya faktor pengalaman, pengetahuan, preferensi, dan pertimbangan. Faktor-faktor tersebut dapat menyebabkan ketidakkonsistenan, ketidakakuratan, dan ketidaksamaan keputusan (Jantan dkk.,

2011). Hal-hal tersebut juga terjadi dalam penentuan kebutuhan pelatihan. Selama ini, penentuan kebutuhan pelatihan menggunakan intuisi dari *assessor* dengan memperhatikan hasil pemetaan pegawai. Intuisi yang terbentuk dari pengetahuan dan pengalaman para assessor tersebut tentu berbeda antara satu *assessor* dengan *assessor* yang lain sehingga rekomendasi tentang kebutuhan pelatihan dimungkinkan akan berbeda antara satu *assessor* dengan *assessor* yang lain dan tingkat kesulitannya pun juga berbeda.

Data mining merupakan sebuah metode akuisisi pengetahuan yang populer yang memungkinkan untuk mengekstrak informasi-infomasi implisit dan berharga dari dari sebuah data. Metode ini digunakan diberbagai macam bidang mulai dari bidang pemasaran, keuangan, kedokteran, perindustrian dan berbagai bidang lain termasuk bidang manajemen SDM. Strohmeier dan Piazza (2013) dalam penelitiannya mengatakan telah terjadi kenaikan yang cukup signifikan pada penggunaan *data mining* dalam bidang manajemen SDM.

Data mining memiliki banyak teknik dan salah satu teknik data mining yang cukup populer adalah pohon keputusan (*decision tree*). Setidaknya sampai tahun 2013 terdapat 28 penelitian terkait SDM yang menggunakan teknik pohon keputusan (Strohmeier dan Piazza, 2013).

Salah satu algoritma pohon keputusan yang banyak digunakan adalah algoritma C4.5. Jantan dkk. (2011) menyatakan bahwa C4.5 adalah algoritma yang potensial untuk digunakan dalam bidang manajemen SDM setelah membandingkan algoritma ini dengan beberapa algoritma yang lain baik yang berupa algoritma klasifikasi maupun *clustering*. Penelitian-penelitian terakhir terkait bidang manajemen SDM yang menggunakan algoritma C4.5 diantaranya yang dilakukan oleh (Saptarini, 2012), dan Sharma dan Goyal (2015), dan Li dkk. (2014) yang kesemuanya menunjukkan tingkat akurasi yang baik.

Menurut Rokach dan Maimon (2014), algoritma C4.5 sangat sensitif terhadap *noise* yang terdapat pada data, padahal menurut Han dkk. (2012) data dunia nyata cenderung tidak lengkap, tidak konsisten, dan *noisy*. Hussain dkk. (2013) dalam penelitiannya menggunakan PCA untuk mengatasi masalah tersebut. PCA digunakan untuk melakukan *feature extraction* pada data yang akan diklasifikasi menggunakan algoritma C4.5. Khalid dkk. (2014) dalam penelitiannya juga melakukan *feature extraction* dengan PCA dan variannya untuk meminimalisir efek dari noise pada saat proses pembelajaran. Hasil pengujian yang dilakukan oleh Hussain dkk. (2013) menunjukkan bahwa penggunaan kombinasi algoritma PCA dan C4.5 dapat meningkatkan efisiensi proses pembentukan pohon keputusan dan dapat juga meningkatkan akurasi dari proses klasifikasi.

Algoritma PCA dapat memberikan kinerja yang baik meskipun digunakan pada data yang distribusinya tidak seragam (Martinez dan Kak, 2001). Penggunaan algoritma PCA akan membuat data yang dihasilkan menjadi atribut kontinu (*continuous*). Penggunaan atribut kontinu pada algoritma C4.5 bukanlah suatu masalah karena algoritma C4.5 dapat melakukan klasifikasi pada data yang memiliki atribut kontinu. Penelitian Hacibeyoglu dkk. (2011) serta Kareem dan Duaimi (2014) membuktikan jika atribut kontinu didiskritisasi secara global akan meningkatkan efisiensi proses klasifikasi bahkan dapat meningkatkan akurasi. Hussain dkk. (2013) dalam penelitiannya belum melakukan diskritisasi global pada data yang akan diklasifikasi sehingga pendekatan yang dilakukannya masih berpotensi untuk ditingkatkan lagi dengan melakukan diskritisasi pada data yang akan diklasifikasi.

1.2 Rumusan Masalah

Berdasar latar belakang yang telah disebutkan, rumusan masalah dari penelitian ini adalah bagaimana menentukan rekomendasi pelatihan pengembangan diri bagi pegawai negeri sipil dengan memanfaatkan data SDM yang berupa data pemetaan pegawai dan apakah performa algoritma C4.5 akan meningkat jika

dilakukan pra-pemrosesan pada data yang digunakan dengan melakukan ekstraksi fitur dan diskritisasi.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Jenis pelatihan dibatasi pada pelatihan pengembangan diri yang diperuntukkan bagi PNS yang memangku jabatan fungsional.
2. Data yang digunakan sebagai studi kasus adalah data pemetaan PNS yang bekerja di Badan Kepegawaian Negara (BKN).
3. Proses akuisisi data dilakukan secara manual mengingat data disediakan dalam bentuk *hard copy*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menentukan rekomendasi pelatihan pengembangan diri bagi pegawai negeri sipil dengan menggunakan algoritma C4.5, PCA, dan diskritisasi serta membandingkan performa dari pendekatan tersebut dengan performa menggunakan algoritma C4.5 dan PCA, serta algoritma C4.5 saja.

1.5 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat membantu pihak yang bertanggungjawab atas manajemen SDM, khususnya di instansi pemerintah, untuk menentukan rekomendasi pelatihan pengembangan diri bagi pegawai yang bekerja di instansinya. Bagi Badan Kepegawaian Negara hasil penelitian ini dapat diintegrasikan dengan Sistem Informasi *Assessment Center* sehingga penentuan rekomendasi pelatihan pengembangan diri dapat dilakukan oleh sistem tersebut.

1.6 Kontribusi Penelitian

Dari studi literatur yang telah dilakukan tidak diketahui adanya penelitian terdahulu yang membahas tentang penentuan pelatihan untuk pegawai berdasarkan data sumber daya manusia. Salah satu metode yang digunakan untuk menyelesaikan

masalah tersebut yaitu metode PCA, diskritisasi, dan C4.5. Metode tersebut belum pernah digunakan pada penelitan-penelitian sebelumnya. Metode tersebut terinspirasi dari metode dari penelitan sebelumnya yang menggunakan algoritma PCA dan C4.5 tanpa melakukan diskritisasi.

1.7 Metode Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yaitu

1. Studi Literatur

   Studi literatur dilakukan dengan membaca berbagai buku, jurnal, artikel ilmiah maupun sumber-sumber lain yang ada di internet, serta memahami proses-proses dalam data mining.

2. Pengumpulan Data

   Pengumpulan data dilakukan dengan mewawancarai *assessor* di Badan Kepegawaian Negara dan membaca laporan hasil pemetaan pegawai Badan Kepegawaian Negara, Kantor Regional I Badan Kepegawaian Negara Yogyakarta, Kantor Regional I Badan Kepegawaian Negara Bandung, dan Kantor Regional I Badan Kepegawaian Negara Medan

3. Analisis dan Perancangan

   Pada tahap ini, data dianalisis kemudian hasilnya digunakan untuk merancang sistem yang akan dibangun. Penjelasan detail tentang tahap ini dapat dilihat pada Bab 4.

4. Implementasi

   Pada tahap implementasi dilakukan pembangunan perangkat lunak dengan bahasa pemrograman Python dan PHP berdasarkan hasil perancangan yang telah disusun pada tahap sebelumnya.

5. Pengujian dan Analisis Hasil

   Pengujian sistem dilakukan dengan mengukur akurasi, presisi, *recall, F-Measure*. Hasil pengujian dianalisis dan dibandingkan antara metode C4.5, PCA dan C4.5, serta PCA, diskritisasi. dan C4.5.

1.8 Sistematika Penulisan

Dalam penulisan tugas akhir ini, sistematika penulisan yang digunakan adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini memberi gambaran umum tentang tugas akhir ini yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, serta sistematika penulisan.

BAB II KAJIAN PUSTAKA

Bab ini berisi kajian terhadap penelitian-penelitian sebelumnya yang memiliki keterkaitan dengan penelitian yang dilakukan. Penelitian-penelitian tersebut dijadikan sebagai dasar rujukan untuk penelitian ini.

BAB III DASAR TEORI

Pada bab ini dijelaskan mengenai ringkasan dasar teori terkait *data mining* dan *assesment center* yang digunakan dalam penelitian ini.

BAB IV ANALISA DAN PERANCANGAN

Bab ini mengandung uraian mengenai tahap – tahapan yang dilalui dalam penelitian ini meliputi pengumpulan data, perancangan proses pembuatan model, dan pembuatan aplikasi untuk pengguna.

BAB V IMPLEMENTASI

Bab ini berisi tentang penerapan hasil perancangan ke dalam bentuk program komputer dengan bahasa pemrograman terpilih.

BAB VI HASIL DAN PEMBAHASAN

Bab ini menjelaskan langkah – langkah pengujian, hasil pengujian dan pembahasan hasil pengujian.

BAB VII KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran - saran jika akan dilakukan penelitian yang sejenis.

BAB II
KAJIAN PUSTAKA

Menurut Strohmeier dan Piazza (2013), penelitian tentang *data mining* di bidang HRM (*Human Resource Management*) sudah cukup banyak. Setidaknya sampai dengan tahun 2011 sudah ada 121 penelitian. Penelitian-penelitian tersebut didominasi bidang *staffing* yang meliputi aktivitas perencanaan kebutuhan pegawai, rekrutmen pegawai, pembagian kerja sampai pemecatan pegawai. Penelitan SDM dalam hal *staffing* dilakukan diantaranya dilakukan oleh Jantan dkk. (2011), Saptarini (2012), dan Li dkk. (2014). Jantan dkk. (2011) menggunakan dan membandingkan beberapa algoritma klasifikasi dan *clustering* untuk memprediksi kelayakan seorang dosen untuk dipromosikan. Hasil penelitannya menunjukkan bahwa algoritma C4.5 menghasilkan akurasi paling tinggi sehingga algoritma C4.5 dirasa cukup potensial untuk digunakan dalam penelitian terkait SDM. Saptarini (2012) menggunakan algoritma C4.5 yang dikombinasikan dengan logika fuzzy untuk memprediksi jabatan karyawan berdasar hasil tes psikologi. Li dkk. (2014) dalam penelitiannya menggunakan algoritma C4.5 untuk memprediksi unit yang tepat bagi seorang pegawai dengan memperhatikan kemampuannya.

Pada rangking kedua, penelitian bidang SDM didominasi sub bidang pengembangan yang meliputi aktivitas pelatihan dan perencanaan karir (Strohmeier dan Piazza, 2013). Penelitian terkait sub bidang pengembangan diantaranya dilakukan oleh Chen dkk. (2007) Pada penelitannya, Chen dkk. (2007) membuat sebuah sistem pakar yang digunakan untuk menentukan strategi pelatihan yang tepat bagai pegawai dengan memperhatikan kemampuan belajar, pekerjaan, dan faktor-faktor lain.

Penelitian SDM pada rangking ketiga diduduki oleh penelitian dalam bidang manajemen kinerja (Strohmeier dan Piazza, 2013). Salah contoh penelitian dalam bidang tesebut adalah penelitian yang dilakukan oleh Sharma dan Goyal (2015). Sharma dan Goyal (2015) membandingkan algoritma C4.5 dan Naïve Bayes untuk

menentukan tingkatan kinerja pegawai. Hasil dari penelitiannya menunjukkan algoritma C4.5 menghasilkan akurasi yang lebih baik dibanding Naïve Bayes dalam kasus penentuan tingkatan kinerja dengan tingkat akurasi 93%.

Data yang diklasifikasi dapat berupa hasil proses *feature extraction*. Hussain dkk. (2013) dan Martono (2012) melakukan *feature extraction* dengan menggunakan algoritma *Principal Component Analysis* (PCA)/Analisis Komponen Utama. Namun. kedua penelitian tersebut memiliki perbedaan. Hussain dkk. (2013) menggunakan variabel baru hasil PCA untuk diproses dengan algoritma C4.5, sedangkan Martono (2012) menggunakan variabel asli yang terpilih dari hasil rotasi faktor Varimax. Pada penelitian Hussain dkk. (2013), hasil pengujian menunjukkan adanya peningkatan akurasi dengan nilai rata-rata sebesar 6,46% pada 29 *dataset* UCI yang digunakan. Pada penelitian Martono (2012), hasil pengujian menunjukkan adanya penurunan akurasi sebesar 5,05% pada *dataset* jantung koroner.

Pra-pemrosesan lain yang dapat dilakukan adalah dengan melakukan diskritisasi atribut data. Saptarini (2012) melakukan diskritisasi dengan menggunakan algoritma Fuzzy dan hasil pengujian menunjukkan peningkatan akurasi 4%. Hacibeyoglu dkk. (2011) melakukan diskritisasi terawasi pada 6 *dataset* UCI untuk melakukan pengujian dan hasilnya menunjukkan peningkatan akurasi. Peningkatan akurasi rata-rata dengan menggunakan algoritma k-NN sebesar 12,31%, algoritma Naïve Bayes sebesar 1,86%, algoritma C4.5 1,71%, dan algoritma CN2 sebesar 2,79%. Kareem dan Duaimi (2014) melakukan diskritisasi tidak terawasi pada 3 *dataset* UCI untuk melakukan pengujian dan hasilnya menunjukkan peningkatan akurasi sebesar 1% pada *dataset bank marketing* dan diabetes dan 4% pada *dataset credit approval*.

Perbedaan penelitian ini dengan penelitian sebelumnya adalah

1. Penelitian ini bertujuan untuk menentukan kebutuhan pelatihan pengembangan diri yang sebelumnya belum pernah diteliti oleh peneliti lain.

2. Metode yang digunakan dalam penelitian ini adalah dengan melakukan ekstraksi fitur pada data yang akan diklasifikasi dengan algoritma PCA kemudian hasilnya dilakukan diskritisasi menggunakan algoritma diskritisasi terawasi. Hussain dkk. (2013) dan Martono (2012) juga melakukan ekstraksi fitur pada data yang akan diklasifikasi tetapi tidak melakukan diskritisasi setelah dilakukan proses PCA.

Ringkasan kajian pustaka yang telah diuraikan di atas dapat dilihat pada Tabel 2.1 dan Tabel 2.2.

Tabel 2.1 Kajian pustaka terkait bidang SDM

Peneliti	Metode	Keterangan
Chen dkk. (2007)	Sistem pakar berbasis aturan	**Tujuan**: menentukan strategi pelatihan yang cocok untuk pegawai Parameter yang digunakan adalah *seniority*, departemen tempat pegawai bekerja, jabatan, jumlah pelatihan yang pernah diikuti, dan golongan.
Jantan dkk. (2011)	C4.5, Random Forest, MLP, RBFN, K-Star	**Tujuan klasifikasi**: menentukan kelayakan seorang dosen untuk dipromosikan **Kelas target**: layak dipromosikan, tidak layak dipromosikan **Hasil**: Algoritma C4.5 memiliki akurasi yang paling tinggi dibandingkan keempat algoritma yang lain dengan tingkat akurasi rata-rata dari pengujian menggunakan 8 dataset adalah 66.76%.
Saptarini (2012)	C4.5, Logika Fuzzy	**Tujuan klasifikasi**: mengelompokkan jenis jabatan karyawan berdasarkan hasil tes psikologi **Pre-processing**: dilakukan fuzzifikasi terhadap variabel input dikarenakan variabel input bernilai kontinu **Kelas target**: administrasi, keuangan, humas, dan laboran/teknisi **Hasil**: Hasil pengujian menunjukkan rerata akurasi algoritma fuzzy C4.5 adalah 84.9% dan rerata akurasi algoritma C4.5 konvensional 80,2%.
Li dkk. (2014)	C4.5	**Tujuan**: memprediksi unit yang tepat bagi seorang pegawai berdasar kemampuan bahasa, kemampuan komputer, jenjang pendidikan, dan kemampuan praktek. **Kelas target**: 6 Unit **Hasil**: Pohon keputusan yang dibangun dapat mengklasifikasikan pegawai dengan benar dan cepat

Tabel 2.1 (lanjutan)

Peneliti	Metode	Keterangan
Sharma dan Goyal (2015)	C4.5, Naïve Bayes Classifier	**Tujuan klasifikasi**: menentukan tingkatan performa pegawai **Pra-pemrosesan**: analisis korelasi untuk menghilangkan atribut yang *redundant*, dan diskritisasi atribut kontinu **Kelas Target**: 3 kelas, *good*, *satisfactory*, dan *need improvement* **Hasil**: Algoritma C4.5 menghasilkan akurasi yang lebih tinggi dibandingkan algoritma Naïve Bayes.

Tabel 2.2 Kajian pustaka terkait algoritma yang digunakan

Peneliti	Metode	Keterangan
Hacibeyoglu dkk. (2011)	k-NN, Naïve Bayes, C4.5, CN2, Entropy Based Discretization	***Dataset***: Dataset yang digunakan adalah 6 *dataset* dari UCI **Langkah-langkah**: Pada penelitian ini dilakukan diskritisasi terhadap data yang akan diproses menggunakan algoritma k-NN, Naïve Bayes, C4.5, dan CN2. Diskritisasi dilakukan dengan menggunakan algoritma diskritisasi berbasis *entropy*. **Hasil**: Hasil pengujian menunjukkan bahwa diskritisasi dapat meningkatkan akurasi dari proses klasifikasi dan *clustering* pada 80% *dataset* yang digunakan.
Hussain dkk. (2013)	C4.5, PCA	**Dataset**: *Dataset* yang digunakan adalah 40 *dataset* dari UCI. **Langkah-langkah**: Pada penelitian ini, data yang akan diproses menggunakan algoritma C4.5 dicari atribut utamanya dulu menggunakan algoritma PCA. **Hasil**: Hasil pengujian pada 29 *dataset* menunjukkan peningkatan akurasi rata-rata 6,46% jika dibandingkan pemrosesan dengan algoritma C4.5 biasa, 2 *dataset* akurasinya sama, dan 9 *dataset* menunjukkan penurunan akurasi.
Martono (2012)	C4.5, PCA	***Dataset***: data yang digunakan adalah penyakit jantung koroner dari *dataset* UCI **Langkah-langkah**: Pada penelitian ini, data yang akan diproses menggunakan algoritma C4.5 dicari atribut utamanya dulu menggunakan algoritma PCA. Namun pada saat diproses dengan algoritma C4.5 yang diproses bukan variabel baru hasil dari proses PCA melainkan variabel asli yang diperoleh dengan mengembalikan variabel baru ke variabel asli menggunakan fungsi rotasi Varimax. **Hasil**: Hasil pengujian menunjukkan bahwa akurasi yang dihasilkan lebih rendah daripada penggunaan algoritma C4.5 biasa. Kombinasi C4.5 dan PCA ini menghasilkan akurasi 75,42% sedangkan algoritma C4.5 biasa menghasilkan akurasi 80,47%.

Tabel 2.2 (lanjutan)

Peneliti	Metode	Keterangan
Kareem dan Duaimi (2014)	C4.5, clustering	**Dataset**: Dataset yang digunakan adalah 3 dataset dari UCI. **Langkah-langkah**: Pada penelitian ini dilakukan diskritisasi menggunakan algoritma diskritisasi tak terawasi terhadap data yang akan diproses.Data yang telah didiskritisasi kemudian diproses menggunakan algoritma C4.5.. **Hasil**: Hasil pengujian menunjukkan bahwa diskritisasi menggunakan algoritma diskritisasi tak terawasi dapat meningkatkan akurasi dari proses klasifikasi dengan peningkatan akurasi rata-rata 1,62%.

BAB III
DASAR TEORI

3.1 Assesment Center

Metode *Assessment Center* adalah sebuah prosedur yang digunakan oleh manajemen sumber daya manusia (SDM) untuk mengevaluasi dan mengembangkan SDM dalam hal yang relevan dengan kebutuhan organisasi. *Assessment Center* dalam kaitannya dengan manajemen SDM digunakan untuk tujuan yaitu penentuan personil yang akan dipromosikan, mendiagnosis kekuatan dan kelemahan dalam hal keterampilan yang terkait pekerjaan, dan membangun keterampilan yang terkait dengan pekerjaan (Thornton dan Rupp, 2006).

Badan Kepegawaian Negara (BKN) diberi tugas oleh Pemerintah untuk melakukan manajemen kepegawaian. BKN telah mempelopori berdirinya *Assessment Center*. Perka BKN nomor 12 tahun 2008 tentang Pedoman Penyelenggaraan *Assessment Center* PNS sebagai dasar didirikannya *Assessment Center*. *Assessment Center* saat ini difungsikan untuk keperluan seleksi pemangku jabatan, pengembangan keahlian, identifikasi kader pemimpin, dan juga pemetaan pegawai.

Pada aktivitas pemetaan pegawai, terdapat 14 aspek pokok psikologi yang diukur (Badan Kepegawaian Negara, 2011):

1. Potensi kecerdasan: Kemampuan seseorang menggunakan seluruh aspek intelektual yang dimiliki.
2. Daya konseptual: Kemampuan untuk mengidentifikasi, berpikir induktif, mengkombinasikan, menghubungkan, mengabstraksikan, berpikir logis melalui bahasa, dan membangun gagasan.
3. Daya analisis: Kemampuan untuk memecah pola, berpikir deduktif, berpikir logis, membayangkan, dan membuat kesimpulan.
4. Fleksibilitas berpikir: Kelincahan berpikir/mengubah alur pikir dan kemampuan

untuk berpikir dengan menggunakan pendekatan yang bervariasi

5. Kemampuan numerikal: Kemampuan untuk berpikir praktis aritmatis dan teoritis dengan menggunakan bilangan, serta berpikir logis-matematis.
6. Sistematika kerja: Bekerja secara teratur sesuai dengan prosedur/aturan, rapih, terarah menuju hasil.
7. Hasrat berprestasi: Dorongan untuk selalu meningkatkan kinerja dengan lebih baik dan di atas standar secara terus menerus.
8. Inisiatif: Kemampuan untuk mengambil langkah-langkah aktif tanpa menunggu perintah.
9. Stabilitas emosi: Kemampuan untuk dapat mengendalikan emosi dalam menghadapi berbagai situasi.
10. Kepercayaan diri: Keyakinan pada kemampuan dan tampilan diri.
11. Penyesuaian diri: Kemampuan untuk memposisikan diri dalam perubahan lingkungan dan bekerja secara efektif dalam situasi dan kondisi yang berbeda.
12. Kerjasama: Kemampuan untuk bekerja secara kooperatif dalam kelompok yang berbeda untuk mencapai tujuan organisasi.
13. Toleransi terhadap stress: Kemampuan untuk tetap bekerja secara produktif dan efektif meskipun dalam situasi yang sulit atau di bawah tekanan.
14. Kepemimpinan: Kemampuan mendominasi orang lain, meyakinkan, mempengaruhi, dan memotivasi orang lain serta bertanggungjawab atas keputusan tindakan yang diambil.

Masing-masing aspek psikologi tersebut dinilai dengan 5 tingkatan seperti pada Tabel 3.1.

Assessment Center pada umumnya digunakan untuk mendiagnosa kelemahan pegawai dan untuk menyediakan pelatihan keterampilan di bidang tertentu. Hal tersebut juga telah dilakukan di *Assessment Center* milik BKN. Dalam setiap laporan dari kegiatan penilaian kompetensi baik untuk seleksi jabatan maupun pemetaan

pegawai selalu disertai rekomendasi pelatihan yang diperlukan oleh masing-masing pegawai. Penentuan rekomendasi pelatihan tersebut dilakukan oleh *assessor* dengan melihat nilai kompetensi dari pegawai.

Tabel 3.1 Rincian nilai aspek psikologi

Nilai	Keterangan
1	jauh di bawah rata-rata kemampuan orang pada umumnya
2	di bawah rata-rata kemampuan orang pada umumnya
3	seperti rata-rata kemampuan orang pada umumnya
4	di atas rata-rata kemampuan orang pada umumnya
5	jauh di atas rata-rata kemampuan orang pada umumnya

Pelatihan dan pengembangan adalah proses penyampaian pengetahuan, keterampilan, kemampuan, dan karakteristik lain yang diperlukan oleh seorang individu agar dapat mendukung fungsi organisasi menjadi lebih efektif (Thornton dan Rupp, 2006).

Pelatihan pengembangan diri (*personnel development training*) merupakan salah satu jenis pelatihan yang diberikan bagi PNS. Bagi pemegang jabatan fungsional, terdapat 6 macam pelatihan pengembangan diri sebagai berikut (Kantor Regional I BKN, 2011):

1. *Achievement Motivation Training*
2. *Effective Communication Skill*
3. *Human Skill Improvement*
4. *Personnel Effectiveness*
5. *Readiness To Change*
6. *Team Building*

3.2 Data Mining

Data mining adalah proses untuk menemukan informasi yang berharga dari sekumpulan data berukuran besar secara otomatis (Larose, 2014). Data mining adalah bagian utama dari *knowledge discovery in database* (KDD), yang merupakan proses

mengubah data mentah menjadi informasi yang bermanfaat (Tan dkk., 2005). Sumber data untuk proses *data mining* dapat berupa basis data, *data warehouse*, web, penyimpanan informasi lainnya, atau data yang mengalir ke dalam suatu sistem secara dinamis (Han dkk., 2012).

KDD dibagi ke dalam 9 tahap yang gambarannya dapat dilihat pada Gambar 3.1. Kesembilan tahapan KDD tersebut sebagai berikut (Maimon dan Rokach, 2010):

1. Membangun sebuah pemahaman tentang domain aplikasi

   Orang yang bertanggung jawab dalam sebuah proyek KDD perlu memahami dan mendefinisikan tujuan dari *end-user* dan lingkungan tempat proses KDD akan dilakukan.

2. Memilih dan membangun *dataset*

   Pada tahap ini dilakukan pencarian informasi mengenai ketersediaan data, pengumpulan data tambahan yang diperlukan, dan pengintegrasian semua data menjadi sebuah *dataset*.

3. Pra-pemrosesan dan pembersihan terhadap data

   Pada tahap ini, reliabilitas data ditingkatkan dengan menangani *missing values* dan penghapusan *noise* dan *outliers*.

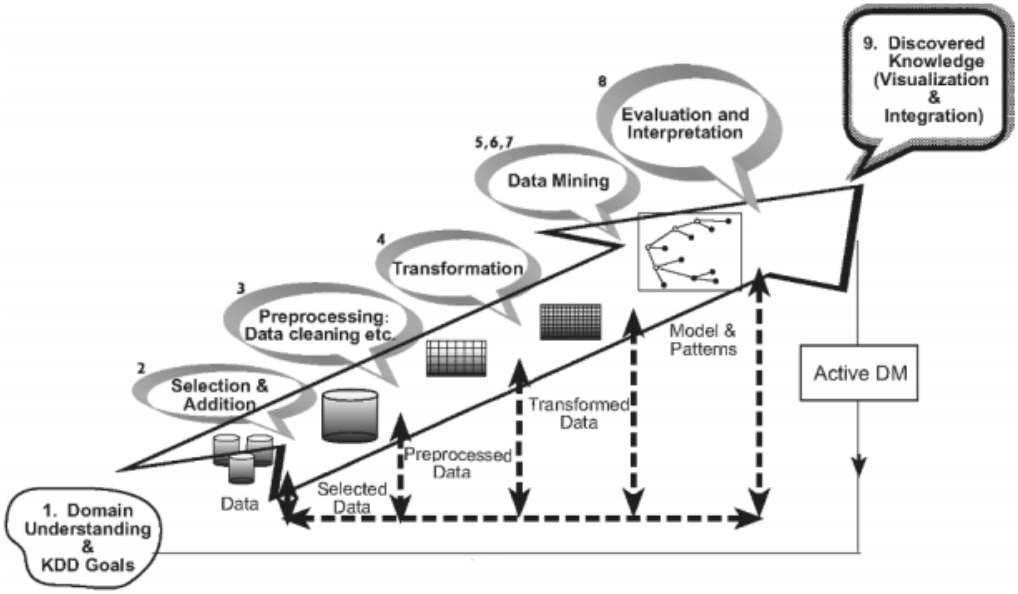
4. Transformasi data

   Pada tahap ini, proses pengolahan data agar menjadi lebih baik dilakukan. Metode yang digunakan antara lain pengurangan dimensi (misal pemilihan dan ekstraksi fitur) dan transformasi atribut (misal diskritisasi).

5. Menentukan jenis pekerjaan data mining yang akan dilakukan

   Ada 6 jenis pekerjaan yang dilakukan dengan *data mining* yaitu, deskripsi, prediksi, klasifikasi, estimasi, asosiasi, dan *clustering* dan lain-lain. Penentuan jenis pekerjaan mana yang akan dilakukan harus memperhatikan tujuan dari proyek KDD.

6. Memilih algoritma *data mining* yang akan digunakan



Gambar 3.1 Proses KDD pada basisdata (Maimon dan Rokach, 2010)

7. Mengimplementasikan algoritma *data mining*

   Eksekusi algoritma *data mining* bisa dilakukan berkali-kali sampai didapatkan hasil yang memuaskan dengan cara melakukan pengaturan parameter tiap kali perulangan.

8. Mengevaluasi hasil proses *data mining*

   Pada tahapan ini, efek dari langkah-langkah pra-pemrosesan dipertimbangkan berkaitan dengan hasil dari proses *data mining*. Selain itu, pengetahuan yang diperoleh juga didokumentasikan.

9. Menggunakan pengetahuan yang diperoleh dari proses *data mining*

   Pengetahuan yang diperoleh dari proses *data mining* dapat divisualisasikan dan diintegrasikan. Dengan visualisasi, penganalisis data dapat mengekplorasi data dan hasil *data mining* dari berbagai macam sudut pandang. Hasil dari *data mining* juga dapat diintegrasikan dengan sistem informasi maupun sistem pendukung keputusan (Tan dkk., 2005).

Menurut Larose (2014), pekerjaan-pekerjaan yang paling umum dilakukan dengan data mining sebagai berikut:

1. Deskripsi

   Data mining dapat digunakan untuk mendeskripsikan data supaya pola data tersebut lebih jelas sehingga lebih mudah untuk dipahami oleh manusia. Salah satu metode *data mining* yang cocok untuk keperluan ini adalah pohon keputusan.

2. Estimasi

   Data mining untuk keperluan estimasi dilakukan dengan memperkirakan nilai variabel target yang berupa nilai numerik dengan menggunakan sekumpulan variabel prediksi dalam bentuk numerik atau kategorik. Sebagai contoh, estimasi tekanan darah berdasar data pasien rumah sakit yang meliputi umur pasien, jenis kelamin, *body mass index*, dan *blood sodium level*. Hubungan antara tekanan darah dan variabel prediksi dalam data pelatihan akan menghasilkan sebuah model estimasi.

3. Klasifikasi

   Klasifikasi hampir sama dengan estimasi hanya saja variabel targetnya dalam bentuk kategorik, bukan numerik. Contoh dalam klasifikasi pendapatan dapat dibagi menjadi kategori, pendapatan kelas tinggi, kelas menengah, dan kelas rendah.

4. Prediksi

   Prediksi hampir sama dengan estimasi maupun klasifikasi. Perbedaannya hasil prediksi merupakan gambaran di masa yang akan datang. Contoh, prediksi harga untuk 3 bulan ke depan, prediksi jumlah kecelakaan lalu lintas ketika batas kecepatan dinaikkan.

5. *Clustering*

   Clustering merujuk pada aktivitas pengelompokan baris data (*record*), hasil pengamatan, atau kasus ke dalam suatu *cluster*. Sebuah *cluster* berisi kumpulan

dari baris data yang memiliki kemiripan satu sama lain, dan berbeda dengan baris data yang ada pada *cluster* lainnya. Algoritma *clustering* akan membagi-bagi seluruh *dataset* ke dalam *cluster-cluster* yang homogen sehingga kesamaan antar baris data dalam *cluster* tersebut menjadi maksimal, dan kesamaan baris data suatu *cluster* dengan baris data di *cluster* lainnya menjadi minimal.

6. Assosiasi

   Data mining untuk keperluan assosiasi bertujuan untuk menemukan atribut yang sering muncul bersamaan dan mendapatkan aturan untuk menghitung hubungan antara atribut-atribut tersebut. Aturan assosiasi adalah bentuk aturan "*if antecedent then consequent*" yang dilengkapi dengan ukuran *support* dan *confidence* dari aturan tersebut.

3.3 Data Cleaning

Data dunia nyata cenderung tidak lengkap, tidak konsisten, dan *noisy*. Data-data dengan kecenderungan tersebut diistilahkan sebagai *dirty data* (data kotor) (Han dkk., 2012). *Dirty data* akan dapat menyebabkan kesalahan dalam pengambilan keputusan dan analisis (Chu dkk., 2016). Untuk mengatasi masalah tersebut, dalam tahapan *data mining* diperlukan pembersihan data atau yang disebut juga data *cleaning* (*cleansing*). Tahapan *data cleaning* meliputi upaya –upaya untuk menangani *missing values*, *noise*, dan data yang tidak konsisten (Han dkk., 2012).

3.3.1 Outlier

Outlier adalah hasil pengamatan yang tampak menyimpang dari sebagian besar hasil pengamatan yang lain (Ben-gal, 2010). *Outlier* diartikan juga suatu kejadian yang tidak biasa di dunia nyata. *Outlier* mungkin dihasilkan dari mekanisme yang berbeda dari sistem dan frekuensi kemunculannya sangat rendah (Rokhman dkk., 2016). *Outlier* dapat terjadi karena kesalahan sistem, perubahan perilaku sistem, tindakan curang, intrusi jaringan, atau kesalahan manusia(Doja dkk., 2012)

Outlier yang berada jauh dari pusat distribusi normal dapat menyebabkan bias yang signifikan pada operasi statistik, misal pada rata-rata dan standar deviasi. *Outlier* juga berpengaruh pada perkiraan koefisien korelasi pada model regresi. Keberadaan *outlier* dapat menyebabkan perhitungan *information gain* pada algoritma pohon keputusan menjadi tidak akurat dan akibatnya akurasi dari model pohon keputusan menjadi turun (Last dan Kandel, 2001). Ben-gal (2010) juga menyatakan bahwa *outlier* dapat menyebabkan model menjadi tidak sesuai spesifikasi, bias pada parameter estimasi dan hasil yang tidak sesuai. Oleh karena itu, penghapusan *outlier* akan memberikan dampak positif dalam proses *data mining*. Salah satu dampak positifnya adalah peningkatan akurasi klasifikasi seperti terbukti pada penelitian yang dilakukan oleh (Doja dkk., 2012), dan (Krishnan dkk., 2015).

3.3.2 Deteksi Outlier

Penghapusan *outlier* merupakan salah satu cara peningkatan performa. Cara tersebut didahului proses deteksi *outlier* sehingga proses deteksi *outlier* ini merupakan langkah penting dalam tahapan data mining. Berbagai metode untuk deteksi *outlier* telah dikembangkan dan sebagian besar berfokus pada pemrosesan data numerik. Metode-metode untuk data numerik yang umum digunakan di antaranya metode berbasis statistik, metode berbasis jarak (*distance*), dan metode berbasis *density*.

Data kategorik atau data bukan numerik harus dipetakan ke dalam data numerik dulu untuk dapat dideteksi *outlier*-nya. Salah satu metode untuk mendeteksi *outlier* pada data kategorik adalah metode AVF (*Attribute Value Frequency*) yang menggunakan frekuensi data untuk memetakan data bukan numerik ke dalam bentuk numerik. Metode ini diawali dengan menghitung frekuensi kemunculan sebuah nilai dari suatu atribut pada masing-masing atribut. Metode ini sudah memiliki beberapa pengembangan diantaranya MR-AVF, AEFV, NAVF, dan OPAVF (Rokhman dkk., 2016).

3.3.3 Algoritma WAVF

Algoritma WAVF (*Weighted Attribute Value Frequency*) merupakan salah satu pengembangan metode AVF. Metode ini menggantikan penggunaan frekuensi nilai atribut pada algoritma AVF dengan probabilitas nilai atribut. Selain itu, metode ini juga meningkatkan performa dari metode AVF dengan mempertimbangkan tingkat kemunculan yang rendah (*sparseness)* dari tiap-tiap atribut. Tingkat *sparseness* digunakan sebagai fungsi pembobotan untuk probabilitas nilai atribut. Fungsi pembobotan tersebut membuat data yang frekuensi nilai atributnya paling jarang muncul diduga kuat sebagai *outlier* (Rokhman dkk., 2016). Algoritma WAVF dapat dilihat pada Gambar 3.2.

3.4 Class Imbalance Problem (CIP)

CIP adalah kondisi terdapat satu kelas yang ukurannya sangat kecil sekali atau sangat besar sekali jika dibandingkan dengan kelas lainnya (Amin dkk., 2016). Misal, pada suatu *dataset* 99% datanya dilabeli kelas A dan 1% dilabeli kelas B. Pada contoh tersebut, kelas A merupakan kelas mayoritas dan B adalah kelas minoritas. Kondisi tersebut memungkinkan *classifier* untuk mendapatkan akurasi 99% hanya dengan mengabaikan 1% kelas B. Akurasi yang diperoleh akan cenderung tinggi namun kelas minor terabaikan, *classifier* cenderung mengklasifikasikan data sebagai kelas mayoritas, padahal pada beberapa kasus kelas yang menjadi perhatian adalah kelas-kelas minor, misalnya pada kasus pendeteksian intrusi. Kasus pendeteksian intrusi bertujuan untuk mendeteksi adanya intrusi di antara aliran data yang normal dan pada umumnya jumlah data yang mengandung intrusi lebih kecil daripada jumlah data normal. Jika CIP pada *dataset* ini tidak ditangani maka intrusi yang jumlahnya minoritas tidak akan bisa terdeteksi dan tujuan dari klasifikasi tidak tercapai.

```
Algoritma   :  WAVF
Input       :  dataset berukuran n x m yang akan diproses
               jumlah outlier, disebut sebagai k
Output      :  data outlier sebanyak k
```

1. Semua data dibaca dan dilabeli sebagai bukan outlier
2. Untuk semua obyek x_i, i = 1..n, kerjakan
   Untuk semua attribut a_h, h = 1..m. kerjakan
   Hitung probabilitas nilai atribut h pada objek x_i dan sebut sebagai $p(x_{ih})$
3. Untuk semua atribut a_h, h = 1..m, kerjakan
   R_h = maksimum(a_h) - minimum(a_h)
4. Untuk semua titik x_i, i = 1..n, kerjakan
   WAVF(x_i) = 0
   Untuk semua attribut a_h, h = 1..m. kerjakan
   WAVF(x_i) += $p(x_{ih})$ * R_h
5. Pilih k obyek yang memiliki nilai WAVF paling rendah sebagai outlier

Gambar 3.2 Pseudocode Algoritma WAVF (Rokhman dkk., 2016)

3.4.1 Solusi CIP

Secara garis besar ada dua solusi untuk CIP, yaitu solusi pada level data dan solusi pada level algoritma. Solusi pada level algoritma dilakukan dengan memodifikasi atau mengoptimasi *classifier* agar dapat bekerja dengan baik pada *imbalanced class* (Santoso dkk., 2017). Solusi pada level algoritma lebih membutuhkan pengetahuan khusus tentang ranah pengetahuan yang akan diklasifikasi (Amin dkk., 2016). Contoh solusi pada level algoritma diantaranya adalah algoritma-algoritma *cost sensitive learning*, algoritma berbasis *ensamble*.

Solusi pada level data tidak tergantung pada *classifier* tertentu yang ini merupakan keuntungan dari solusi ini. Solusi ini dilakukan dengan *re-sampling* untuk menyesuaikan distribusi data. Ide dasarnya adalah mengurangi jumlah data kelas mayoritas (under-sampling) atau meningkatkan jumlah data kelas minoritas (*over-sampling*) agar distribusi data menjadi seimbang(Amin dkk., 2016). Teknik *over-sampling* merupakan teknik yang paling banyak digunakan dibanding teknik *under-sampling* karena teknik *under-sampling* berpotensi menghilangkan informasi penting

yang ada pada kelas mayoritas(Santoso dkk., 2017). Ada 3 pendekatan utama dalam *re-sampling* yaitu (Amin dkk., 2016):

1. Metode *basic sampling*

   Basic *under-samping* dilakukan dengan cara menghapus sebagian data yang merupakan anggota dari kelas mayoritas sedangkan *basic over-sampling* dilakukan dengan cara menggandakan data yang merupakan anggota kelas dari kelas minoritas. Metode basic *under-sampling* mempunyai kekurangan yaitu peluang terbuangnya data yang penting dari kelas mayoritas sehingga dapat mengakibatkan performa *classifier* menurun. Berbeda dengan *basic under-sampling*, *basic over-sampling* tidak berdampak pada penurunan performa *classifier* tapi dapat mengakibatkan proses *training* menjadi lebih lama.

2. Metode *advanced sampling*

   Metode ini melibatkan pendekatan *data mining* atau statistik untuk melakukan *under-sampling* maupun *over-sampling*. Contoh algoritma yang merupakan *advanced sampling* yaitu SMOTE, ADASYN, MTDF.

3. Metode *random under/oversampling*

   Metode ini hampir sama dengan *basic sampling*, hanya saja pemilihan data yang akan dihapus/digandakan dipilih secara acak.

3.4.2 Algoritma SMOTE

Metode SMOTE (*Synthetic Minority Over-sampling Technique*) mengatasi CIP dengan cara melakukan *over-sampling* pada kelas minoritas. *Over-sampling* yang dilakukan tidak dengan menduplikasi data kelas minor, tapi dengan membuat data sintetis berdasar data kelas minor. Pembuatan data sintetis tidak dioperasikan pada *data space* tetapi dioperasikan pada *feature space*. Pembuatan data sintetis dilakukan dengan mengambil tiap data di kelas minor kemudian membuat data sintesis disepanjang garis segmen yang melibatkan beberapa atau seluruh k *nearest neighbour* dari kelas minor(Chawla dkk., 2002). Data sintetik memicu *classifier*

membuat area keputusan yang lebih luas dan kurang spesifik sehingga tidak menyebabkan *overfitting*(Chawla 2003).

SMOTE dapat dilakukan pada data numerik maupun kategorik atau pun kombinasi keduanya, numerik dan kategorik. SMOTE yang digunakan pada data numerik dan kategorik dikenal sebagai SMOTE Nominal *Continuous* (SMOTE-NC), sedangkan SMOTE yang digunakan untuk data kategorik dikenal dengan SMOTE Nominal atau disebut juga SMOTE-N.

Berbeda dengan SMOTE untuk data numerik, SMOTE-N mencari *nearest neighbour* dengan menggunakan *Value Difference Metric* (VDM). VDM melihat kesamaan nilai fitur terhadap keseluruhan vektor fitur. Sebuah matriks untuk menyimpan jarak antara nilai fitur yang sesuai degan keseluruhan vektor fitur. Perhitungan jarak dilakukan dengan menggunakan persamaan (3.1)(Chawla dkk., 2002).

$$\delta(V_1,V_2) = \sum_{i=1}^{n} \left| \frac{P_{1i}}{P_1} - \frac{P_{2i}}{P_2} \right|^{const} \qquad (3.1)$$

Pada persamaan (3.1), V_1 dan V_2 adalah dua nilai fitur dari 2 vektor fitur yang akan dihitung jaraknya. P_1 adalah jumlah kemunculan fitur V_1 pada keseluruhan data. n adalah jumlah kelas. P_{1i} adalah jumlah kemunculan fitur V_1 pada data dengan label kelas i. Hal yang sama juga berlaku untuk P_2 dan P_{2i}. const adalah nilai konstanta yang biasanya bernilai 1. Jarak dua buah vektor fitur diperoleh dengan menjumlahkan keseluruhan jarak vektor fitur.

Pembuatan data sintetis dilakukan dengan memilih nilai fitur yang paling banyak diantara nilai fitur yang lain untuk keseluruhan suatu vektor fitur dan k *nearest neigbour*-nya. Misal F1 adalah vektor fitur dengan nilai [A C F] dan F2 serta F3 adalah *nearest neigbour* yang digunakan. Nilai F2 adalah [A D E] dan nilai F3 adalah [B C E]. Nilai fitur yang paling banyak pada kolom pertama adalah A. Nilai fitur yang paling banyak pada kolom kedua adalah C dan nilai fitur yang paling banyak pada kolom

ketiga adalah E. Ketiga nilai fitur yang paling banyak tadi digabungkan menjadi sebuah data sintetis yaitu [A C E].

3.5 Principal Component Analysis (PCA)

PCA merupakan salah satu algoritma untuk melakukan ekstraksi fitur (*feature extraction*). Proses ekstraksi fitur yang dilakukan pada suatu data akan menghasilkan data dengan fitur baru. Fitur baru tersebut merupakan hasil pemetaan dari fitur asli (Motoda dan Liu, 1998). Ektraksi fitur merupakan salah satu bentuk proses pengurangan dimensi (*dimension reduction*) karena data yang dihasilkan dari proses tersebut akan memiliki dimensi yang lebih kecil dari data aslinya (Khalid dkk., 2014).

Pencarian pola pada data yang berdimensi rendah lebih mudah dilakukan daripada pencarian pola pada data yang berdimensi tinggi. Data yang berdimensi tinggi sulit untuk direpresentasikan ke dalam bentuk grafis sehingga pencarian pola pada data berdimensi tinggi menjadi tidak mudah. PCA dapat digunakan untuk mengatasi masalah pada data berdimensi tinggi tersebut (Smith, 2002).

PCA berusaha menjelaskan struktur korelasi kumpulan variabel prediksi menggunakan kumpulan kombinasi linier dari variabel tersebut. Kombinasi linier ini disebut komponen. Total variabilitas dari *dataset* yang dihasilkan oleh kumpulan m variabel seringkali dapat dijelaskan dengan kumpulan yang berukuran lebih kecil berupa k kombinasi linier dari variabel tersebut. Hal ini berarti informasi yang tersimpan dalam k komponen jumlahnya hampir sama dengan informasi yang berada pada m variabel asli meskipun ukuran k lebih kecil dari m. Jadi, PCA dapat mereduksi dimensi data tanpa harus kehilangan informasi dari data asli secara signifikan sehingga data asli dengan m variabel dapat digantikan dengan k komponen sehingga *dataset* menjadi berukuran lebih kecil. PCA diaplikasikan pada variabel prediksi saja dengan mengabaikan variabel target (Larose dan Larose, 2015).

PCA menggabungkan inti dari atribut-atribut data dengan membuat sebuah sekumpulan variabel yang lebih kecil. Data awal dapat diproyeksikan ke dalam

kumpulan variabel yang lebih kecil ini. PCA sering dapat mengungkapkan hubungan yang sebelumnya tidak pernah diketahui sehingga memungkinkan interpretasi yang baru terhadap data (Han dkk., 2012).

Langkah-langkah menganalisis data menggunakan metode PCA sebagai berikut (Smith, 2002):

1. Menyusun data ke dalam bentuk matriks m x l, dengan m adalah data yang digunakan dan l adalah banyaknya variabel data sehingga untuk data dengan 4 variabel (A, B, C, D) dan n baris data akan terbentuk matriks w berukuran m x 4 seperti pada persamaan (3.2).

$$w = \begin{bmatrix} A_1 & B_1 & C_1 & D_1 \\ A_2 & B_2 & C_2 & D_2 \\ \dots & \dots & \dots & \dots \\ A_n & B_n & C_n & D_n \end{bmatrix} \quad (3.2)$$

2. Mengurangi setiap data pada matriks dengan nilai rata-ratanya. Pencarian rata-rata dari setiap variabel menggunakan persamaan (3.3).

$$\overline{A} = \sum_{i=1}^{m} \frac{A_i}{m}; \overline{B} = \sum_{i=1}^{m} \frac{B_i}{m}; \overline{C} = \sum_{i=1}^{m} \frac{C_i}{m}; \overline{D} = \sum_{i=1}^{m} \frac{D_i}{m}; \quad (3.3)$$

Matriks w’ seperti pada persamaan (3.4) didapatkan setelah dilakukan pengurangan dengan menggunakan nilai rata-rata.

$$w' = \begin{bmatrix} A_1 - \overline{A} & B_1 - \overline{B} & C_1 - \overline{C} & D_1 - \overline{D} \\ A_2 - \overline{A} & B_2 - \overline{B} & C_2 - \overline{C} & D_2 - \overline{D} \\ \dots & \dots & \dots & \dots \\ A_n - \overline{A} & B_n - \overline{B} & C_n - \overline{C} & D_n - \overline{D} \end{bmatrix} \quad (3.4)$$

3. Menghitung matriks kovarian kvr dengan persamaan (3.5) dan (3.6).

$$\mathrm{cov}(A, B) = \sum_{i=1}^{m} \frac{(A_i - \overline{A})(B_i - \overline{B})}{m - 1} \quad (3.5)$$

$$\text{cov}(A,A) = \text{var}(A) = \sum_{i=1}^{m} \frac{(A_i - \overline{A})(A_i - \overline{A})}{m-1} \quad (3.6)$$

Matriks yang dihasilkan dari penelitian ini adalah matriks dimensi m x m sehingga terbentuk matriks kovarian pada persamaan (3.7).

$$kvr = \begin{bmatrix} \text{var}(A) & \text{var}(A,B) & \text{var}(A,C) & \text{var}(A,D) \\ \text{var}(B,A) & \text{var}(B) & \text{var}(B,C) & \text{var}(B,D) \\ \text{var}(C,A) & \text{var}(C,B) & \text{var}(C) & \text{var}(C,D) \\ \text{var}(D,A) & \text{var}(D,B) & \text{var}(D,C) & \text{var}(D) \end{bmatrix} \quad (3.7)$$

4. Menghitung vektor eigen dan nilai eigen dari matriks kovarian dengan menggunakan persamaan kvr.Q=λ.Q, dengan λ adalah nilai eigen dan Q adalah vektor eigen dari matriks kovarian kvr, nilai eigen dan vektor eigen dapat diselesaikan dengan menggunakan persamaan (3.8).

   $$\text{Det}(\lambda I - kvr)Q = 0 \quad (3.8)$$

   Matriks I adalah matriks identitas. Cacah vektor eigen dan nilai eigen yang dihasilkan akan sebanding dengan dimensi dari matriks kovarian kvr. Proporsi dari nilai eigen mengambarkan menggambarkan seberapa besar variansi data yang terjadi dalam eigen yang dapat mewakili variansi dalam data keseluruhan.
5. Memilih komponen dan membentuk vektor fitur. Vektor fitur adalah bagian dari vektor eigen yang dipilih untuk digunakan. Nilai eigen dari vektor eigen diurutkan dari yang terbesar sampai yang terkecil dalam rangka memilih vektor eigen. Urutan ini menggambarkan urutan signifikansi komponen. Sejumlah p komponen dengan nilai eigen terbesar dapat digunakan dan l — p komponen yang nilai eigen-nya kecil dapat diabaikan. Pengabaian komponen yang nilai eigen-nya kecil hanya akan menyebabkan hilangnya informasi dalam jumlah kecil pula. Namun pengabaian tersebut akan memberikan keuntungan berkurangnya jumlah dimensi data sehingga pada akhirnya diperoleh p dimensi dengan p < l.

   Untuk mendapatkan vektor fitur, kolom pertama vektor fitur diisikan vektor eigen

yang bersesuaian dengan nilai eigen terbesar pertama. Kolom kedua vektor fitur diisikan vektor eigen yang bersesuaian dengan nilai eigen terbesar kedua dan seterusnya sampai kolom ke-p. Bentuk matriks vektor fitur dapat ditunjukkan pada persamaan (3.9).

$$[F] = \left[eigen_1 \quad eigen_2 \quad \ldots \quad eigen_p \right] \qquad (3.9)$$

Menurunkan *dataset* baru dengan cara mengalikan matriks vektor fitur yang telah di-*transpose* dengan matriks data awal yang telah dikurangi nilai rata-ratanya. Operasi ini ditunjukkan dengan persamaan (3.10).

$$Final = [F]^T . [w]^T \qquad (3.10)$$

3.6 Diskritisasi Berbasis Entropi

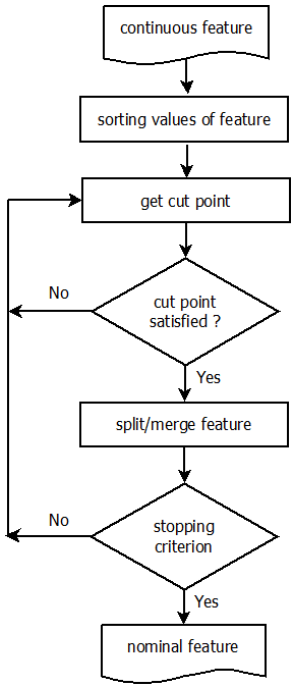
Diskritisasi adalah bagian dari tahap pra-pemrosesan pada *data mining* yang digunakan untuk mengubah fitur yang kontinu menjadi diskrit. Diskritisasi akan membagi variabel kontinu ke dalam kategori-kategori (Dash dkk., 2011). Tujuan dari diskritisasi adalah untuk mengurangi jumlah kemungkinan nilai atribut dari sebuah atribut kontinu dengan cara membaginya ke dalam beberapa interval nilai berdasarkan titik potong (*cut point*) yang telah ditentukan. Penentuan titik potong dapat dilakukan sendiri oleh pengguna atau mengunakan proses perhitungan (Hacibeyoglu dkk., 2011). Langkah-langkah dalam melakukan diskritisasi dapat dilihat pada Gambar 3.3.

Diskritisasi yang dilakukan pada variabel kontinu akan memberikan keuntungan-keuntungan sebagai berikut:

1. Variabel diskrit hanya memerlukan sedikit memori
2. Variabel diskrit lebih dapat memberi gambaran yang bermakna
3. Data menjadi lebih mudah untuk dipahami, digunakan, dan dijelaskan
4. Proses klasifikasi pada variabel diskrit lebih efisien dan lebih akurat

Ada beberapa klasifikasi diskritisasi. Berdasarkan waktu prosesnya, diskritisasi dibagi menjadi diskritisasi global dan diskritisasi lokal. Diskritisasi global dilakukan

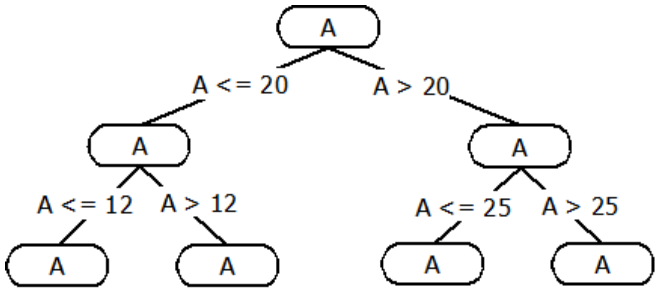
sebelum proses induksi, sedangkan diskritisasi lokal dilakukan saat proses induksi berlangsung. Beberapa penelitian menunjukkan bahwa diskritisasi global sering menghasilkan hasil yang lebih baik dibanding diskritisasi lokal.



Gambar 3.3 Tahapan proses diskritisasi (Hacibeyoglu dkk., 2011)

Berdasar penggunaan label data, diskritisasi dibagi menjadi 2 yaitu diskritisasi tak terawasi (*unsupervised discretization*) dan diskritisasi terawasi (*supervised discretization*). Diskritisasi tak terawasi tidak melibatkan label dari data dalam proses partisinya, sedangkan diskritisasi terawasi melibatkan label data dalam proses partisinya. Contoh algoritma diskritisasi tak terawasi adalah k-means clustering, equal width binning, dan equal frequency binning (Dash dkk., 2011).

Diskritisasi berbasis entropi (*entropy based discretization*) adalah salah satu jenis algoritma diskritisasi terawasi yang menggunakan mekanisme *top-down*. Tujuan dari algoritma ini adalah mendapatkan partisi yang mengandung baris data dari kelas yang sama sebanyak mungkin. Untuk dapat mencapai tujuan tersebut, entropi digunakan.



Gambar 3.4 Entropy-based discretization (Fayyad dan Irani, 1993)

Data yang akan didiskritisasi diurutkan terlebih dahulu kemudian nilai yang menjadi batas dari 2 kelas dijadikan sebagai kandidat titik potong. Masing-masing kandidat titik potong dihitung *information entropy*-nya dengan menggunakan persamaan (3.11). Kandidat titik potong dengan nilai *information entropy* terendah akan dipilih sebagai titik potong sehingga akan didapatkan dua buah partisi. Kedua partisi tersebut kemudian dipartisi lagi secara rekursif sampai kriteria pemberhentian tercapai (Fayyad dan Irani, 1993). Gambaran diskritisasi berdasar entropy dapat dilihat pada Gambar 3.4. Peluang algoritma ini untuk dapat meningkatkan akurasi cukup besar karena algoritma ini menggunakan informasi kelas dalam menentukan titik potong (Han dkk., 2012).

$$E(A,T;S) = \frac{|S_1|}{|S|} * Ent(S_1) + \frac{|S_2|}{|S|} * Ent(S_2) \qquad (3.11)$$

S pada persamaan (3.11) adalah data yang digunakan yang akan dipotong dengan titik potong T pada atribut A. S_1 dan S_2 adalah data dari dua interval yang menggunakan titik potong T. Ent adalah entropi dari data yang dihitung menggunakan persamaan (3.12). Pada persamaan tersebut, $p(C_i,D)$ adalah perbandingan data sampel yang ada dalam kelas C_i dan jumlah data dalam D. n adalah jumlah kelas yang terdapat dalam D.

$$Ent(D) = \sum_{i=1}^{n} - p(C_i, D) * \log_2(p(C_i, D)) \quad (3.12)$$

Ada banyak kriteria pemberhentian yang bisa digunakan. Kriteria pemberhentian yang digunakan pada penelitian ini adalah kriteria MDLP dan kriteria jumlah interval(*bin*). Jika kriteria pemberhentiannya menggunakan jumlah interval, proses partisi akan berhenti ketika jumlah interval yang diinginkan sudah tercapai. Salah satu teknik penentuan jumlah interval yang baik untuk pembelajaran terawasi adalah dengan menggunakan teknik Dougherty yang persamaannya dapat dilihat pada persamaan (3.13)(Alvarez dkk., 2013).

$$ival = \max\left(1, \lfloor 2\log_{10}(m) \rfloor\right) \quad (3.13)$$

Pada persamaan (3.13), ival adalah jumlah estimasi interval yang merupakan nilai pembulatan ke bawah dari log m. m adalah jumlah data yang akan didiskritisasi. Jika nilai log m kurang dari 1, maka jumlah intervalnya adalah 1.

Fayyad dan Irani(1993) mengusulkan sebuah kriteria pemberhentian dengan menggunakan MDLP (Minimum Desription Length Principle) yang akan menghentikan proses partisi ketika Gain(S, A) < δ. Proses partisi yang memenuhi kriteria tersebut akan ditolak. Nilai Gain(S, A) diperoleh dari persamaan (3.14) sedangkan nilai δ didapatkan dari persamaan (3.15) dan (3.16).

$$Gain(S, A) = Ent(S) - \sum_{i=1}^{k} \frac{|S_i|}{|S|} * Ent(S_i) \quad (3.14)$$

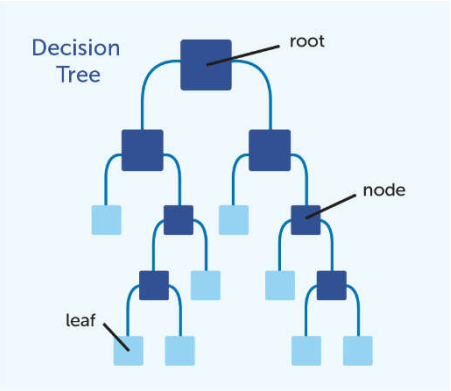
$$\delta = \frac{\log_2(m-1) + \Delta(A, T; S)}{m} \quad (3.15)$$

$$\Delta(A, T; S) = \log_2(3^n - 2) - \left[n * Ent(S) - n_1 * Ent(S_1) - n_2 * Ent(S_2)\right] \quad (3.16)$$

Pada persamaan (3.14), k adalah jumlah partisi. Pada persamaan (3.15), m adalah jumlah data dalam S. Pada persamaan (3.16), n adalah jumlah kelas pada himpunan S, n_i adalah jumlah kelas pada himpunan S_i.

3.7 Pohon Keputusan

Pohon keputusan adalah sebuah kumpulan dari *node-node* keputusan yang dihubungkan dengan cabang (*branch*), diperluas ke bawah dari *root* (akar) *node* sampai pada *leaf* (daun) *node*. Dimulai dari *root node* yang ditempatkan di bagian atas pohon keputusan, tiap-tiap atribut diuji pada tiap *node-node* keputusan. Tiap-tiap atribut memiliki kemungkinan untuk menghasilkan cabang. Setiap cabang kemudian mengarah baik ke *node* keputusan lain atau ke sebuah *leaf node. Leaf node* adalah *node* yang berada di paling ujung. Gambaran pohon keputusan dapat dilihat pada gambar 3.5 (Larose, 2014). Dari gambaran di atas, dapat disimpulkan bahwa ada 3 jenis node, yaitu *root node, internal node*, dan *leaf node*. *Root node* tidak memiliki masukan dan dapat memiliki nol atau lebih keluaran. *Internal node* hanya memiliki satu masukan dan dua atau lebih keluaran. *Leaf node* hanya memiliki satu masukan dan tidak memiliki keluaran. Setiap internal node berisi kondisi pengujian atribut untuk memecah baris data yang memiliki karakteristik berbeda. Setiap *leaf node* diberikan sebuah label kelas (Tan dkk., 2005).



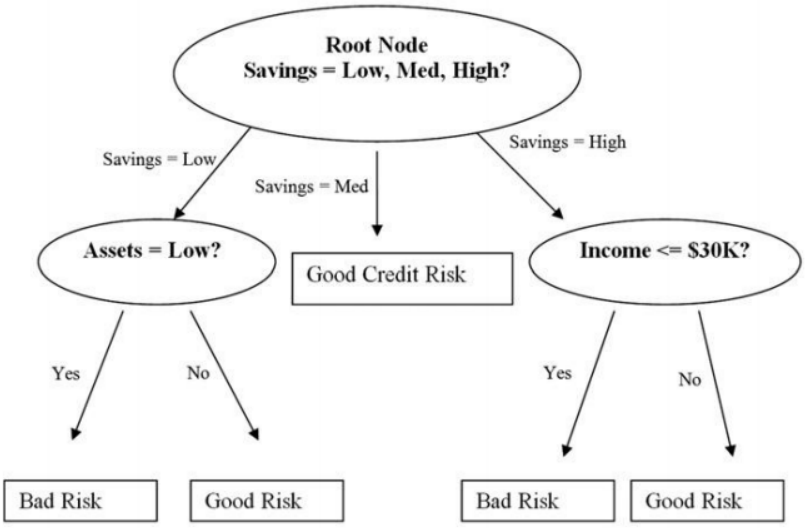
Gambar 3.5 Struktur pohon keputusan (Grisanti, 2016)

Sebuah pohon keputusan adalah serangkaian pertanyaan yang disusun secara sistematis sehingga masing-masing pertanyaan tentang atribut dijawab berdasarkan nilai dari atribut tersebut. Pada gambar 3.6 yang menggambarkan tentang pohon keputusan untuk penentuan resiko kredit, pertanyaan tentang *savings* (nilai

tabungan) dijawab berdasar nilai dari atribut *savings* sehingga muncul 3 cabang (Wu dan Kumar, 2009).

Pada tahun 2006, diadakan voting tentang algoritma data mining yang paling populer terhadap para peneliti di bidang data mining. Dipilih 10 algoritma terpopuler dari 18 algoritma yang menjadi kandidat. Dua dari 10 algoritma tersebut adalah algoritma berbasis pohon keputusan yaitu C4.5 dan CART (Wu dan Kumar, 2009).

Pohon keputusan cukup populer dalam *data mining* karena pembangunan pohon keputusan tidak membutuhkan domain pengetahuan atau pengaturan parameter sehingga cocok untuk eksplorasi penemuan pengetahuan (*exploratory knowledge discovery*). Pohon keputusan juga dapat menangani data multi dimensi. Langkah pembelajaran dan klasifikasi pada pohon keputusan sederhana dan cepat (Han dkk., 2012) dan beberapa kelebihan-kelebihan lain yang membuat algoritma pohon keputusan menjadi populer dan digunakan di berbagai bidang.



Gambar 3.6 Pohon keputusan sederhana (Larose, 2014)

Menurut Larose (2014), penggunaan pohon keputusan memerlukan beberapa persyaratan yang harus terpenuhi, yaitu:

1. Algoritma pohon keputusan merupakan jenis pembelajaran terawasi sehingga data pelatihan harus memiliki variabel target.

2. Data pelatihan harus banyak dan bervariasi karena pohon keputusan belajar dari contoh. Jika contoh kurang sistematis untuk sebagian baris yang bisa didefinisikan, klasifikasi, dan prediksi untuk bagian tersebut akan bermasalah atau tidak mungkin dilakukan.
3. Nilai variabel target harus bersifat diskrit. Pohon keputusan tidak dapat diaplikasikan pada variabel target yang bersifat kontinu.

Banyak algoritma yang dapat dipakai dalam pembentukan pohon keputusan antara lain ID3, CART, C4.5, CHAID, QUEST (Rokach dan Maimon, 2014).

3.7.1 Kelebihan dan Kekurangan

Menurut Rokach dan Maimon (2014), algoritma pohon keputusan memiliki kelebihan-kelebihan berikut ini:

1. Pohon keputusan menggambarkan pengetahuan dalam bentuk pohon sehingga sangat intuitif dan mudah untuk dipahami oleh manusia. Pohon keputusan juga dapat dikonversi ke dalam sekumpulan aturan.
2. Pohon keputusan dapat menangani input yang berupa nilai nominal muapun numerik.
3. Pohon keputusan cukup dapat diandalkan untuk mengklasifikasikan nilai diskrit.
4. Pohon keputusan dapat menangani *dataset* yang di dalamnya mengandung kesalahan.
5. Pohon keputusan dapat digunakan pada *dataset* yang memiliki nilai yang hilang (*missing value).*

Selain memiliki kelebihan, algoritma pohon keputusan juga memiliki kekurangan-kekurangan berikut ini:

1. Beberapa algoritma seperti ID3 dan C4.5 mempersyaratkan atribut target hanya boleh bernilai diskrit.
2. Pohon keputusan menggunakan metode *divide and conquer* sehingga punya kecenderungan akan berjalan baik jika atribut yang terkait secara relevan tidak

terlalu banyak dan akan bekerja kurang baik jika banyak terdapat interaksi yang kompleks.

3. Karakteristik serakah (*greedy*) dari pohon keputusan menyebabkan kerugian lain yang harus ditunjukkan. *Over-sensitivity* terhadap data pelatihan, atribut yang tidak relevan, dan *noise* membuat pohon keputusan menjadi tidak stabil. Sebuah perubahan kecil pada pemisahan yang dekat dengan *root* akan mengubah seluruh *subtree* di bawahnya. Pohon keputusan dapat memilih atribut yang bukan merupakan atribut terbaik dikarenakan variasi yang kecil pada data pelatihan,
4. Usaha yang diperlukan untuk menangani *missing value* dianggap sebagai suatu kelemahan meskipun kemampuan untuk menangani *missing value* dianggap sebagai suatu kelebihan. Algoritma pohon keputusan akan menggunakan mekanisme tertentu untuk menangani *missing values*. Untuk tujuan mengurangi kemunculan tes pada *missing values*, C4.5 mengabaikan data yang *missing value* ketika melakukan perhitungan *information gain*. Data yang memiliki atribut yang *missing values* kemudian dijadikan 1 kelompok untuk dijadikan sebagai *subtree*.

3.7.2 Algoritma C4.5

C4.5 adalah algoritma untuk menyelesaikan masalah klasifikasi dalam *machine learning* dan *data mining*. Algoritma yang dibuat oleh J.Ross Quinlan ini termasuk dalam jenis pohon keputusan dengan dengan model pembelajaran terawasi. Algoritma C4.5 menggunakan konsep *information gain* atau *entropy reduction* untuk memilih kriteria pemisahan (*split*) yang optimal (Wu dan Kumar, 2009).

Algoritma C4.5 termasuk jenis pohon keputusan yang pembangunannya menggunakan model *top-down*. Langkah-langkah pembangunan pohon keputusan ini sebagai berikut (Ye, 2014):

1. Memilih *root node*
2. Menerapkan metode pemilihan pemisah (*split selection*) untuk memilih kriteria pemisah (*split criterion*) yang terbaik dan membagi data pelatihan berdasar

node/atribut yang terpilih. Algoritma C4.5 dapat menggunakan kriteria *information gain* (atau disebut *gain*) maupun *gain ratio* tetapi *default* kriterianya adalah *gain ratio* (Wu dan Kumar, 2009). *Information gain*, yang rumusnya terdapat pada persamaan (3.14), adalah selisih *information entropy* sebelum dilakukan pemisahan dan sesudah dilakukan pemisahan. *Information gain* bias terhadap atribut bernilai banyak (*multivalued attribute*). Untuk mengatasi hal tersebut, C4.5 menggunakan *gain ratio* yang merupakan normalisasi dari nilai *gain* (Han dkk., 2012). Rumus *gain ratio* dapat dilihat pada persamaan (3.17), sedangkan rumus Gain(S,A) dan Ent(S_i) dapat dilihat pada persamaan (3.14) dan (3.12). Pada persamaan (3.17), k adalah jumlah partisi dalam S.

$$GainRatio(S,A) = \frac{Gain(S,A)}{\sum_{i=1}^{k} Ent(S_i)} \tag{3.17}$$

3. Cek apakah kriteria pemberhentian sudah terpenuhi atau belum. Jika sudah terpenuhi, pembangunan pohon keputusan akan dihentikan. Jika tidak terpenuhi, ulangi kembali langkah ke-2 dengan memilih sebuah *node* untuk pemisahan.

Pemberhentian dengan menggunakan kriteria pemberhentian berdasar homogenitas data akan dilakukan ketika tiap *leaf node* memiliki data yang homogen. Data disebut homogen ketika seluruh data pada *leaf node* tersebut mempunyai nilai target yang sama.

Terkadang homogenitas data pada *leaf node* sulit untuk tercapai karena *noise* pada data yang diklasifikasi. Pada kasus tersebut, pemberhentian dilakukan ketika homogenitas data lebih kecil dari nilai ambang batas (*threshold*) tertentu misal entropy(D) < 01 (Ye, 2014).

Algoritma C4.5 menangani atribut kontinu dengan membagi nilai atribut menjadi dua bagian berdasar suatu nilai ambang batas. Nilai ambang batas dicari yang terbaik yaitu nilai ambang batas yang dapat memaksimalkan *gain ratio*. Semua nilai

di atas nilai ambang batas dimasukkan ke dalam bagian pertama, dan nilai lainnya dimasukkan ke dalam bagian kedua (Wu dan Kumar, 2009).

3.8 Cross Validation

K-fold cross-validation merupakan salah satu dari variasi teknik pengujian *cross-validation*. *K-fold cross-validation* dilakukan dengan membagi data menjadi set pelatihan dan set pengujian. Data asli kemudian dibagi menjadi k bagian yang disebut *fold*. Masing-masing *fold* memiliki ukuran yang hampir sama. Pelatihan dan pengujian dilakukan sebanyak k kali. Pada iterasi ke-1, *fold* ke-1 digunakan untuk pengujian dan *fold* ke-2 sampai dengan *fold* ke-k digunakan untuk pelatihan. Pada iterasi ke-2, *fold* ke-2 digunakan untuk pengujian. *Fold* ke-1, *fold* ke-3, dan seterusnya digunakan untuk pelatihan. Proses ini dilakukan sebanyak k kali sehingga semua *fold* pernah digunakan untuk pelatihan tepat sebanyak 1 kali dan inilah keuntungan dari *k-fold cross-validation* (Larose, 2014). Gambaran prosesnya dapat dilihat pada Tabel 3.2. Perkiraan akurasi dengan menggunakan *k-fold cross-validation* adalah keseluruhan klasifikasi yang benar dari k iterasi, dibagi jumlah baris pada data awal (Han dkk., 2012).

Tabel 3.2 Gambaran 4-fold cross-validation

Iterasi	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**
Iterasi ke-1	Lat	Lat	Lat	Uji
Iterasi ke-2	Lat	Lat	Uji	Lat
Iterasi ke-3	Lat	Uji	Lat	Lat
Iterasi ke-4	Uji	Lat	Lat	Lat

BAB IV
ANALISIS DAN PERANCANGAN

4.1 Analisis Sistem

4.1.1 Data Pemetaan Pegawai

Data yang digunakan dalam penelitian ini merupakan hasil dari tes psikologi untuk keperluan pemetaan pegawai. Pegawai yang dites akan dinilai 14 aspek psikologisnya. Keempat belas aspek psikologis tersebut adalah

1. Potensi kecerdasan (pk)
2. Daya konseptual (dk)
3. Daya analisis (da)
4. Fleksibilitas berpikir (fb)
5. Kemampuan numerikal (kn)
6. Sistematika kerja (sk)
7. Hasrat berprestasi (hb)
8. Inisiatif (if)
9. Stabilitas emosi (se)
10. Kepercayaan diri (kd)
11. Penyesuaian diri (pd)
12. Kerjasama (ks)
13. Toleransi terhadap stress (ts)
14. Kepemimpinan (kp)

Nilai dari aspek-aspek psikologis tersebut mulai dari 0 sampai dengan 5 dan dimungkinkan penambahan + dan – untuk masing-masing tingkatan nilai, contoh 3-, 3, dan 3+. Nilai 3 berarti aspek psikologis pegawai tersebut tepat memenuhi semua indikator nilai 3. Nilai 3+ berarti indikator aspek psikologis pegawai tersebut dominan pada nilai 3 tetapi ada beberapa indikator yang memenuhi indikator nilai 4. Nilai 3-

berarti indikator aspek psikologis pegawai tersebut dominan pada nilai 3 tetapi ada beberapa indikator yang nilai 3 yang belum terpenuhi.

Selain menilai aspek psikologis pegawai, assessor juga memberikan rekomendasi pelatihan pengembangan diri berdasar nilai aspek-aspek psikologis seorang pegawai. Seorang pegawai dimungkinkan untuk mendapatkan lebih dari rekomendasi pelatihan pengembangan diri seperti terlihat dalam contoh pada Tabel 4.1. Jenis-jenis pelatihan pengembangan diri yang digunakan sebagai berikut:

1. *Achieve Motivation Training*
2. *Effective Communication*
3. *Human Skill Development*
4. *Personnel Effectiveness*
5. *Readiness To Change*
6. *Team Building*

Data pemetaan pegawai yang digunakan berasal dari data BKN Pusat, Kanreg I BKN Yogyakarta, Kanreg III BKN Bandung, dan Kanreg VI BKN Medan yang keseluruhannya berjumlah 474 data. Data-data tersebut dipindahkan dari bentuk laporan ke dalam basis data.

4.1.2 Deskripsi Sistem

Sistem yang dibangun ini adalah suatu sistem yang digunakan untuk menentukan rekomendasi pelatihan pengembangan diri bagi pegawai negeri sipil berdasarkan hasil tes pemetaan pegawai. Hasil tes pemetaan pegawai berupa nilai 14 aspek psikologis yang sudah dijelaskan pada sub bab 4.1.1. Rekomendasi atas sebuah pelatihan diperoleh dengan menggunakan 3 algoritma dan masing-masing algoritma tersebut akan menghasilkan aturan-aturan. Aturan yang digunakan untuk mendapatkan rekomendasi atas sebuah pelatihan adalah aturan yang memiliki kinerja paling baik. Penelitian ini menggunakan 6 jenis pelatihan dan 1 jenis pelatihan akan

memiliki 1 aturan sehingga akan didapatkan 6 buah aturan yang digunakan untuk mendapatkan rekomendasi pelatihan.

Sistem ini dibagi menjadi 2 bagian yaitu bagian *Back-end* dan bagian *Front-end*. Bagian *Back-end* merupakan berfungsi untuk membangun model rekomendasi pelatihan, sedangkan bagian *Front-end* menyediakan sarana interaksi dengan pengguna untuk mendapatkan rekomendasi pelatihan dari model yang telah terbentuk. Rancangan sistem secara sederhana dapat dilihat pada Gambar 4.1.

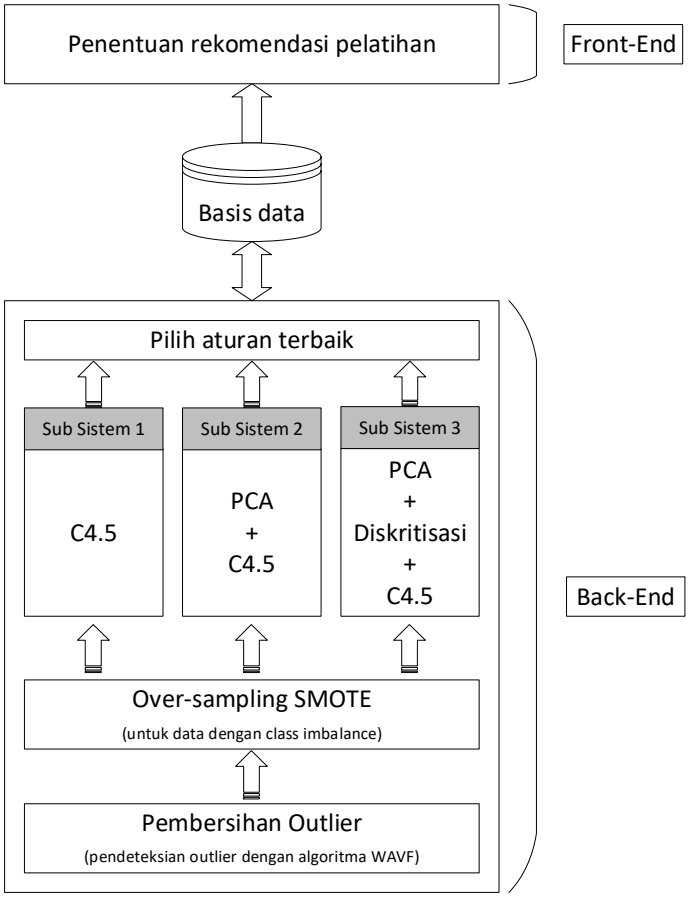
Tabel 4.1 Contoh data hasil pemetaan pegawai

Pegawai	AGUS	BUDI
Potensi kecerdasan	2+	3-
Daya konseptual	2	2+
Daya analisis	2-	2+
Fleksibilitas berpikir	3	2+
Stabilitas memori	2	3
Kemampuan numerikal	3	3+
Sistematika kerja	3+	2-
Hasrat berprestasi	3	2-
Inisiatif	3-	3+
Stabilitas emosi	3-	2+
Kepercayaan diri	3+	3-
Penyesuaian diri	3+	3-
Kerjasama	3	3-
Toleransi terhadap stress	3+	3+
Kepemimpinan	2-	1
Rekomendasi pelatihan	*Personal effectiveness, Human skill development, Team building*	*Personal effectiveness, Achievement Motivation Training*

Gambar 4.1 menunjukkan bahwa bagian *Back-end* terdiri dari 3 sub sistem yang masing-masing menggunakan metode yang berbeda. Ketiga sub sistem tersebut sebagai berikut:

1. Sub sistem 1 menggunakan algoritma C4.5 saja. Sub sistem ini menerapkan algoritma C4.5 untuk menentukan rekomendasi pelatihan bagi pegawai.

2. Sub sistem 2 menggunakan algoritma PCA dan C4.5. Sub sistem ini melakukan pra pemrosesan data dengan algoritma PCA kemudian menentukan rekomendasi pelatihan dengan algoritma C4.5.
3. Sub sistem 3 menggunakan algoritma PCA, diskritisasi, dan C4.5. Sub sistem ini melakukan pra pemrosesan data dengan algoritma PCA dan diskritisasi kemudian menentukan rekomendasi pelatihan dengan algoritma C4.5.



Gambar 4.1 Rancangan sistem

Sebelum data diproses oleh ketiga sub sistem, data dibersihkan dari *outlier* kemudian di-*over-sampling*. *Outlier* dideteksi dengan menggunakan algoritma WAVF dan proses *over-sampling* dilakukan dengan menggunakan algoritma SMOTE. Data yang di-*over-sampling* hanyalah data yang mengalami CIP saja.

Pada bagian ini juga terdapat fungsi pemilihan aturan-aturan yang memiliki kinerja terbaik. Pemilihan dilakukan dengan membandingkan performa masing-masing sub sistem. Aturan-aturan dari sub sistem yang memiliki performa terbaik akan disimpan ke dalam basis data.

Bagian *Front-end* akan menentukan rekomendasi pelatihan berdasar data yang dimasukkan dengan menggunakan aturan yang terpilih untuk masing-masing pelatihan. Apabila aturan yang terbaik berasal dari sub sistem PCA dan C4.5, data masukan akan ditransformasi dulu dengan PCA sebelum diklasifikasi dengan aturan hasil algoritma C4.5. Data masukan akan ditransformasi dengan PCA kemudian didiskritisasi sebelum diklasifikasi dengan aturan hasil algoritma C4.5, apabila aturan yang terbaik berasal dari sub sistem PCA, diskritisasi, dan C4.5. Data akan langsung diklasifikasi dengan algoritma C4.5, apabila aturan terbaik berasal dari sub sistem C4.5.

Bagian *Back-end* dan bagian *Front-end* dihubungkan dengan basis data. Aturan terbaik yang didapatkan dari proses pada bagian *Back-end* akan disimpan ke dalam basis data. Bagian *Front-end* akan membaca aturan yang tersimpan dalam basis data tersebut untuk dapat menghasilkan rekomendasi pelatihan.

4.2 Perancangan Bagian Back-End

4.2.1 Penyimpanan Data Pemetaan Pegawai

Nilai aspek psikologis yang terdapat dalam data pemetaan pegawai bertipe nominal. Oleh karena itu, data tersebut harus dikodekan ke dalam tipe numerik sebelum dilakukan proses selanjutnya. Aturan pengkodeannya dapat dilihat pada Tabel 4.2.

Untuk menggambarkan hubungan antara data pemetaan dan rekomendasi pelatihannya, masing-masing data pelatihan disimpan ke dalam sebuah tabel sehingga terdapat 6 buah tabel, yaitu tabel pelatihan AMT, tabel pelatihan *Effective Communication Skill*, tabel pelatihan *Human Skill Improvement*, tabel pelatihan

Personnel Effectiveness, tabel pelatihan *Readiness to Change*, dan tabel pelatihan *Team Building* . Masing-masing data pelatihan terdiri dari 2 kelas, yaitu kelas ya dan kelas tdk(tidak). Contoh penyimpanan data pelatihan AMT ke dalam basis data dapat dilihat pada Tabel 4.3.

Tabel 4.2 Pengkodean nilai pemetaan pegawai

1		2		3		4		5		0
Nilai	**Kode**	**Nilai**	**Kode**	**Nilai**	**Kode**	**Nilai**	**Kode**	**Nilai**	**Kode**	**Kode**
1-	1.25	2-	2.25	3-	3.25	4-	4.25	5-	5.25	0.5
1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	
1+	1.75	2+	2.75	3+	3.75	4+	4.75	5+	5.75	

Tabel 4.3 Contoh data pelatihan AMT di dalam basis data

id	pk	dk	da	fb	kn	sk	hp	in	se	kd	pd	ks	ts	kp	Training AMT
1	3-	2+	2+	2+	3	3+	2-	2-	3+	2+	3-	3-	3+	1	Ya
2	2-	2-	2	3-	2-	3-	3+	3+	3+	3-	3+	3+	3+	2+	Tdk
3	3-	2+	2+	2+	2+	3+	2-	2-	3+	2	2+	2+	2+	2-	Ya

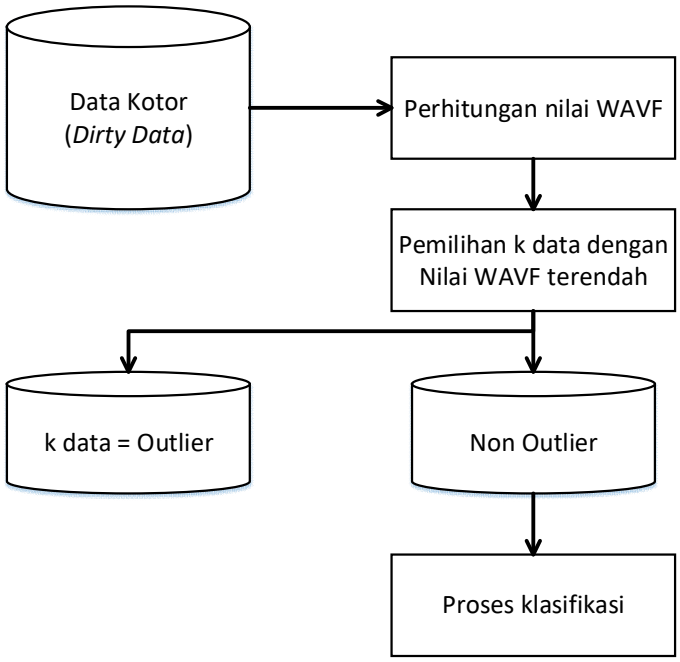
Tabel 4.3 menggambarkan penyimpanan data pelatihan AMT. Kolom id menggambarkan nomor pegawai. Kolom pk, dk, da, dan kolom-kolom selanjutnya sampai kolom kp merupakan nilai aspek psikologis pegawai yang aturan penyingkatannya mengacu pada bab 4.1.1. Pada baris pertama, data dengan ID adalah 1 memiliki kelas ya. Maksud dari data tersebut adalah pegawai dengan nilai aspek psikologis pada baris pertama perlu mengikuti pelatihan AMT. Baris kedua yang nilai kelasnya adalah tdk sehingga pegawai pada data baris kedua tidak perlu untuk mengikuti pelatihan AMT.

4.2.2 Rancangan Pra-pemrosesan Data

Sebelum dilakukan proses selanjutnya, data dilakukan pra-pemrosesan terlebih dahulu. Pra-pemrosesan ditujukan untuk menghilangkan *outlier* dan menangani ketidakseimbangan kelas pada data.

Penanganan Outlier

Data-data yang ada pada *real-world* adalah *dirty data* yang sangat dimungkinkan di dalamnya terdapat *outlier*. Begitu juga data pemetaan pegawai yang digunakan pada penelitian ini. *Outlier* yang terdapat pada data pemetaan pegawai perlu dihapus terlebih dahulu untuk meningkatkan performa klasifikasi yang dihasilkan. Untuk mengenali data-data yang merupakan *outlier*, algoritma WAVF digunakan. Gambaran penggunaan algoritma WAVF untuk menangani *outlier* pada penelitian ini dapat dilihat pada Gambar 4.2.



Gambar 4.2 Alur penanganan outlier dengan algoritma WAVF

Tabel 4.4 Contoh data yang akan dicari outlier-nya

id	pk	dk	da	fb	kn	sk	hb	if	se	kd	pd	ks	ts	kp
158	1.5	1.75	1.75	2.25	1.5	2.25	2.25	1.75	2.25	2.25	2.5	2.5	2.5	1.5
189	2.5	1.5	1.5	1.5	1.5	2.5	1.5	1.5	2.5	2.25	2.25	2.25	1.5	1.5
457	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	2.5	2.5	2.5	1.5	1.5	1.5
454	2.75	1.5	1.5	1.5	1.5	1.5	1.5	1.5	2.5	1.5	1.5	1.5	2.5	1.5
445	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	2.5	1.5	1.5	2.5	2.5	1.5

Penanganan *outlier* diawali dengan penghitungan nilai WAVF seperti terlihat pada Gambar 4.2. Data yang ada pada Tabel 4.4 dihitung nilai probabilitas atributnya.

Data aspek pk untuk id 158 diperoleh dengan menghitung frekuensi kemunculan nilai 1,5 pada kolom pk di keseluruhan baris. Kemunculan nilai 1,5 adalah 3 kali dan jumlah keseluruhan baris adalah 5 sehingga diperoleh nilai atribut 3/5. Perhitungan tersebut diulangi untuk keseluruhan atribut pada tiap baris sehingga diperoleh hasil seperti nampak pada Tabel 4.5.

Tabel 4.5 Hasil perhitungan probabilitas atribut

id	pk	dk	da	Fb	kn	sk	Hb	if	se	kd	pd	ks	ts	kp
158	3/5	1/5	1/5	1/5	5/5	1/5	1/5	1/5	1/5	2/5	2/5	2/5	3/5	5/5
189	1/5	4/5	4/5	4/5	5/5	1/5	4/5	4/5	4/5	2/5	1/5	1/5	2/5	5/5
457	3/5	4/5	4/5	4/5	5/5	3/5	4/5	4/5	4/5	1/5	2/5	2/5	2/5	5/5
454	1/5	4/5	4/5	4/5	5/5	3/5	4/5	4/5	4/5	2/5	2/5	2/5	3/5	5/5
445	3/5	4/5	4/5	4/5	5/5	3/5	4/5	4/5	4/5	2/5	2/5	2/5	3/5	5/5

Setelah nilai probabilitas atribut didapatkan, nilai *range* untuk masing-masing atribut dihitung. Sebagai contoh, nilai maksimal untuk atribut pk adalah 3/5 dan nilai minimal atribut pk adalah 1/5. Nilai *range* atribut pk adalah pengurangan nilai maksimal dari atribut tersebut dikurangi nilai minimalnya seperti pada perhitungan di bawah ini.

Range(pk) = 3/5 – 1/5 = 2/5

Range(dk) = 4/5 – 1/5 = 3/5

Range(da) = 4/5 – 1/5 = 3/5

Range(fb) = 4/5 – 1/5 = 3/5

Range(kn) = 4/5 – 1/5 = 3/5

Range(sk) = 5/5 – 5/5 = 0

Range(hb) = 4/5 – 1/5 = 3/5

Range(if) = 4/5 – 1/5 = 3/5

Range(se) = 4/5 – 1/5 = 3/5

Range(kd) = 2/5 – 1/5 = 1/5

Range(pd) = 2/5 – 1/5 = 1/5

Range(ks) = 2/5 – 1/5 = 1/5

Range(ts) = 3/5 – 2/5 = 1/5

Range(kp) = 5/5 – 5/5 = 0

Pada saat perhitungan nilai WAVF, nilai *range* untuk masing-masing atribut tersebut digunakan sebagai bobot pengali untuk masing-masing nilai probabilitas atribut. Hasil akhir perhitungan nilai WAVF dapat dilihat pada Tabel 4.6 dan berikut ini contoh perhitungan nilai WAVF.

WAVF(158) = (2/5*3/5) + (3/5*1/5) + (3/5*1/5) + (3/5*1/5) + (3/5*1/5) + (0*1) + (3/5*1/5) + (3/5*1/5) + (3/5*1/5) + (1/5*2/5) + (1/5*2/5) + (1/5*2/5) + (1/5*3/5) + (0*1) = 1.92

Tabel 4.6 Hasil perhitungan nilai WAVF

id	Nilai WAVF
158	1,92
189	3,80
457	4,00
454	3,92
445	4,08

Pada Tabel 4.6, nilai WAVF yang paling kecil adalah data dengan id 158. Selain itu, nilai WAVF untuk data dengan id 158 ini terpisah jauh dari data yang lain. Data yang lain nilai WAVF-nya mendekati nilai 4 sedangkan nilai WAVF data dengan id 158 mendekati nilai 2. Berdasar hal tersebut dapat dinyatakan *outlier* dari data tersebut adalah data dengan id 158.

Over-sampling

Tabel 4.7 menunjukkan bahwa data pelatihan yang tidak mengalami *class imbalance problem* (CIP) hanya pelatihan *Achievement Motivation Training* dan *Team Building*. Dari 4 data pelatihan yang mengalami CIP, ada 3 data pelatihan yang kelas minornya adalah kelas Ya dan ada 1 pelatihan yang kelas minornya adalah kelas Tdk, yaitu pelatihan *Personnel Effectiveness*. Data pelatihan yang kelas minornya adalah kelas Ya yaitu pelatihan *Effective Communication Skill, Human Skill Improvement*, dan *Readiness to Change* dengan persentase hanya kisaran belasan persen. Kondisi CIP

tersebut akan menyebabkan peluang model untuk merekomendasikan ketiga pelatihan tersebut menjadi kecil sedangkan model untuk pelatihan *Personnel Effectivenes* berpeluang kecil untuk tidak merekomendasi pelatihan tersebut. Algoritma C4.5 yang digunakan untuk klasifikasi pada penelitian ini juga memiliki kendala dengan CIP karena pengukuran entropy hanya dapat berjalan baik ketika datanya seimbang, seluruh kelas memiliki proporsi yang sama (Kishners dkk., 2016). Oleh karena itu, pada keempat data pelatihan tersebut perlu dilakukan *over-sampling*.

Tabel 4.7 Rekap data pelatihan disertai Imbalance Ratio (IR)

Nama Pelatihan	Jumlah Data (Total = 451)		% Kelas Minor	IR
	Kelas Ya	Kelas Tdk		
Achieve Motivation T.	200	251	44.35	0.80
Effective Comuncation	56	395	12.42	0.14
Human Skill Improvement	59	392	13.08	0.15
Personnel Effectiveness	317	134	29.71	2.37
Readiness to Change	40	411	8.87	0.10
Team Building	234	217	48.11	1.08

Proses *over-sampling* akan dilakukan dengan menggunakan algoritma SMOTE. Karena data yang digunakan pada penelitian ini adalah data kategorik, algoritma SMOTE yang digunakan adalah algoritma SMOTE untuk data nominal atau disebut juga SMOTE-N. Besaran persentase proses sampling untuk keempat data pelatihan dapat dilihat pada Tabel 4.8.

Tabel 4.8 Daftar persentase sampling per data pelatihan

Nama Pelatihan	Kelas Mayor		Kelas Minor			
	Label	Jumlah	Label	Jumlah	%	% Over-sampling
Effective Comuncation	Tdk	395	Ya	56	12.42	500
Human Skill Improvement	Tdk	392	Ya	59	13.08	500
Personnel Effectiveness	Ya	317	Tdk	134	29.71	100
Readiness to Change	Tdk	411	Ya	40	8.87	900

Proses over-sampling dilakukan menggunakan program WEKA versi 3.8 (www.cs.waikato.ac.nz/ml/weka/) dengan cara merubah format data pelatihan yang sudah bersih dari *outlier* ke dalam format ARFF. Data dalam bentuk ARFF diproses dengan algoritma SMOTE sesuai persentase yang telah ditentukan. Hasil dari proses *over-sampling*-nya kemudian dimasukkan ke dalam basis data.

4.2.3 Rancangan Proses Pembuatan Aturan

Pembuatan aturan melibatkan 3 buah sub sistem yang salah satunya adalah sub sistem PCA, diskritisasi, dan C4.5. Sub sistem tersebut menggunakan algoritma yang paling banyak jika dibandingkan dengan 2 sub sistem yang lain dan sub sistem tersebut menggunakan pendekatan baru yang diusulkan dalam penelitian ini.

Proses dengan Algoritma PCA

Proses ini bertujuan untuk mendapatkan fitur baru dari data yang ada atau disebut juga proses ekstraksi fitur. Langkah-langkah untuk mendapatkan *principal component* (komponen utama) dari suatu data sebagai berikut:

1. Data pemetaan pegawai disusun ke dalam bentuk matriks. Misal untuk 40 data akan diperoleh matriks berukuran 40x14 seperti terlihat pada Gambar 4.3.

pk	dk	da	fb	kn	sk	hb	if	se	kd	pd	ks	ts	kp
2.5	2.75	2.5	2.75	2.75	3.5	2.5	2.5	2.75	2.5	2.5	3.5	2.75	2.5
2.75	2.5	2.5	2.5	2.75	3.5	3.25	2.75	3.25	2.75	2.5	2.75	3.5	2.5
3.5	2.75	2.75	2.75	2.5	3.5	2.75	2.75	3.5	3.5	3.25	3.25	3.5	3.25
3.75	3.5	3.5	3.5	3.5	3.25	2.75	3.25	3.25	3.5	3.5	3.5	3.25	2.5
4.5	3.5	3.5	3.5	3.5	3.5	3.5	3.75	3.5	3.5	3.5	3.5	3.5	3.25
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2.25	1.5	1.5	1.75	2.25	2.5	1.5	2.5	2.75	2.5	2.5	3.5	2.25	1.5
1.75	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	3.5	2.5	0
4.25	3.75	3.75	3.5	3.5	3.5	3.5	3.5	3.75	3.5	3.5	3.75	3.75	3.25
3.25	2.5	2.75	2.75	2.5	3.5	3.25	3.25	3.5	2.5	3.5	3.5	3.25	2.75
3.5	2.5	3.5	2.75	3.75	3.5	3.25	3.25	3.5	3.75	3.25	3.5	3.5	2.75

Gambar 4.3 Susunan data dalam bentuk matriks

2. Setiap data pada matriks dikurangi dengan rata-ratanya sehingga diperoleh matriks seperti pada persamaan (3.4).
3. Matriks kovarian dihitung sehingga dari data contoh seperti pada Gambar 4.3 diperoleh matriks kovarian.

4. Nilai eigen dan nilai vektor dihitung dari matriks kovarian. Perolehan nilai eigen dan vektor eigen seperti pada Gambar 4.4 dianggap dapat mewakili seluruh distribusi data.
5. Jumlah PC yang akan digunakan ditentukan berdasar kriteria tertentu, misalnya akan dipilih PC yang nilai eigen-nya lebih dari 0.1 sehingga terpilih 7 PC. PC terpilih tersebut kemudian digunakan untuk menyusun vektor fitur.

Nilai Eigen	2.393	0.442	0.321	0.267	0.192	0.135	0.111	0.083	0.054	0.037	0.027	0.021	0.015	0.010
Vektor Eigen	0.319	-0.244	0.390	0.571	-0.202	0.232	0.135	0.407	-0.120	0.180	-0.134	0.081	-0.083	0.057
	0.337	-0.176	0.205	-0.223	-0.101	-0.018	-0.328	-0.308	-0.431	0.127	0.147	0.373	0.432	0.014
	0.344	-0.059	0.166	-0.332	-0.018	0.221	0.031	-0.144	-0.221	-0.302	-0.472	-0.446	-0.194	-0.265
	0.250	-0.027	-0.017	-0.156	0.320	0.065	-0.528	0.268	0.048	0.118	0.312	0.079	-0.568	-0.107
	0.307	-0.297	0.402	-0.017	0.381	-0.350	0.233	-0.215	0.464	-0.115	0.163	-0.099	0.062	0.140
	0.196	-0.176	-0.249	0.216	-0.485	-0.144	0.076	-0.150	-0.027	-0.461	0.478	-0.032	-0.181	-0.249
	0.321	-0.225	-0.405	-0.348	-0.307	-0.161	0.126	0.198	0.054	0.259	-0.089	-0.134	-0.065	0.534
	0.235	0.041	-0.131	-0.170	-0.125	0.251	-0.097	0.337	0.550	-0.026	-0.023	0.070	0.473	-0.404
	0.181	-0.189	-0.425	0.317	0.136	-0.036	-0.068	-0.430	0.188	0.186	-0.452	0.323	-0.167	-0.186
	0.229	0.274	-0.066	-0.213	0.159	0.310	0.672	-0.079	-0.115	0.170	0.246	0.318	-0.160	-0.125
	0.150	-0.051	-0.316	0.301	0.331	0.421	-0.070	-0.167	-0.114	0.097	0.285	-0.505	0.283	0.161
	0.028	0.023	-0.042	-0.001	0.093	0.392	-0.093	0.011	0.113	-0.628	-0.108	0.373	-0.023	0.517
	0.210	0.081	-0.278	0.114	0.406	-0.442	0.110	0.429	-0.372	-0.289	-0.126	0.081	0.216	-0.110
	0.402	0.783	0.097	0.207	-0.169	-0.193	-0.166	-0.151	0.120	0.035	-0.049	-0.094	-0.012	0.175

Gambar 4.4 Hasil perhitungan nilai eigen dan vektor eigen

6. Vektor fitur dikalikan dengan matriks data awal sesuai persamaan (3.10) sehingga didapatkan dataset baru dengan dimensi yang lebih kecil seperti terlihat pada Tabel 4.9 yang berisi contoh data dari dataset baru.

Tabel 4.9 Dataset baru hasil proses PCA

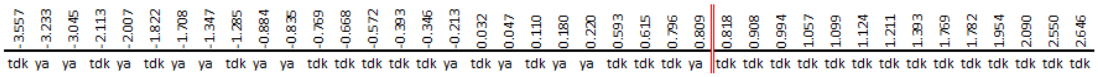
PC1	PC2	PC3	PC4	PC5	PC6	PC7	Kelas
-1.285	0.220	0.258	0.065	-0.389	-0.529	0.299	tdk
-0.768	0.136	0.023	-0.467	0.166	0.051	0.274	tdk
0.109	0.220	-0.652	0.041	0.127	0.035	-0.269	tdk
0.908	-0.226	-0.071	0.296	0.153	0.358	-0.316	tdk
1.953	-0.468	-0.110	0.530	0.989	1.090	-0.506	tdk
…	…	…	…	…	…	…	…
-3.556	1.155	-0.604	-0.570	-1.477	-1.971	0.281	tdk
-3.044	0.381	0.468	-0.807	-1.507	-1.552	0.778	ya
2.090	-0.514	-0.147	0.643	1.070	1.154	-0.394	tdk
-0.213	0.279	-0.468	0.175	0.400	0.178	0.271	ya
0.808	-0.186	-0.104	-0.338	0.145	0.375	-0.319	ya

Tabel 4.9 menunjukan bahwa dimensi data dari *dataset* baru menjadi lebih kecil, dari 14 fitur menjadi 7 fitur saja. Terlihat juga bahwa variabel data yang awalnya berupa nilai diskrit berubah menjadi nilai kontinu. Pada sub sistem PCA dan C4.5, *dataset* baru tersebut akan langsung diklasifikasi dengan algoritma C4.5. *Dataset* baru akan didiskritisasi terlebih dahulu sebelum diklasifikasi dengan algoritma C4.5 pada sub sistem PCA, diskritisasi, dan C4.5. Pada penelitian ini, semua PC akan dicoba dalam proses klasifikasi dan akan dipilih PC yang dapat menghasilkan performa klasifikasi terbaik.

Proses dengan Algoritma Diskritisasi

Proses diskritisasi pada penelitian ini menggunakan diskritisasi berbasis entropi dan kriteria pemberhentian partisinya menggunakan 2 kriteria yaitu jumlah interval dan MDLP. Diskritisasi diterapkan pada dataset baru hasil proses PCA yang contohnya seperti terlihat pada Tabel 4.9. Jika menggunakan kriteria pemberhentian berupa jumlah interval, maka jumlah interval untuk 40 data sampel yang digunakan adalah 3 interval sesuai persamaan (3.13).Diskritisasi berbasis entropi termasuk diskritisasi terawasi sehingga dalam proses diskritisasinya melibatkan label dari masing-masing data.

Untuk mendapatkan titik potong, nilai suatu PC harus diurutkan dulu dari kecil ke besar. Setelah diurutkan, nilai *information entropy* untuk masing-masing kandidat titik potong dihitung dan titik potong yang digunakan adalah titik potong dengan nilai *information entropy* paling kecil. Sebagai contoh, titik potong yang akan dihitung adalah titik potong antara nilai 0.809 dan 0.818 yang merupakan dua buah nilai dari fitur PC1 yang telah diurutkan dari kecil ke besar seperti terlihat pada Gambar 4.5.



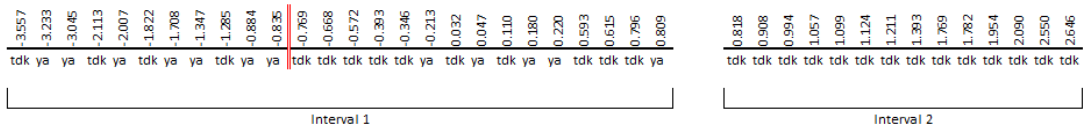
Gambar 4.5 Titik potong pertama untuk atribut PC1

Nilai titik potong antara nilai 0.809 dan 0.818 adalah nilai rata-rata dari kedua nilai tersebut yaitu 0.813. Penghitungan nilai *information entropy* untuk titik potong tersebut dengan menggunakan persamaan (3.11) sebagai berikut:

$$E(A,T;S) = \frac{26}{40}\left(\left(-\frac{14}{26} * log\frac{14}{26}\right) + \left(-\frac{12}{26} * log\frac{12}{26}\right)\right) + \frac{14}{40}\left(\left(-\frac{14}{14} * log\frac{14}{14}\right) + \left(-\frac{0}{14} * log\frac{0}{14}\right)\right)$$

$E(A,T;S)$ =0.647

Titik potong 0.813 adalah titik potong dengan nilai *information entropy* terendah diantara titik potong yang lain sehingga titik potong tersebut terpilih sebagai titik potong untuk diskritisasi pada level pertama. Gambaran hasil proses penentuan interval pada tahap pertama dapat dilihat pada Gambar 4.6.



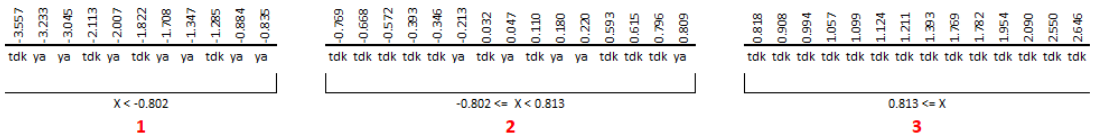
Gambar 4.6 Interval pada atribut PC1 setelah diskritisasi pertama

Jumlah interval yang diharapkan adalah 3, sedangkan jumlah interval yang sudah terbentuk baru 2. Oleh karena itu, perlu dilakukan penentuan titik potong lagi dengan cara memecah interval 1 seperti terlihat pada Gambar 4.6. Nilai titik potong antara nilai -0.835 dan -0.769 adalah -0.802. Penghitungan nilai *information entropy* untuk titik potong tersebut sebagai berikut:

$$E(A,T;S) = \frac{11}{26}\left(\left(-\frac{4}{11} * log\frac{4}{11}\right) + \left(-\frac{7}{11} * log\frac{7}{11}\right)\right) + \frac{15}{26}\left(\left(-\frac{10}{15} * log\frac{10}{15}\right) + \left(-\frac{5}{15} * log\frac{5}{15}\right)\right)$$

$E(A,T;S)$ =0.899

Titik potong 0.802 adalah titik potong dengan nilai *information entropy* terendah diantara titik potong yang lain pada interval 1 sehingga titik potong tersebut terpilih sebagai titik potong untuk diskritisasi pada level kedua. Setelah proses diskritisasi pada level kedua, jumlah interval menjadi 3 interval sehingga kriteria pemberhentian terpenuhi. Hasil akhir diskritisasi untuk PC1 dapat dilihat pada Gambar 4.7. Proses diskritisasi di atas dilakukan juga terhadap PC yang lain sehingga akhirnya didapatkan hasil seperti pada Tabel 4.10.



Gambar 4.7 Hasil akhir proses diskritisasi pada atribut PC1

Tabel 4.10 Sample data hasil PCA yang telah didiskritisasi

PC1	PC2	PC3	PC4	PC5	PC6	PC7	Kelas
1	1	2	2	2	2	3	Tdk
2	1	2	2	2	2	3	Tdk
2	1	2	2	2	2	1	Tdk
3	1	2	3	2	2	1	Tdk
3	1	2	3	3	3	1	Tdk
...	...	...	...	...	...	...	...
1	3	2	1	2	2	3	Tdk
1	2	2	1	1	2	3	Ya
3	1	2	3	3	3	1	Tdk
2	2	2	2	2	2	3	Ya
2	1	2	2	2	2	1	Ya

Kriteria pemberhentian kedua yaitu dengan kriteria MDLP. Jika menggunakan kriteria MDLP, proses *splitting*(pemisahan) akan berhenti ketika nilai Gain(A, T; S) < δ. Contoh penerapan kriteria pada diskritisasi PC1 dapat dilihat pada perhitungan berikut ini:

$$Ent(S_1) = \left(\left(-\frac{14}{26} * log\frac{14}{26}\right) + \left(-\frac{12}{26} * log\frac{12}{26}\right)\right) = 0.996$$

$$Ent(S_2) = \left(\left(-\frac{14}{14} * log\frac{14}{14}\right) + \left(-\frac{0}{14} * log\frac{0}{14}\right)\right) = 0$$

Perhitungan yang pertama dilakukan adalah perhitungan nilai entropi interval 1 (Ent S_1) dan entropi interval 2 (Ent S_2) dengan menggunakan persamaan (3.12). Hasil perhitungan kedua entropi tersebut digunakan untuk perhitungan Δ(A, T; S) dengan menggunakan persamaan (3.16).

$$\Delta(A,T;S) = \log_2(3^2 - 2) - \left[2 * \left(\left(-\frac{28}{40} * \log\frac{28}{40}\right) + \left(-\frac{12}{40} * \log\frac{12}{40}\right)\right) - 2 * 0.996 - 1 * 0\right] = 3.036$$

Selanjutnya, Δ(A, T; S) digunakan untuk perhitungan δ dengan menggunakan persamaan (3.15).

$$\delta = \frac{\log_2(40 - 1) + 3.036}{40} = 0.208$$

Berikut ini adalah perhitungan nilai gain dengan menggunakan persamaan (3.14)

$$Gain(S,A) = \left(\left(-\frac{28}{40} * \log\frac{28}{40}\right) + \left(-\frac{12}{40} * \log\frac{12}{40}\right)\right) - \frac{26}{40} * 0.647 = 0.234$$

Dari perhitungan di atas, nilai δ yang dihasilkan adalah 0.208 dan nilai Gain(S, A) yang dihasilkan adalah 0.234. Hasil perhitungan memperlihatkan nilai Gain(S, A) tidak lebih kecil dari nilai δ sehingga proses partisi bisa diterima. Proses partisi dilakukan terus untuk semua interval yang telah terbentuk sampai kriteria pemberhentian tercapai.

Proses dengan Algoritma C4.5

Langkah pertama pada algoritma C4.5 adalah pemilihan *root node*. Pemilihan *root node* dilakukan dengan cara mencari atribut yang memiliki nilai *gain ratio* paling tinggi. Contoh perhitungan *gain ratio* dengan menggunakan persamaan (3.17) untuk sampel data pada Tabel 4.10 yang merupakan data diskrit untuk atribut PC6 sebagai berikut:

$$Ent(PC6 = 1) = \left(\left(-\frac{0}{1} * log\frac{0}{1}\right) + \left(-\frac{1}{1} * log\frac{1}{1}\right)\right) = 0$$

$$Ent(PC6 = 2) = \left(\left(-\frac{13}{24} * log\frac{13}{24}\right) + \left(-\frac{11}{24} * log\frac{11}{24}\right)\right) = 0.995$$

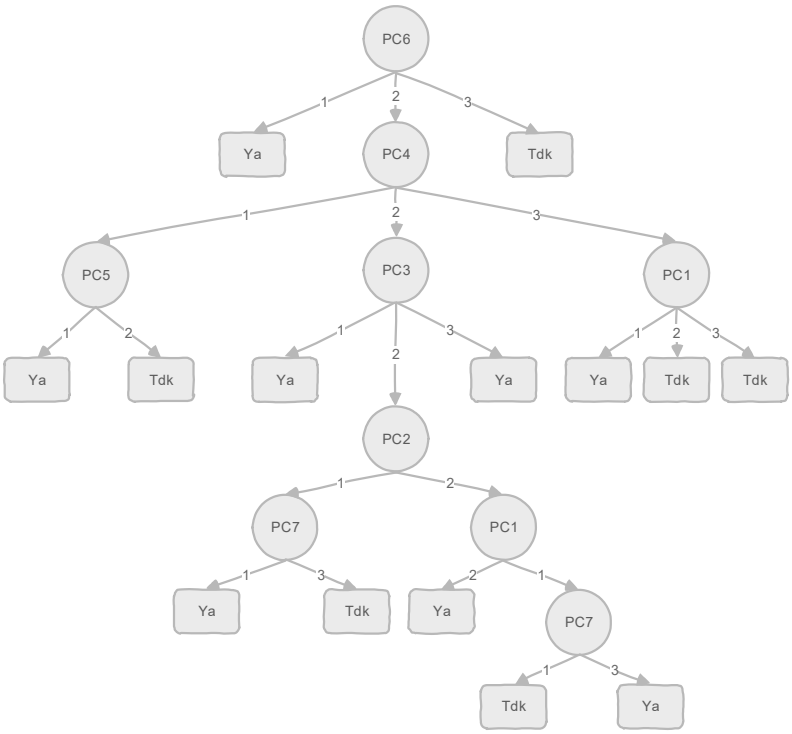
$$Ent(PC6 = 3) = \left(\left(-\frac{15}{15} * log\frac{15}{15}\right) + \left(-\frac{0}{15} * log\frac{0}{15}\right)\right) = 0$$

$$Ent(Total) = \left(\left(-\frac{28}{40} * log\frac{28}{40}\right) + \left(-\frac{12}{40} * log\frac{12}{40}\right)\right) = 0.881$$

$$Gain(S, A) = Ent(Total) - \left(\frac{1}{40} Ent(PC = 1) + \frac{24}{40} Ent(PC = 2) + \frac{15}{40} Ent(PC = 3)\right) = 0.284$$

$$GainRatio(S, A) = \frac{Gain(S, A)}{Ent(PC6 = 1) + Ent(PC6 = 2) + Ent(PC6 = 3)} = 0.286$$

Perhitungan seperti di atas dilakukan untuk semua PC. Hasil perhitungan terhadap semua PC menunjukkan bahwa PC yang memiliki nilai *gain ratio* terbesar adalah PC6 dengan nilai gain ratio sebesar 0.286 sehingga PC menjadi *root node*. Setelah *root node* didapatkan, data pelatihan dibagi sesuai *root node* yang terpilih yaitu PC6.



Gambar 4.8 Pohon keputusan dari data terdiskritisasi

Leaf node PC6 untuk nilai 1 dan 3 sudah homogen terlihat dari nilai entropinya yang bernilai 0 sehingga tidak perlu dilakukan pemisahan lagi, sedangkan *leaf node*

PC6 untuk nilai 2 belum homogen sehingga masih perlu dilakukan pemisahan. Pemilihan atribut terbaik untuk melakukan pemisahan pada data dengan nilai PC6 adalah 2 dilakukan seperti perhitungan di atas dan pemisahan dilakukan terus sampai seluruh pada leaf node memilki nilai target yang sama (homogen). Pohon keputusan yang terbentuk dari data pada Tabel 4.10 dapat dilihat pada Gambar 4.8.

Gambar 4.8 menunjukkan pohon keputusan yang terbentuk dari Tabel 4.10. Gambar tersebut menunjukkan bahwa yang menjadi *root node* dari pohon keputusan di atas adalah atribut PC6. Atribut PC6 dengan nilai 2 masih belum menghasilkan *leaf node* yang homogen sehingga dilakukan pemisahan lagi dengan menggunakan atribut PC4. Pemisahan dengan atribut PC4 ternyata juga tidak langsung menghasilkan *leaf node* yang homogen. Atribut PC4 dengan nilai 1 harus dipisah lagi berdasar atribut PC5, atribut PC4 dengan nilai 2 harus dipisah lagi berdasar atribut PC3, dan atribut PC4 dengan nilai 3 harus dipisah lagi berdasar atribut PC1. Proses pemisahan dilakukan terus sampai kriteria pemberhentian terpenuhi.

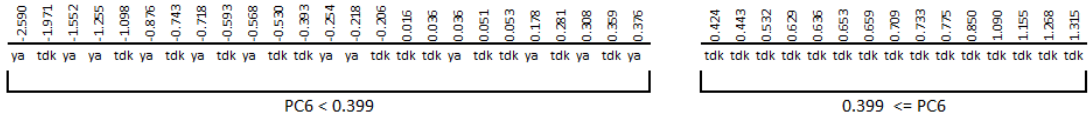
Pada data kontinu seperti yang terlihat pada Tabel 4.9, proses diskritisasi dilakukan sebelum perhitungan *gain ratio*. Proses diskritisasi yang dilakukan sesuai algoritma C4.5 adalah diskritisasi biner. Semua kemungkinan titik potong pada suatu atribut akan diuji coba dan akan dipilih 1 titik potong yang menghasilkan nilai gain tertinggi. Pada atribut PC6, contoh penentuan nilai titik potongnya sebagai berikut. Misal diambil titik potong antara 0.375 dan 0.423 yang nilai titik potongnya adalah nilai rata-rata dari kedua nilai tersebut yaitu 0.399. Selanjutnya, dihitung nilai entropi dan nilai gain berdasar titik potong tersebut.

$$E(A,T;S) = \frac{26}{40}\left(\left(-\frac{14}{26} * log\frac{14}{26}\right) + \left(-\frac{12}{26} * log\frac{12}{26}\right)\right) + \frac{14}{40}\left(\left(-\frac{14}{14} * log\frac{14}{14}\right) + \left(-\frac{0}{14} * log\frac{0}{14}\right)\right)$$

$E(A,T;S)$ =0.647

$$Gain(S,A) = \left(\left(-\frac{28}{40} * \log\frac{28}{40}\right) + \left(-\frac{12}{40} * \log\frac{12}{40}\right)\right) - \frac{26}{40} * 0.647 = 0.234$$

Titik potong 0.399 menghasilkan nilai gain sebesar 0.234 yang merupakan nilai terbesar jika dibandingkan dengan nilai gain dari titik potong lain. Oleh karena itu, titik potong 0.399 digunakan dalam proses diskritisasi sehingga akan menghasilkan interval seperti tampak pada Gambar 4.9.

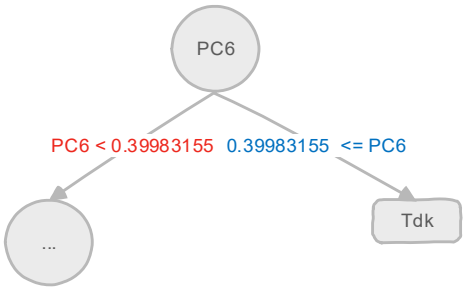


Gambar 4.9 Proses diskritisasi pada atribut PC6

Proses pemilihan titik potong dilakukan pada semua atribut. Selanjutnya, proses penentuan *root node* dilakukan. Semua atribut dihitung *gain ratio*-nya dan atribut dengan *gain ratio* tertinggi akan dipilih sebagai *root node*. Sebagai contoh, perhitungan *gain ratio* dari PC6 sebagai berikut:

$$GainRatio(S,A) = \frac{0.234}{\left(\left(-\frac{14}{26} * \log\frac{14}{26}\right) + \left(-\frac{12}{26} * \log\frac{12}{26}\right)\right) + \left(\left(-\frac{14}{14} * \log\frac{14}{14}\right) + \left(-\frac{0}{14} * \log\frac{0}{14}\right)\right)}$$

Dari perhitungan di atas, didapatkan nilai Gain Ratio(S, A) adalah 0.235 dan nilai tersebut adalah nilai gain ratio tertinggi sehingga PC6 dijadikan sebagai root node. Gambaran awal struktur pohon keputusan dari data pemetaan berbentuk nilai kontinu dapat dilihat pada Gambar 4.10.



Gambar 4.10 Contoh pohon keputusan pada data kontinu

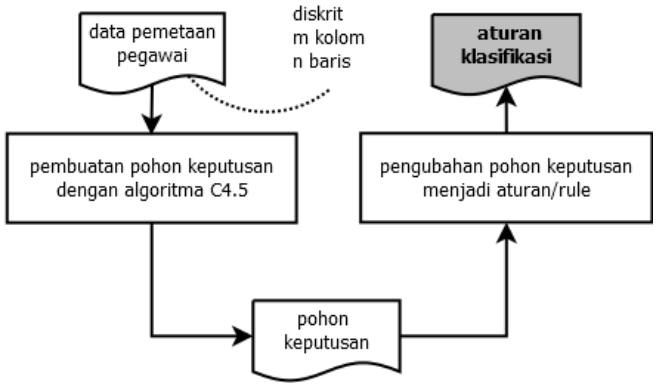
Langkah yang dilakukan setelah pohon keputusan terbentuk adalah mengubah pohon keputusan ke dalam bentuk aturan. Dari pohon aturan pelatihan AMT yang terdapat pada Gambar 4.8 didapatkan aturan-aturan sebagai berikut:

1. Jika PC6 = 1 maka Ya
2. Jika PC6 = 3 maka Tidak
3. Jika PC6 = 2 dan PC4 = 1 dan PC5 = 1 maka Ya
4. Jika PC6 = 2 dan PC4 = 1 dan PC5 = 2 maka Tidak
5. Jika PC6 = 2 dan PC4 = 3 dan PC1 = 1 maka Ya
6. Jika PC6 = 2 dan PC4 = 3 dan PC1 = 2 maka Tidak
7. Jika PC6 = 2 dan PC4 = 3 dan PC1 = 3 maka Tidak
8. Jika PC6 = 2 dan PC4 = 2 dan PC3 = 1 maka Ya
9. Jika PC6 = 2 dan PC4 = 2 dan PC3 = 3 maka Ya
10. Jika PC6 = 2 dan PC4 = 2 dan PC3 = 2 dan PC2 = 1 dan PC7 = 1 maka Ya
11. Jika PC6 = 2 dan PC4 = 2 dan PC3 = 2 dan PC2 = 1 dan PC7 = 3 maka Tdk
12. Jika PC6 = 2 dan PC4 = 2 dan PC3 = 2 dan PC2 = 1 dan PC1 = 2 maka Ya
13. Jika PC6 = 2 dan PC4 = 2 dan PC3 = 2 dan PC2 = 1 dan PC1 = 1 dan PC7 = 1 maka Tidak
14. Jika PC6 = 2 dan PC4 = 2 dan PC3 = 2 dan PC2 = 1 dan PC1 = 1 dan PC7 = 3 maka Ya
15. Kelas default adalah Tidak

Jika nilai kesimpulan dari aturan adalah Ya berarti pegawai yang memenuhi persyaratan sesuai aturan tersebut direkomendasikan untuk mengikuti pelatihan AMT, dan sebaliknya jika nilai kesimpulan adalah Tidak. Kelas *default* dari aturan-aturan di atas adalah Tidak sehingga jika ada data pegawai yang tidak memenuhi persyaratan aturan ke-1 sampai dengan ke-14 maka tidak direkomendasikan mengikuti pelatihan AMT.

4.2.4 Rancangan Sub Sistem

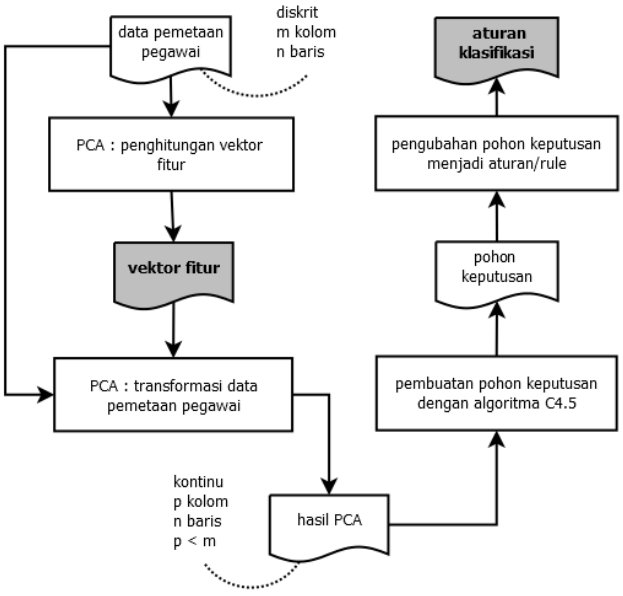
Bagian Back-End terdiri dari 3 buah sub sistem. Sub sistem yang pertama adalah sub sistem C4.5. Penentuan rekomendasi pelatihan pada sub sistem ini hanya dilakukan dengan menggunakan algoritma C4.5 saja. Data pemetaan pegawai diklasifikasi menggunakan algoritma C4.5 sehingga dihasilkan sebuah pohon keputusan. Pohon keputusan tersebut kemudian diubah menjadi bentuk *rule*/aturan. Aturan tersebut merupakan keluaran dari sub sistem ini. Rancangan proses pada sub sistem ini dapat dilihat pada Gambar 4.11.



Gambar 4.11 Rancangan proses pada Sub Sistem C4.5

Sub sistem yang kedua adalah sub sistem PCA dan C4.5. Penentuan rekomendasi pelatihan pada sub sistem ini dilakukan dengan menggunakan algoritma PCA dan C4.5. Algoritma PCA digunakan untuk mengekstrak fitur dari data pemetaan pegawai. Data pemetaan pegawai diproses dengan algoritma PCA dulu sehingga didapatkan sebuah vektor fitur. Vektor fitur kemudian digunakan untuk mengubah

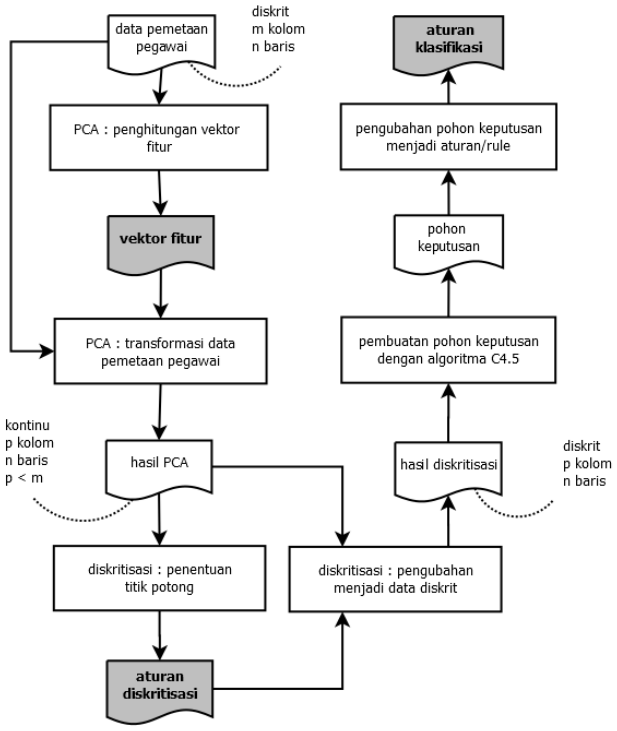
data pemetaan pegawai sehingga didapatkan data dengan fitur baru dan dimensi lebih kecil. Awalnya jumlah kolom pada data adalah p kolom kemudian menjadi m kolom dengan m lebih kecil dari p. Vektor fitur juga digunakan untuk mengubah data pengujian. Data hasil pengubahan yang sudah dalam bentuk kontinu tersebut kemudian diklasifikasi menggunakan algoritma C4.5 sehingga dihasil sebuah pohon keputusan. Pohon keputusan kemudian diubah menjadi bentuk aturan seperti pada sub sistem C4.5. Selain aturan klasifikasi, keluaran dari sub sistem ini adalah vektor fitur yang digunakan untuk melakukan ekstraksi fitur. Rancangan proses pada sub sistem ini dituangkan dalam Gambar 4.12.



Gambar 4.12 Rancangan proses pada Sub Sistem PCA dan C4.5

Sub sistem yang ketiga adalah sub sistem PCA, diskritisasi, dan C4.5. Penentuan rekomendasi pelatihan pada sub sistem ini dilakukan dengan menggunakan metode baru yang diusulkan pada penelitian ini. Metode ini melibatkan algoritma PCA, algoritma diskritisasi berbasis entropi, dan algoritma C4.5. Proses diskritisasinya menggunakan 2 macam kriteria pemberhentian yaitu kriteria jumlah interval dan kriteria MDLP. Proses sub sistem ketiga ini diawali dengan ekstraksi fitur

pada data pemetaan pegawai, seperti pada sub sistem kedua. Setelah data baru didapatkan, proses diskritisasi dilakukan. Proses diskritisasi akan mengubah data hasil pengubahan yang berupa data kontinu menjadi data diskrit. Data yang berupa data diskrit tersebut kemudian diklasifikasi dengan algoritma C4.5. Prosesnya berlanjut sampai didapatkan aturan seperti sub sistem lainnya. Selain memberikan keluaran berupa aturan klasifikasi dan vektor fitur, sub sistem ini juga memberikan keluaran berupa aturan diskritisasi yang digunakan untuk melakukan proses diskritisasi. Gambar 4.13 menunjukkan penggambaran proses pada sub sistem PCA, diskritisasi, dan C4.5.



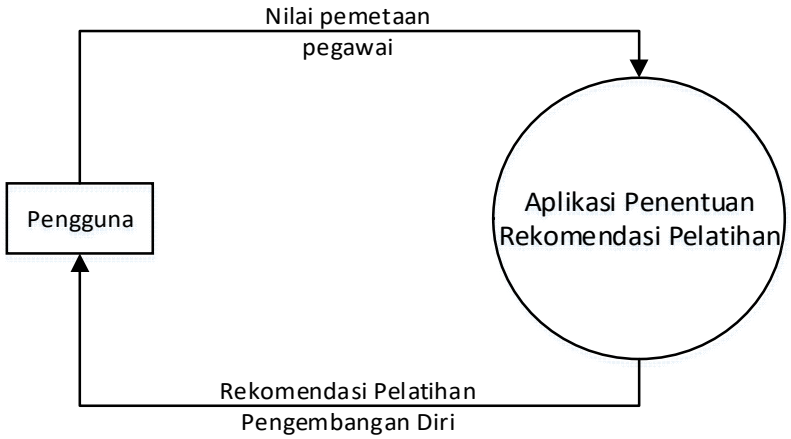
Gambar 4.13 Rancangan proses pada Sub Sistem PCA, diskritisasi, dan C4.5

4.3 Perancangan Bagian Front-End

Bagian *Front-end* menyediakan sarana interaksi dengan pengguna untuk mendapatkan rekomendasi pelatihan sehingga pada bagian ini dilakukan perancangan antarmuka dan juga diagram alir data.

4.3.1 Diagram Alir Data

Sistem memiliki entitas luar yaitu pengguna. Pengguna memasukkan data ke dalam sistem berupa nilai pemetaan pegawai. Sistem akan memproses data masukan tesebut sehingga menghasilkan rekomendasi pelatihan pengembangan diri bagi pegawai. Rangkaian proses tersebut dapat dilihat pada diagram konteks yang terdapat pada Gambar 4.14.

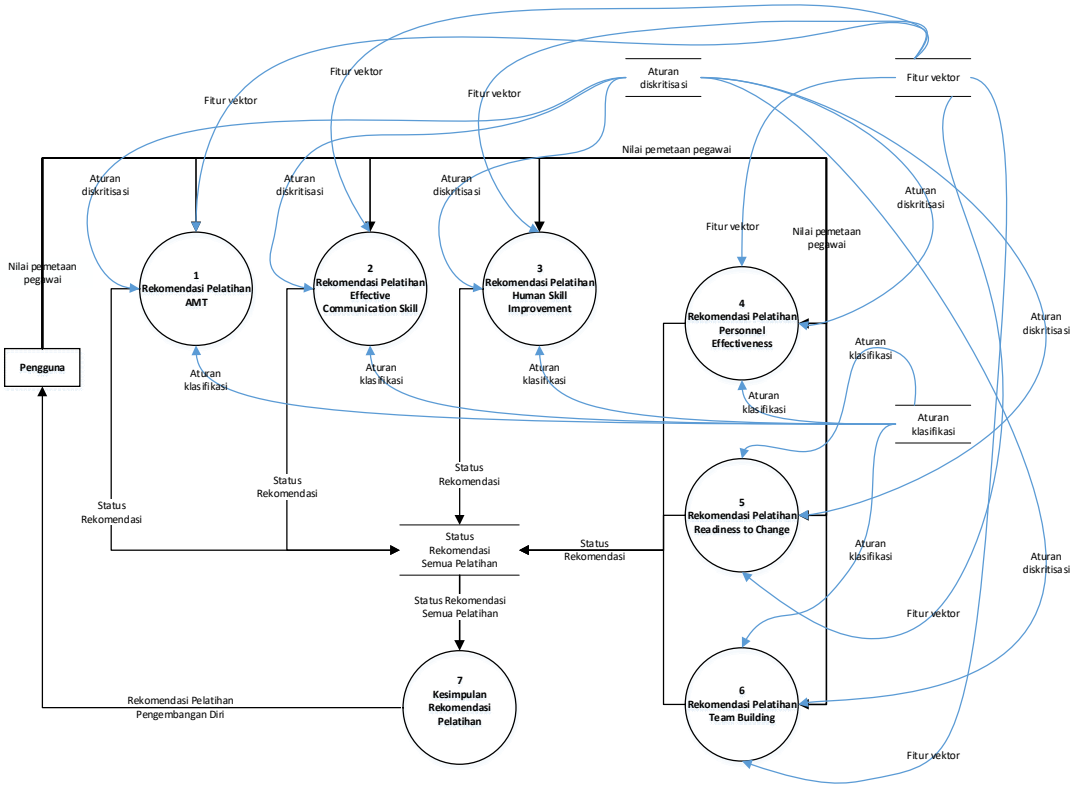


Gambar 4.14 Diagram konteks

Pada tahap selanjutnya, diagram konteks atau disebut juga diagram alir data (DAD) level 0 diperjelas lagi menjadi DAD level 1. Pada DAD level 1, terdapat 7 buah proses yaitu proses Rekomendasi Pelatihan AMT, Rekomendasi Pelatihan *Effective Communication Skill*, Rekomendasi Pelatihan *Human Skill Improvement*, Rekomendasi Pelatihan *Personnel Effectiveness*, Rekomendasi Pelatihan *Readiness to Change*, Rekomendasi Pelatihan *Team Building*, dan Kesimpulan Rekomendasi Pelatihan. Proses pertama sampai dengan keenam terkait dengan jenis pelatihan yang digunakan. Masing-masing proses tersebut akan menentukan apakah pelatihan terkait akan direkomendasikan atau tidak yang ketentuan tersebut diistilahkan sebagai status rekomendasi. Nilai status rekomendasi hanya ada dua macam, yaitu ya(pelatihan terkait direkomendasikan) dan tidak(pelatihan terkait tidak direkomendasikan). Proses Kesimpulan Rekomendasi Pelatihan akan memberikan

kesimpulan tentang pelatihan apa saja yang akan direkomendasikan. Gambaran DAD level 1 dapat dilihat pada Gambar 4.15.

Pada DAD level 2 dari Proses Rekomendasi Pelatihan AMT, terdapat 4 buah proses yaitu proses pemilihan aturan, proses transformasi PCA, proses diskritisasi, dan proses klasifikasi. Proses pemilihan aturan akan mengambil aturan sesuai pelatihan terkait. Gambaran DAD level 2 dapat dilihat pada Gambar 4.16.



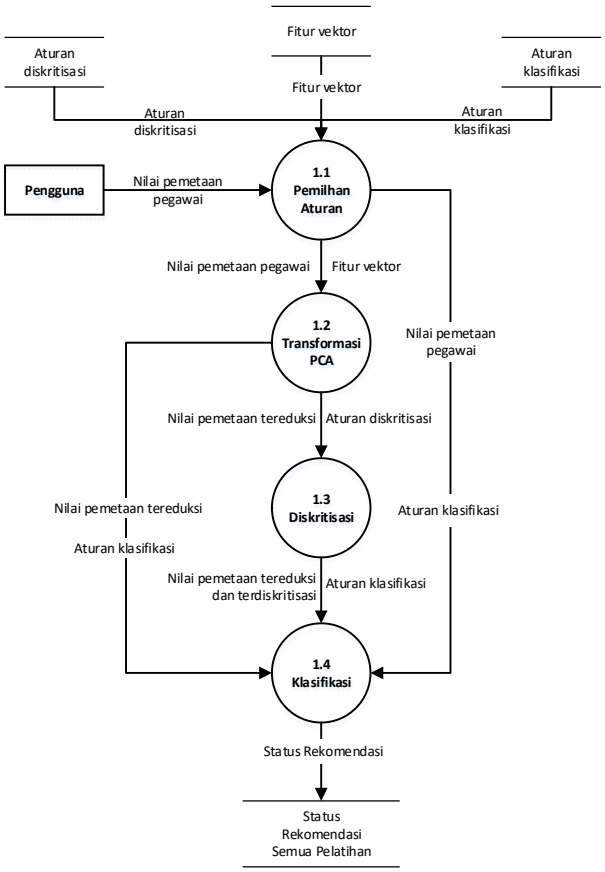
Gambar 4.15 DAD level 1 Bagian Front-End

Proses pemilihan aturan digunakan untuk memilih aturan yang terkait dengan sebuah pelatihan. Misal pelatihan AMT rekomendasi terbaiknya diperoleh dari sub sistem PCA, diskritisasi, dan C4.5. Proses pemilihan peraturan selain akan mengambil aturan klasifikasi, juga akan mengambil fitur vektor dan juga aturan diskritisasi untuk proses pelatihan AMT.

Proses transformasi PCA akan mengubah data pemetaan pegawai menjadi data yang berdimensi lebih kecil dengan menggunakan fitur vektor yang terkait dengan jenis pelatihan yang diinginkan.

Proses diskritisasi akan melakukan diskritisasi data kontinu yang merupakan hasil proses transformasi PCA sesuai aturan diskritisasi.

Proses klasifikasi akan melakukan klasifikasi nilai pemetaan berdasarkan aturan klasifikasi sehingga didapatkan status rekomendasi untuk suatu pelatihan. Status rekomendasi disimpan ke dalam penyimpanan untuk digunakan dalam pengambilan kesimpulan.



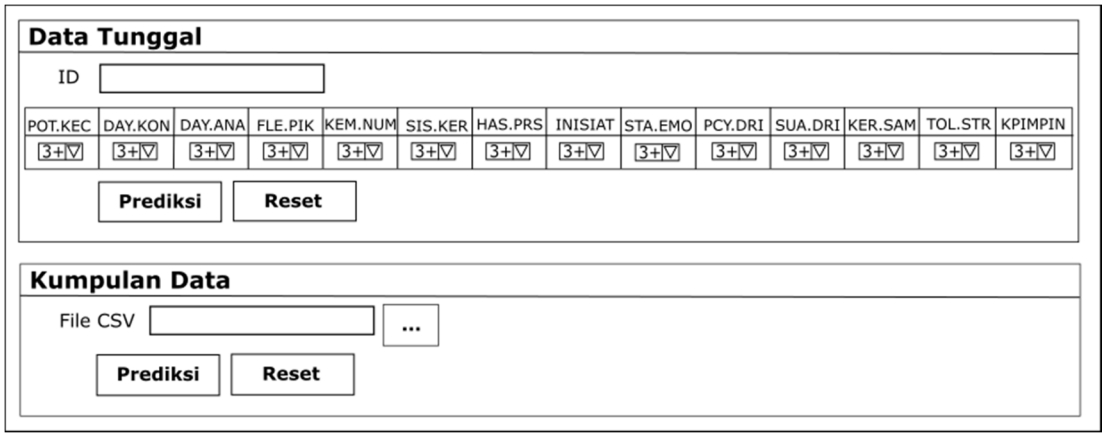
Gambar 4.16 DAD level 2 Proses Rekomendasi Pelatihan AMT

DAD level dua dari proses Rekomendasi Pelatihan Effective Communication Skill, Rekomendasi Pelatihan *Human Skill Improvement*, Rekomendasi Pelatihan

Personnel Effectiveness, Rekomendasi Pelatihan *Readiness to Change*, dan Rekomendasi Pelatihan *Team Building* akan sama dengan DAD level 2 dari Proses Rekomendasi Pelatihan AMT sehingga tidak disediakan gambarnya.

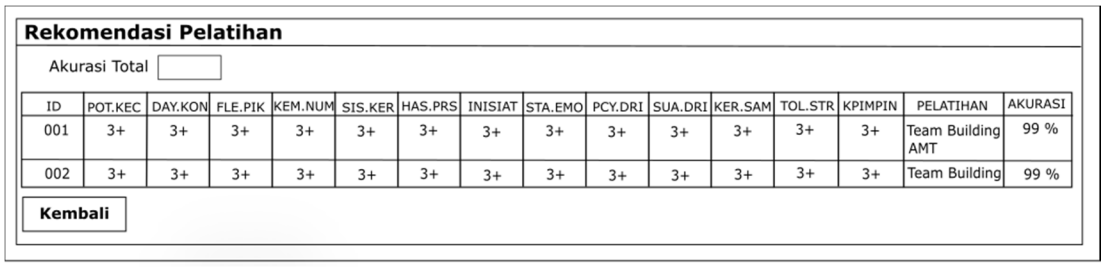
4.3.2 Perancangan Antarmuka

Antarmuka bagian *Front-End* hanya terdiri dari tampilan saja, yaitu tampilan untuk input data dan tampilan rekomendasi pelatihan. Tampilan untuk input data dapat dilihat pada Gambar 4.17, sedangkan tampilan rekomendasi pelatihan dapat dilihat pada Gambar 4.18.



Gambar 4.17 Rancangan halaman untuk memasukkan data

Gambar 4.17 menunjukkan bahwa terdapat dua macam cara memasukkan data, yaitu satu per satu pegawai atau langsung dimasukkan banyak pegawai dengan menyimpannya dalam file CSV (*Comma Separated Value*). Pada cara pemasukan banyak pegawai, lokasi file CSV dimasukkan ke dalam aplikasi. Aplikasi akan memberikan rekomendasinya setelah tombol prediksi di tekan. Jika memasukkan data satu per satu, maka nilai pemetaan dipilih dari combo box sesuai dengan aspek nilai psikologis dari data yang ada.



Gambar 4.18 Rancangan halaman untuk menampilkan hasil rekomendasi

Rekomendasi pelatihan akan ditampilkan disamping nilai pemetaan untuk masing-masing pegawai seperti terlihat pada Gambar 4.18. Selain itu, disamping rekomendasi pelatihan juga ditampilkan akurasi dari rekomendasi yang diberikan oleh sistem dibandingkan rekomendasi yang telah ditentukan oleh *assessor*. Akurasi dari keseluruhan data yang dimasukkan akan ditamplkan pada bagian atas.

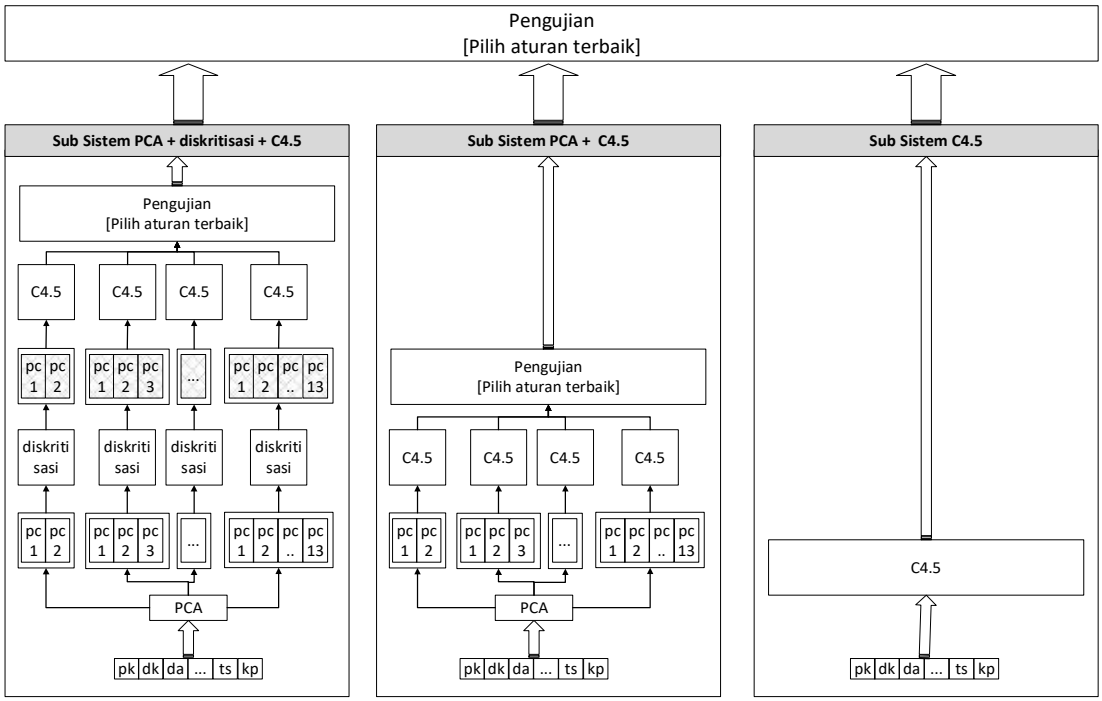
4.4 Rancangan Pengujian

Pengujian dilakukan untuk mengetahui performa sistem secara keseluruhan dan juga untuk mendapatkan aturan yang dapat memberikan performa terbaik yang nantinya akan digunakan untuk memberi rekomendasi. Pengujian akan dilakukan pada bagian *Back-End* dan juga pada bagian *Front-End*.

4.4.1 Pengujian pada Bagian Back-End

Pengujian pada bagian *Back-End* diawali dari internal sub sistem untuk mendapatkan konfigurasi terbaik untuk satu jenis pelatihan. Yang dimaksud konfigurasi terbaik pada sub sistem PCA dan C4.5 adalah jumlah PC yang digunakan yang akan menghasilkan performa terbaik, sedangkan pada pada sub sistem PCA, diskritisasi, dan C4.5, yang dimaksud konfigurasi terbaik adalah jumlah PC yang digunakan dan juga titik potong yang akan menghasilkan performa terbaik. Untuk mendapatkan konfigurasi terbaik tersebut, hampir semua PC diujicobakan dengan menggunakan teknik pengujian 10-*fold cross-validation*. Mulai dari penggunaan 2 PC sampai 13 PC. Pada sub sistem PCA, diskritisasi, dan C4.5, 2 kriteria pemberhentian diujicobakan pada saat proses diskritisasi, yaitu kriteria jumlah interval dan kriteria

MDLP. Proses pada sub sistem C4.5 lebih sederhana karena data pemetaan pegawai langsung diklasifikasi dengan algoritma C4.5. Gambaran proses pengujian internal dapat dilihat pada Gambar 4.19.



Gambar 4.19 Rancangan pengujian

Gambar 4.19 menunjukkan bahwa aturan terbaik dari masing-masing sub sistem dibandingkan performanya terhadap sub sistem yang lain sehingga didapatkan satu aturan yang memiliki performa terbaik untuk satu jenis pelatihan. Aturan terbaik tersebut yang nantinya digunakan untuk merekomendasikan pelatihan terkait. Dengan pengujian ini akan diketahui juga apakah metode baru yang diusulkan pada penelitian ini lebih baik dari dua metode yang lain.

Aturan untuk rekomendasi pelatihan diambil dari sebuah sub sistem yang memberikan performa terbaik dilakukan dengan cara membandingkan performa dari aturan yang dihasilkan dari masing-masing *fold*. Misal, sub sistem yang memberikan aturan terbaik adalah sub sistem PCA, diskritisasi, dan C4.5 dengan menggunakan 9 PC. Pengujian dengan 9 PC dilakukan dengan menggunakan skema 10-*fold-cross-*

validation sehingga dihasilkan 10 buah aturan dimana 1 *fold* menghasilkan 1 aturan. Masing-masing aturan tersebut kemudian diujikan pada keseluruhan data. Aturan yang memberikan performa terbaik pada pengujian tersebut akan digunakan dalam penentuan rekomendasi pelatihan.

Pembandingan performa antar sub sistem maupun di internal sub sistem menggunakan ukuran akurasi. Namun demikian, ukuran performa lain berupa presisi, recall, dan f-Measure tetap digunakan untuk memperjelas hasil pengujian. Perhitungan presisi menggunakan persamaan (4.1), perhitungan *recall* menggunakan persamaan(4.2), perhitungan akurasi menggunakan persamaan (4.3), dan perhitungan *F-Measure* menggunakan persamaan (4.4) yang kesemua perhitungan tersebut berdasarkan *confusion matrix* seperti pada Tabel 4.11.

$$presisi = \frac{TP}{TP + FP} \quad (4.1)$$

$$recall = \frac{TP}{TP + FN} \quad (4.2)$$

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.3)$$

$$F - Measure = 2.\frac{recall.presisi}{recall + pesisi} \quad (4.4)$$

Tabel 4.11 Confusion matrix

		Nilai sebenarnya	
		True	**False**
Nilai Prediksi	**True**	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

4.4.2 Pengujian pada Bagian Front-End

Pada bagian *Front-End* akan dilakukan pembandingan antara rekomendasi pelatihan yang diberikan oleh *assessor* dan rekomendasi pelatihan yang diberikan oleh model. Hasil pembandingan tersebut digunakan untuk melakukan penghitungan akurasi dan *F-measure* berdasar persamaan (4.3) dan (4.4). Sistem dikatakan memiliki performa yang baik apa bila sistem dapat memberikan rekomendasi pelatihan yang sama dengan rekomendasi yang diberikan oleh assessor.

BAB V
IMPLEMENTASI

5.1 Pembangunan Sistem

Sistem ini terbagi menjadi 2 bagian yaitu bagian *Back-end* dan *Front-end* yang keduanya dihubungkan oleh basis data. Basis data yang menghubungkan kedua bagian tersebut dikelola dengan menggunakan DBMS PostgreSQL.

Bagian *Back-end* dibangun dengan menggunakan bahasa pemrograman Python versi 3.5. Bagian *Back-end* tidak memiliki GUI dan dioperasikan hanya menggunakan *console*. Bagian *Front-end* dibangun berbasis web dengan bahasa pemrograman PHP versi 5.6.

5.2 Pembangunan Bagian Back-End

Library eksternal yang banyak digunakan pada pembangunan bagian *Back-end* diantaranya adalah

1. Numpy (www.numpy.org)

   Library ini digunakan untuk melakukan operasi array dan matriks.
2. Scikit-klearn (http://scikit-learn.org/)

   Library ini digunakan pada saat implementasi algoritma PCA.
3. Psycopg2 (http://initd.org/psycopg/)

   Library ini digunakan untuk untuk membangun koneksi antara Python dan DBMS PostgreSQL.

5.2.2 Implementasi Deteksi Outlier

Pendeteksian *outlier* dengan menggunakan algoritma WAVF diawali dengan menghitung nilai probabilitas untuk masing-masing nilai pada tiap atribut. Penghitungan nilai probabilitas untuk masing-masing nilai pada tiap atribut dimulai dengan menghitung nilai frekuensi untuk nilai tersebut yang potongan kodenya dapat dilihat pada Gambar 5.1.

```
arr_freq = []
for i in range(1,15):
    unique, counts = np.unique(arr_kolom[i], return_counts=True)
    j = 0
    dict_freq = {}
    for val in unique:
        dict_freq.update ({val : counts[j]})
        j+=1
    arr_freq.append(dict_freq)

```

Gambar 5.1 Kode program untuk menghitung nilai frekuensi

```
arr_data_prob = []
total_data = len (data)
for baris in data:
    idxKolom = 0
    temp = []
    for i in range(1,len(baris)):
        temp.append(arr_freq[idxKolom][baris[i]]/total_data)
        idxKolom += 1
    arr_data_prob.append(temp)
```

Gambar 5.2 Kode program untuk menghitung nilai probabilitas

```
first=1
arr_kolom = []
for baris in arr_data_prob:
    if first==1:
        idxKolom = 0
        for val in baris:
            arr_kolom.append([val, val])
            idxKolom += 1
        first = 0
    else:
        idxKolom = 0
        for val in baris:
            if val < arr_kolom[idxKolom][0]:
                arr_kolom[idxKolom][0] = val
            elif val > arr_kolom[idxKolom][1]:
                arr_kolom[idxKolom][1] = val
            idxKolom += 1
arr_range = []
for baris in arr_kolom:
    arr_range.append(baris[1]-baris[0])
```

Gambar 5.3 Kode program untuk menghitung nilai range tiap atribut

Gambar 5.1 menunjukkan perulangan pada keseluruhan data pemetaan untuk mendapatkan frekuensi masing-masing nilai pada tiap atribut. Setelah nilai frekuensi

didapatkan, nilai probabilitas dihitung dengan membagi nilai frekuensi dengan jumlah keseluruhan data seperti yang terlihat pada Gambar 5.2.

Proses selanjutnya adalah menghitung nilai *range* untuk masing-masing atribut yang potongan kodenya terdapat pada Gambar 5.3. Baris ke-56 sampai dengan baris ke-72 berfungsi untuk mencari nilai maksimum dan minimum untuk masing-masing atribut. Baris ke 73 sampai dengan baris ke-75 berfungsi untuk menghitung nilai *range* masing-masing atribut dengan mengurangkan nilai maksimum dan minimum.

Nilai probabilitas masing-masing nilai dan nilai *range* masing-masing atribut telah didapatkan sehingga nilai WAVF masing-masing baris data dapat dihitung. Gambar 5.4 baris ke-81 sampai dengan baris ke-88 menunjukkan potongan kode untuk menghitung nilai WAVF untuk masing-masing data, sedangkan baris ke-89 sampai dengan baris ke-90 menunjukkan potongan kode untuk mengurutkan data berdasar nilai WAVF-nya. Setelah diurutkan, pemilihan k data dengan nilai WAFV yang berbeda jauh dari nilai data secara umum dapat dilakukan.

```
81  dict_wafv = {}
82  for idxBaris in range(len(arr_data_prob)):
83      idxKolom = 0
84      wafv = 0
85      for val in arr_data_prob[idxBaris]:
86          wafv += val * arr_range[idxKolom]
87          idxKolom += 1
88      dict_wafv.update({data[idxBaris][0]:wafv})
89  import operator
90  sorted_x = sorted(dict_wafv.items(), key=operator.itemgetter(1))
```

Gambar 5.4 Kode program untuk menghitung nilai WAVF

5.2.3 Implementasi SMOTE-N

Proses over-sampling dilakukan menggunakan program WEKA versi 3.8. Data pelatihan yang sudah bersih dari *outlier* diubah ke dalam format ARFF kemudian diproses dengan menggunakan program WEKA. Kode program untuk mengubah data menjadi format ARFF dapat dilihat pada Gambar 5.5. Baris ke-3 sampai dengan baris ke-4 pada gambar tersebut berfungsi untuk mengambil data suatu pelatihan dari basis

data. Baris ke-6 sampai dengan baris ke-17 berfungsi untuk menuliskan data yang telah diambil ke dalam suatu file sesuai format data ARFF sehingga didapatkan file berformat ARFF seperti pada Gambar 5.6.

```
1  from libpostgre import getARFF
2  data = getARFF(id_pelatihan)
3  from hanfile import write_into_file, append_into_file
4  arr_kolom =
5  ('pk','dk','da','fb','kn','sk','hb','if','se','kd','pd','ks','ts','kp')
6  logfile="pelatihan %s clean.arff" % (id_pelatihan)
7  write_into_file(logfile,"@relation readines_to_change_training \n\n")
8  for i in range(14):
9      tmpStr = '@attribute %s {0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25,
10 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}' % (arr_kolom[i]);
11     append_into_file(logfile,tmpStr+'\n')
12 append_into_file(logfile,'@attribute class {ya, tdk}\n')
   append_into_file(logfile,'@data\n')
13 for baris in (data):
14     append_into_file(logfile,baris+'\n')
```

Gambar 5.5 Kode program untuk mengubah dalam ke dalam format ARFF

```
@relation readines_to_change_training

@attribute pk {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute dk {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute da {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute fb {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute kn {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute sk {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute hb {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute if {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute se {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute kd {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute pd {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute ks {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute ts {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute kp {0.0, 1.25, 1.5, 1.75, 2.25, 2.5, 2.75, 3.25, 3.5, 3.75, 4.25, 4.5, 4.75, 5.25, 5.5, 5.75}
@attribute class {ya, tdk}
@data
2.5,2.75,2.5,2.75,2.75,3.5,2.5,2.5,2.75,2.5,2.5,3.5,2.75,2.5,tdk
2.75,2.5,2.5,2.5,2.75,3.5,3.25,2.75,3.25,2.75,2.5,2.75,3.5,2.5,tdk
3.5,2.75,2.75,2.75,2.5,3.5,2.75,2.75,3.5,3.5,3.25,3.25,3.5,3.25,tdk
3.75,3.5,3.5,3.5,3.5,3.25,2.75,3.25,3.25,3.5,3.5,3.5,3.25,2.5,tdk
```

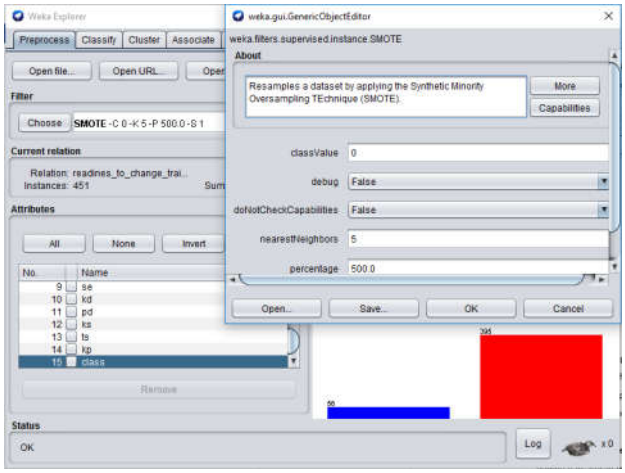
Gambar 5.6 Contoh data dalam format file ARFF

Data yang telah dalam bentuk ARFF diproses dengan menggunakan fitur SMOTE yang terdapat pada program WEKA. Persentase *over-sampling* yang digunakan merujuk kepada Tabel 4.8.. Gambar 5.7 menunjukkan penggunaan program WEKA untuk proses *over-sampling*. Hasil proses *over-sampling* disimpan ke dalam bentuk ARFF. File ARFF hasil proses *over-sampling* tersebut kemudian diubah ke dalam bentuk CSV agar dapat dimasukkan ke dalam basis data.

5.2.4 Implementasi Algoritma PCA

Sub sistem PCA dan C4.5 serta sub sistem PCA, diskritisasi, dan C4.5 menggunakan algoritma PCA untuk melakukan ekstraksi fitur. Implementasi penggunaan algoritma dalam sub sistem tersebut dapat dilihat pada Gambar 5.8. Penjelasan mengenai potongan kode pada gambar tersebut sebagai berikut:

1. Proses mean centering menggunakan library Sklearn dilakukan pada baris ke-28 sampai dengan baris ke-30.
2. Penghitungan matriks kovarian menggunakan library Numpy dilakukan pada baris ke- 33.



Gambar 5.7 Penggunaan filter SMOTE pada WEKA

3. Penghitungan nilai eigen dan vektor eigen dengan menggunakan library Numpy dilakukan pada baris ke-34.
4. Penyusunan dan pengurutan vektor eigen berdasarkan nilai eigen dilakukan pada baris ke-36 sampai dengan baris ke-40.
5. Pemilihan *principal component* dan penyusunan vektor fitur dilakukan pada baris ke-42 sampai dengan baris ke-45. Contoh fitur vektor yang dihasilkan dapat dilihat pada Gambar 5.9.

6. Pengubahan dataset lama menjadi *dataset* baru berdimensi lebih kecil dan penyertaan kembali label data dilakukan dengan memanggil fungsi transformUsingFeatureVector dilakukan pada baris ke-47.

```
28  From sklearn.preprocessing import StandardScaler
29  nilai_fit = StandardScaler().fit(X)
30  X_std = nilai_fit.transform(X)
31
32  import numpy as np
33  cov_mat = np.cov(X_std.T)
34  eig_vals, eig_vecs = np.linalg.eig(cov_mat)
35
36  eig_pairs = [(np.abs(eig_vals[i]), eig_vecs[:,i]) for i in
37  range(len(eig_vals))]
38
39  eig_pairs.sort()
40  eig_pairs.reverse()
41
42  arr_pc=[]
43  for idxJmlPC in range(jml_pc):
44      arr_pc.append(eig_pairs[idxJmlPC][1].reshape(14,1))
45  matrix_w = np.hstack((arr_pc))
46
47  merge = transformUsingFeatureVector(matrix_w, X, y)
```

Gambar 5.8 Kode program untuk mengimplementasikan algoritma PCA

5.2.5 Implementasi Algoritma Diskritisasi

Metode yang digunakan untuk melakukan diskritisasi adalah *entropy based discritization* (EBD). Penentuan titik potong terbaik dilakukan dengan mencari titik potong dengan *information entropy* terkecil (gain terbesar). Sebelum dilakukan penentuan titik potong, nilai yang akan didiskritisasi diurutkan terlebih dahulu. Potongan kode program untuk melakukan penentuan titik potong terbaik dapat dilihat pada Gambar 5.10 yang penjelasannya sebagai berikut:

1. Pengurutan nilai yang akan didiskritisasi dilakukan pada baris ke-75.
2. Penghitungan nilai gain untuk tiap kandidat titik potong dilakukan pada baris ke-76 sampai dengan baris ke-80.
3. Pencarian titik potong terbaik dengan nilai gain terbesar pada baris ke-82.

4. Pemisahan data berdasar titik potong terbaik yang telah didapatkan dilakukan pada baris ke-85.

```
Feature Vector (2 PC --> 14 x 2)
[0.26688512832;0.271154295027;]
[0.308411590849;0.26217959205;]
[0.312840703607;0.249654783873;]
[0.303735342682;0.130308330242;]
[0.252090287938;0.276615237608;]
[0.278439551986;-0.0882786766172;]
[0.302368544988;0.100088106423;]
[0.305635238539;0.0821937083192;]
[0.214815739829;-0.384942101771;]
[0.258842625737;-0.17210597107;]
[0.268197552136;-0.325594671679;]
[0.188990470238;-0.477478898027;]
[0.22220694072;-0.395113585023;]
[0.21781612652;0.097125013532;]
```

Gambar 5.9 Vektor fitur hasil proses PCA

```
75  arr_thres = sorted(alt)
76  for i in range(len(arr_thres)):
77      if i != end:
78          thres = (arr_thres[i] + arr_thres[i+1])/2
79          arr_gain.append((thres,  gainDisc(arr_tbl[j],  col,  result,
80  thres)))
81
82  arr_max = max(arr_gain,key=lambda x: x[1])
83  arr_batas.append(arr_max[0])
84
85  subtresult = get_subtables_disc(arr_tbl[j], col, arr_max[0])
```

Gambar 5.10 Kode program untuk mendapatkan titik potong terbaik

Kriteria pemberhentian yang digunakan dalam proses diskritisasi ada dua macam yaitu jumlah interval dan MDLP. Batas jumlah interval yang digunakan diperoleh dengan menggunakan teknik Dougherty pada persamaan **Error! Reference source not found.**. Implementasi penggunaan teknik tersebut dapat dilihat pada Gambar 5.11.

```
45  def getDoughertyBin(length):
46      log_result = math.log(length, 10)
47      nilai = 1 if log_result < 1 else math.floor(2*log_result)
48      return nilai
```

Gambar 5.11 Kode program untuk mendapatkan jumlah interval terbaik

```
184 arr_max = max(arr_gain,key=lambda x: x[1])
185 N=len(arr_tbl[j][col])
186 GainATS = arr_max[1]
187 kanan = (math.log(N-1, 2)+deltaATS(arr_tbl[j], col, result,
188 thres))/N
    mdlp = True if GainATS > kanan else False
```

Gambar 5.12 Kode program untuk mengimplementasikan kriteria MDLP

Kriteria pemberhentian kedua yaitu kriteria MDLP. Jika menggunakan kriteria MDLP, proses pemisahan akan berhenti ketika nilai Gain(S, A) < δ. Potongan kode program dalam menerapkan kriteria tersebut dapat dilihat pada Gambar 5.12 yang penjelasannya sebagai\ berikut:

1. Penghitungan nilai Gain(S, A) yang mengacu pada persamaan (3.14) dilakukan pada baris ke-184.
2. Penghitungan nilai δ yang mengacu pada persamaan (3.15) dilakukan pada baris ke-187. Kode program pada baris tersebut akan memanggil fungsi lain yang terdapat pada Gambar 5.13.
3. Pembandingan untuk mengetahui terpenuhi tidaknya keriteria MDLP dilakukan pada baris ke-188.

Gambar 5.13 menunjukkan deltaATS yang digunakan untuk melakukan perhitungan pada persamaan (3.16).

```
105 def deltaATS(table, x, res_col, thres):
106     k=len(utils.deldup(table[res_col]))
107     depan = k*info(table, res_col)
108     belakang = infoxDeltaATS(table, x, res_col,thres)
109     return  math.log(3**k-2, 2)-round(depan-belakang)
110
111 def infoxDeltaATS(table, col, res_col,thres):
112     s = 0
113     for subt in utils.get_subtables_disc(table, col, thres):
114         s +=  len(utils.deldup(subt[res_col])) * info(subt, res_col)
115     return s
```

Gambar 5.13 Kode program untuk menghitung detaATS

Hasil akhir dari proses diskritisasi yang dilakukan pada bagian *Back-end* adalah aturan diskritisasi dari tiap atribut yang contohnya dapat dilihat pada Gambar 5.14.

Aturan diskritisasi pada gambar tersebut menunjukkan aturan diskritisasi untuk 2 buah atribut. Aturan diskritisasi tersebut akan digunakan untuk melakukan diskritisasi pada bagian *Front-end*.

```
R1:[1, '<10.170978575776473'];
[2, '>=10.170978575776473 and <11.387723594153371'];
[3, '>=11.387723594153371 and <11.437447965031485'];
[4, '>=11.437447965031485 and <12.170576390929345'];
[5, '>=12.170576390929345 and <12.612728553242874'];
[6, '>=12.612728553242874'];
R2:[1, '<-2.7299137981655814'];
[2, '>=-2.7299137981655814 and <-1.6651950647978109'];
[3, '>=-1.6651950647978109 and <-1.6356220222513227'];
[4, '>=-1.6356220222513227 and <-1.386160901714419'];
[5, '>=-1.386160901714419 and <-1.3655981450532249'];
[6, '>=-1.3655981450532249'];
```

Gambar 5.14 Aturan hasil proses diskritisasi

5.2.6 Implementasi Algoritma C4.5

Library algoritma C4.5 yang digunakan pada penelitian ini dibuat oleh Timofey Trukhanov yang diperoleh dari alamat https://github.com/geerk/C45algorithm. Library tersebut memproses data masukan berupa file JSON. Library tersebut hanya dapat memproses data diskrit saja. Keluaran dari library tersebut adalah rule/aturan yang diperoleh dari pohon keputusan hasil algoritma C4.5.

Library buatan Timofey Trukhanov tersebut dikembangkan lagi untuk dapat memenuhi kebutuhan dari penelitian ini. Pengembangan yang dilakukan sebagai berikut:

1. Penambahan fitur kriteria pemisah (*split criterion*) yaitu gain ratio
2. Kemampuan menangani data kontinu
3. Penambahan *default class* pada aturan yang dihasilkan
4. Kemampuan melakukan klasifikasi berdasar aturan yang dihasilkan

Proses pembuatan pohon keputusan dengan algoritma C4.5 diawali dengan menghitung nilai gain ratio untuk masing-masing atribut. Nilai gain ratio masing-masing atribut kemudian dibandingkan satu sama lain sampai didapatkan nilai

terbesar yang akan menjadi *root node*. Selanjutnya dilakukan pemisahan berdasarkan *root node*. Proses tersebut dilakukan sampai kriteria pemberhentian tercapai dan keseluruhan proses tersebut dilakukan secara rekursif. Kode program dari fungsi mine_c45 yang digunakan untuk pembuatan pohon keputusan dapat dilihat pada Gambar 5.15.

```
1   def mine_c45(table, result, split_criteria):
2       if split_criteria == 'ig':
3           arr_gain = [(k, gain(table, k, result)) for k in
    table.keys() if k != result]
4       else:
5           arr_gain = [(k, gainRatio(table, k, result)) for k in
    table.keys() if k != result]
6       col = max(arr_gain,key=lambda x: x[1])[0]
7       tree = []
8       subtresult,thres = get_subtables(table, col, result)
9       for subt in subtresult:
10          v = subt[col][0]
11          if thres=='': mathsign='='
12          else:
13              mathsign='>' if v > thres else '<='
14              v=thres
15          if is_mono(subt[result]):
16              temp=['%s%s%s' % (col, mathsign, v),
17                    '%s=%s' % (result, subt[result][0])]
18              tree.append(temp)
19          else:
20              del subt[col]
21              if len(subt)==1 and len(subt[result])>0:
22                  temp=['%s%s%s' % (col, mathsign, v),
23                    '%s=%s' % (result, the_most_freq(subt, result))]
24                  tree.append(temp)
25              else:
26                  tree.append(['%s%s%s' % (col,mathsign,v)] +
27  mine_c45(subt, result, split_criteria))
28      return tree
```

Gambar 5.15 Kode program untuk mengimplementasikan algoritma C4.5

Fungsi mine_c45 yang terlihat pada Gambar 5.15 akan memanggil fungsi-fungsi lain diantaranya fungsi yang ada pada Gambar 5.16. Pada Gambar 5.16 baris ke-56 sampai dengan baris ke-57 terdapat fungsi untuk menghitung nilai gain, sedangkan pada baris ke-66 sampai dengan baris ke-72 terdapat fungsi untuk menghitung nilai gain ratio. Kedua fungsi tersebut dipanggil fungsi mine_c45 untuk

menghitung nilai gain/gain ratio dari masing-masing atribut kemudian dicari atribut dengan nilai gain/gain ratio yang paling besar untuk digunakan sebagai kriteria pemisah. Data yang sudah dipisahkan akan diperiksa apakah sudah homogen atau belum dengan menggunakan perintah pada Gambar 5.15 baris ke-15. Jika belum homogen, akan dilakukan pemisahan lagi dengan memanggil fungsi mine_c45 seperti terlihat pada Gambar 5.15 baris ke-27. Setelah proses rekursif tersebut selesai, fungsi mine_c45 akan memberikan keluaran berupa pohon keputusan.

```
56  def gain(table, x, res_col):
57      return info(table, res_col)-infox(table, x, res_col)
58
59  def splitinfo(table, col, res_col):
60      s = 0
61      subtresult,thres = utils.get_subtables(table, col, res_col)
62      for subt in subtresult:
63          s += info(subt, res_col)
64      return s
65
66  def gainRatio(table, x, res_col):
67      EPSILON = 0.001
68      depan = info(table, res_col)
69      belakang = infox(table, x, res_col)
70      pembagi = splitinfo (table, x, res_col)
71      nilai = depan-belakang/pembagi if pembagi>EPSILON else -EPSILON
72      return  nilai
```

Gambar 5.16 Kode program untuk menghitung nilai gain ratio

```
5   def freq(table, col, v):
6       return table[col].count(v)
7
8   def info(table, res_col):
9       s = 0
10      for v in utils.deldup(table[res_col]):
11          p =  freq(table, res_col, v) / float(len(table[res_col]))
12          s += p * math.log(p, 2)
13      return –s
14
15  def infox(table, col, res_col):
16      s =
17      subtresult,thres=utils.get_subtables(table, col, res_col)
18      for subt in subtresult:
19
20  s+=(float(len(subt[col]))/len(table[col]))*info(subt,res_col)
        return s
```

Gambar 5.17 Kode program untuk menghitung nilai entropi

Gambar 5.17 baris ke-8 sampai dengan baris ke-13 menunjukkan fungsi untuk menghitung nilai entropi, sedangkan baris ke-15 sampai dengan baris ke-20 digunakan untuk menghitung nilai *information entropy* setelah dilakukan pemisahan.

```
117 def tree_to_rulesv2(table, result, tree):
118     tmp_rules=__tree_to_rules(tree)
119     def_class = find_default_class(table, result, tmp_rules)
120     tmp_rules.append('~defaultClass=%s' % (def_class))
121     return formalize_rules(tmp_rules),tmp_rules
122 
123 def __tree_to_rules(tree, rule=''):
124     rules = []
125     for node in tree:
126         if isinstance(node, str):
127             rule += node + ','
128         else:
129             rules += __tree_to_rules(node, rule)
130     if rules:
131         return rules
132     return [rule]
```

Gambar 5.18 Kode program untuk menghasilkan aturan

Pohon keputusan yang dihasilkan oleh fungsi mine_c45 diubah ke dalam bentuk aturan dengan menggunakan potongan kode yang terdapat pada Gambar 5.18 yang penjelasannya sebagai berikut:

1. Baris ke-123 sampai dengan baris ke-132 adalah fungsi yang digunakan untuk mengubah pohon keputusan ke dalam bentuk aturan yang dipisahkan dengan koma. Fungsi ini dipanggil pada baris ke-118.
2. Hasil proses pada baris ke-118 kemudian digunakan untuk mencari default class dengan menggunakan fungsi find_default_class pada baris ke-119.

Aturan hasil konversi dari pohon keputusan yang sudah ditambahi informasi *default class* dijadikan keluaran fungsi pada baris ke-121. Aturan hasil konversi disusun dalam bentuk if-then dan juga dalam bentuk CSV. Contoh aturan hasil konversi dapat dilihat pada Gambar 5.19.

```
pc1=1,pc2=4,play=ya,|pc1=1,pc2=2,play=ya,|pc1=1,pc2=5,play=tdk,|pc
1=1,pc2=6,play=ya,|pc1=1,pc2=1,play=ya,|pc1=2,pc2=3,play=tdk,|pc1=
2,pc2=2,play=ya,|pc1=2,pc2=4,play=ya,|pc1=2,pc2=6,play=tdk,|pc1=2,
pc2=1,play=ya,|pc1=5,pc2=6,play=tdk,|pc1=5,pc2=4,play=tdk,|pc1=5,p
c2=2,play=tdk,|pc1=5,pc2=3,play=tdk,|pc1=6,pc2=6,play=tdk,|pc1=6,p
c2=4,play=tdk,|pc1=6,pc2=5,play=tdk,|pc1=6,pc2=2,play=ya,|pc1=6,pc
2=3,play=tdk,|pc1=4,pc2=6,play=tdk,|pc1=4,pc2=2,play=tdk,|pc1=4,pc
2=4,play=tdk,|pc1=4,pc2=3,play=tdk,|pc1=3,play=tdk,|~defaultClass=
tdk
```

Gambar 5.19 Aturan hasil algoritma C4.5 untuk pelatihan AMT

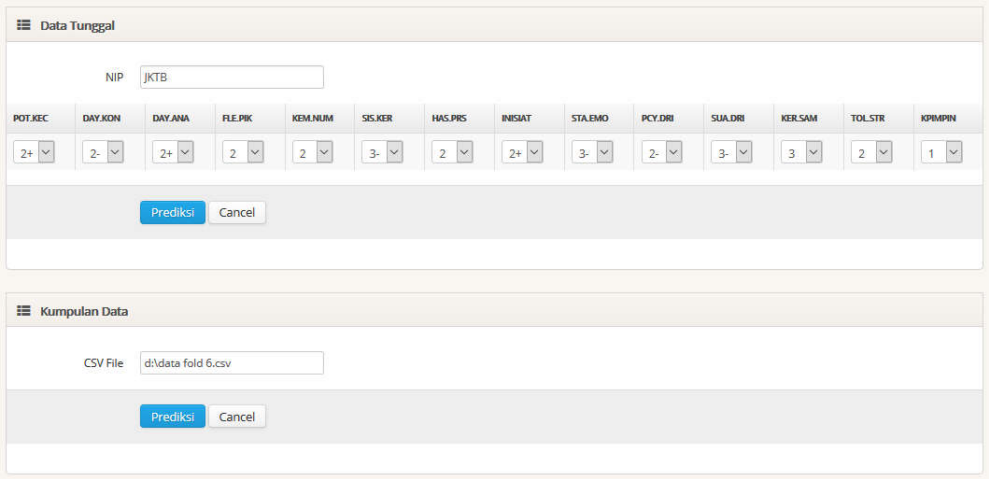
Gambar 5.19 menunjukkan nilai-nilai yang dipisahkan oleh tanda "|". Nilai-nilai tersebut adalah aturan-aturan untuk penentuan rekomendasi. Tiap-tiap kriteria dan keputusan dalam sebuah aturan dipisahkan oleh tanda koma. Misal pada aturan pertama pc1=1,pc2=4,play=ya. Maksud dari aturan tersebut adalah jika nilai pc1 adalah 1 dan nilai pc2 adalah 4 maka pelatihan direkomendasikan mengikuti pelatihan AMT.

5.2.7 Implementasi Pengujian

5.3 Pembangunan Bagian Front-End

Pembangunan bagian *Front-end*, selain menggunakan bahasa pemrograman PHP, juga menggunakan bahasa pemrograman web yang sudah digunakan secara umum yaitu, HTML, CSS, dan Javascript. Pembangunan bagian *Front-end* terdiri dari 2 halaman yaitu halaman input data dan halaman rekomendasi pelatihan.

Pembangunan halaman input data yang tampilannya terlihat pada Gambar 5.20 didasarkan pada rancangan yang terdapat pada Gambar 4.17. Pada halaman tersebut, terdapat dua tombol Prediksi. Tombol Prediksi pertama digunakan untuk memproses data yang dimasukkan satu per satu, sedangkan tombol yang kedua digunakan untuk memproses sekumpulan data yang sudah dimasukkan ke dalam file CSV. Kode program HTML yang digunakan untuk membuat halaman ini dapat dilihat pada Gambar 5.21.



Gambar 5.20 Tampilan halaman untuk input data

Setelah tombol Prediksi pada halaman input data ditekan, data akan dikirimkan ke server untuk diproses. Proses dilakukan dengan membaca aturan-aturan terbaik untuk tiap jenis pelatihan yang telah tersimpan di dalam basis data. Jika aturan tersebut menyatakan untuk mendapatkan rekomendasi suatu pelatihan harus dilakukan proses ekstraksi fitur, maka dilakukan ekstraksi fitur dengan menggunakan kode yang potongannya terlihat pada Gambar 5.22. Ekstraksi fitur dilakukan dengan menggunakan vektor fitur yang sudah tersimpan di dalam basis data.

```
401 <td><select name="optionsRadiosKP" class="cmbTable">
402   <option value="0">-</option><option value="1.25">1-</option>
403   <option value="1.5">1</option><option value="1.75">1+</option>
404   <option value="2.25">2-</option><option value="2.5">2</option>
405   <option value="2.75">2+</option><option value="3.25">3-</option>
406   <option value="3.5">3</option><option value="3.75">3+</option>
407   <option value="4.25">4-</option><option value="4.5">4</option>
408   <option value="4.75">4+</option><option value="5.25">5-</option>
409   <option value="5.5">5</option><option value="5.75">5+</option>
410 </select></td></tr></tbody></table>
411 <div class="form-actions">
412       <button type="submit" class="btn btn-
413 primary">Prediksi</button>
414       <button class="btn">Cancel</button>
415 </div> <!-- /form-actions -->
```

Gambar 5.21 Kode program untuk halaman input data

```
169 for ($c=0;$c<sizeof($array_pca_t);$c++) {
170       $dot[$c]=0;
171       for ($d=0;$d<sizeof($array_pca_t[$c]);$d++) {
172             //echo
173 '['.$c.']['.$d.']*['.$d.']='.$array_pca_t[$c][$d].'*'.$arrdata[$d]
174 .'<br>';
175             $dot[$c]+=$array_pca_t[$c][$d]*$arrdata[$d];
176       }
177 }
```

Gambar 5.22 Kode program untuk mengekstrak fitur dari data

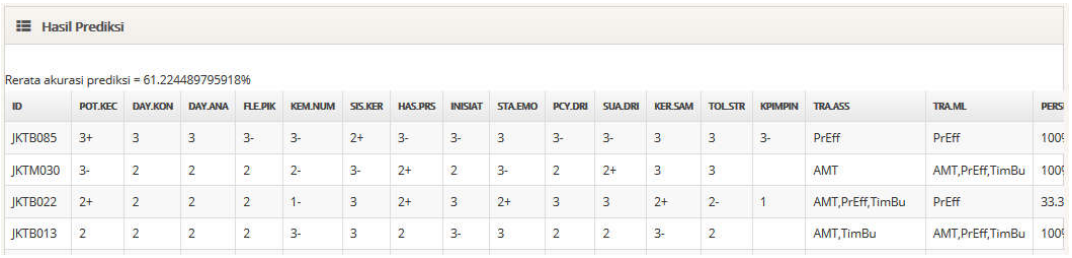
Jika rekomendasi terbaik untuk suatu pelatihan diperoleh dari aturan yang dihasilkan oleh sub sistem PCA, diskritisasi, dan C4.5, maka dilakukan diskritisasi setelah dilakukan ekstraksi fitur. Proses diskritisasi dilakukan dengan menggunakan potongan kode program yang terdapat pada Gambar 5.23. Potongan kode program pada gambar tersebut akan mendiskritisasi data yang dimasukkan berdasar aturan diskritisasi yang telah tersimpan pada $array_disc.

```
184 for ($i=0;$i<sizeof($arrdata);$i++) {
185    if ($array_disc[$i+1][1]=='ALL') { $arrdata_disc[$i]=1; }
186    else {
187       for ($j=1;$j<=sizeof($array_disc[$i+1]);$j++) {
188          if (substr($array_disc[$i+1][$j],0,1)=='<') {
189             $operator = '<';
190          } else { $operator = '>='; }
191          $pjg_operator = strlen($operator);
192          $operand=floatval(substr($array_disc[$i+1][$j],
    $pjg_operator,strlen($array_disc[$i+1][$j])-$pjg_operator));
193          if ($operator == '<' and floatval($arrdata[$i]) < $operand)
194 {
195             $arrdata_disc[$i]=$j;
196          } elseif ($operator == '>=' and floatval($arrdata[$i]) >=
    $operand) {
197             $arrdata_disc[$i]=$j;
198           }
199       }
200    }
201 }
```

Gambar 5.23 Kode program untuk mendiskritisasi nilai kontinu

Setelah data masukan diproses menggunakan aturan-aturan terbaik untuk tiap jenis pelatihan, rekomendasi pelatihan didapatkan. Rekomendasi tersebut kemudian ditampilkan bersanding dengan rekomendasi pelatihan yang diberikan oleh

assessor. Selain itu, ditampilkan juga akurasi dari rekomendasi yang diberikan oleh mesin seperti terlihat pada Gambar 5.24.



Hasil Prediksi

Rerata akurasi prediksi = 61.224489795918%

ID	POT.KEC	DAY.KON	DAY.ANA	FLE.PIK	KEM.NUM	SIS.KER	HAS.PRS	INISIAT	STA.EMO	PCY.DRI	SUA.DRI	KER.SAM	TOL.STR	KPIMPIN	TRA.ASS	TRA.ML	PERS
JKTB085	3+	3	3	3-	3-	2+	3-	3-	3	3-	3-	3	3	3-	PrEff	PrEff	100
JKTM030	3-	2	2	2	2-	3-	2+	2	3-	2	2+	3	3		AMT	AMT,PrEff,TimBu	100
JKTB022	2+	2	2	2	1-	3	2+	3	2+	3	3	2+	2-	1	AMT,PrEff,TimBu	PrEff	33.3
JKTB013	2	2	2	2	3-	3	2	3-	3	2	2	3-	2		AMT,TimBu	AMT,PrEff,TimBu	100

Gambar 5.24 Tampilan halaman untuk menampilkan rekomendasi pelatihan

BAB VI
HASIL DAN PEMBAHASAN

Pada bab ini dilakukan pengujian terhadap sistem yang telah dibangun. Pengujian dilakukan dalam 2 tahapan pengujian yaitu pengujian pada bagian *Back-end* dan pengujian pada bagian *Front-end*. Pengujian yang dilakukan pada bagian *Back-end* akan dijelaskan pada sub bab 6.1, sedangkan pengujian pada bagian *Front-end* akan dijelaskan pada sub bab 6.3.

6.1 Hasil Deteksi Outlier

Proses deteksi *outlier* dilakukan dengan menggunakan algoritma WAVF. Masing-masing data dihitung nilai WAVF-nya. Data dengan nilai WAVF kecil berarti bahwa frekuensi nilai atribut tersebut jarang muncul sehingga dimungkinkan data tersebut adalah *outlier*.

Data awal yang digunakan pada penelitian ini adalah 474 data. Setelah dilakukan perhitungan nilai WAVF, didapatkan nilai WAVF untuk masing-masing data yang contoh hasilnya seperti terlihat pada Gambar 6.1. Gambar tersebut menunjukkan 28 data dengan nilai WAVF terendah. Kolom selisih berisi selisih nilai WAVF baris terkait dengan baris sebelumnya.

Dari data awal yang digunakan dipilih 23 data dengan nilai WAVF terkecil. Pemilihan 23 data tersebut dilakukan dengan memilih batas nilai WAVF-nya terlebih dahulu. Penentuan nilai batas dilakukan dengan melihat selisih nilai. Baris ke-23 pada gambar 6.1 memperlihatkan bahwa selisih nilai baris tersebut dengan baris sesudahnya adalah yang paling besar di antara selisih 2 baris yang lainnya. Oleh karena itu, batas yang digunakan adalah data dengan nilai WAVF kurang dari atau sama dengan 0,00010637. Data pegawai yang memenuhi kriteria tersebut akan dipisahkan dari data yang akan digunakan untuk klasifikasi.

Pada gambar 6.1 terlihat bahwa data yang termasuk outlier itu dapat berupa data yang nilainya sangat rendah, data yang nilainya tinggi, maupun data yang nilainya berada di tengah-tengah. Data yang memiliki nilai sangat rendah dapat dilihat pada baris ke-1. Data yang memiliki nilai yang tinggi misalnya data pada baris ke-3, baris ke-6, dan baris ke-21.

6.2 Hasil Pengujian Performa Model

Pengujian ini dilakukan pada bagian *Back-end*. Pengujian dilakukan secara terpisah untuk masing-masing jenis pelatihan. Hasil akhir pengujian yang dilakukan adalah aturan rekomendasi terbaik untuk tiap jenis pelatihan. Aturan tersebut dapat berasal dari sub sistem C4.5, sub sistem PCA dan C4.5, sub sub sistem PCA, diskritisasi, dan C4.5. Di internal sub sistem pun juga dilakukan pengujian dengan skema pengujian 10-*fold-cross-validation* seperti yang telah dijelaskan pada sub bab 4.4.1 .

Jika pada pengujian didapatkan bahwa rekomendasi terbaik berasal dari sub sistem yang menggunakan algoritma PCA, maka akan dilakukan rotasi faktor. Rotasi faktor akan membuat interpretasi terhadap *principal component* (PC) dapat dilakukan dengan lebih mudah karena rotasi faktor menghasilkan struktur yang lebih sederhana. Dengan rotasi faktor, hubungan antara PC dan variabel asli akan dapat diketahui. Pada penelitian ini, metode rotasi faktor yang digunakan adalah metode Varimax dan prosesnya dilakukan dengan bantuan program *data mining* yaitu Tanagra versi 1.4 (https://eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html).

Proses rotasi faktor akan menghasilkan suatu nilai *loading* (bobot) yang menggambarkan hubungan antara variabel asli dan PC. Semakin besar nilai *loading*-nya maka semakin kuat hubungan antara variabel asli dan PC. Untuk menentukan variabel asli yang memiliki hubungan kuat dengan PC diperlukan batas nilai *loading*. Penentuan batas nilai *loading* dilakukan sesuai preferensi dari peneliti. Namun, jika ingin mendapatkan hubungan yang sangat baik dapat digunakan variabel yang nilai *loading*-nya di atas 0,71 (Tabachnick dan Fidell, 2013).

no	id	pk	dk	da	fb	kn	sk	hb	if	se	kd	pd	ks	ts	kp	class	wavf	selisih
1	452	1-	1-	1-	1	1	1+	1-	1-	1-	1	1	1	1	0	ya	0	0.00000111
2	195	2+	2	3-	2	1	3	2+	3	3	2	3-	3	3	2-	tdk	0.00000111	0.00000775
3	98	3	3-	3-	2+	3	3	3-	3-	3-	3	3-	3-	3-	2	tdk	0.00000886	0.00000000
4	91	2+	1	1	2	2	2	2-	2-	2	2	2+	2	2-	2	ya	0.00000886	0.00000443
5	224	3	3-	3	3-	3-	3	3	2+	3-	3-	3	3-	2+	3	ya	0.00001330	0.00000000
6	236	3+	3-	3+	3+	3	3+	3+	3-	3+	3	3-	3	3	2+	tdk	0.00001330	0.00001330
7	166	2	2-	2-	2-	2	2	2+	2+	2	2+	2+	3-	3-	2-	tdk	0.00002659	0.00001073
8	165	2-	2-	2-	2-	1+	2-	1+	2-	2+	2	2+	3-	3-	1+	ya	0.00003732	0.00001143
9	77	2+	2	2-	2	2-	2	2+	2	2-	2-	2-	2	2	1+	ya	0.00004875	0.00000000
10	240	3-	2+	2	2+	3-	2	2+	2+	3-	3-	3-	3-	2+	2-	tdk	0.00004875	0.00000443
11	309	3	3+	3	3	3	3-	3+	3	3+	3	3-	3	3	3	tdk	0.00005319	0.00000000
12	117	1	1	1	2-	3-	2+	2	2-	2	2	2	2-	2-	0	ya	0.00005319	0.00000443
13	384	2-	1+	2-	2-	1	2	2	2	2	3	2	3	3-	0	tdk	0.00005762	0.00000000
14	270	2-	3-	2	2	2	2-	2+	2	3-	2	2+	3	3	0	tdk	0.00005762	0.00000443
15	415	3	2	2+	2+	2	3	2	2	2+	2	3-	3	3-	0	ya	0.00006205	0.00000443
16	359	2+	2	2	2	2	2	2	2	2	2	2	2	2	0	ya	0.00006648	0.00000000
17	104	2+	2	2	2	2	2+	3-	2-	2	2+	3-	2+	2	2-	tdk	0.00006648	0.00000443
18	422	3-	3-	3	3	3	3	1	1	2+	1	2-	2+	2	0	ya	0.00007091	0.00000443
19	105	2	2	2+	2+	2-	3-	3	3	3-	2+	2+	2	2	2	tdk	0.00007535	0.00000886
20	413	2-	1	1+	2-	2	2	2	2-	2	2	2	2+	2	0	ya	0.00008421	0.00000586
21	323	3	2+	3-	3-	3	3	3	3-	3-	3	3-	3-	3	2-	ya	0.00009007	0.00000000
22	226	3	2	2	1	2+	2+	2+	2+	2	3-	1+	2	2	1	ya	0.00009007	0.00000000
23	214	4	3	3	2+	3	3	2+	2+	2	2+	2-	3-	2-	2	ya	0.00009007	**0.00001630**
24	316	3	3-	3	3-	3-	3	3	3	3-	3	3-	3-	3-	3-	tdk	0.00010637	0.00000443
25	264	3-	2	2+	2-	2-	2	2	2	2	2	2	2+	2+	0	ya	0.00011080	0.00000000
26	140	3	3-	3-	3	3	3	3	3	3-	3-	3	3	3-	3-	tdk	0.00011080	0.00000443
27	312	3	2	2	2	2	2	2	2	3	2+	2+	3	2	0	tdk	0.00011524	0.00000000
28	155	2	2	2-	2-	2-	2	2-	2-	2+	2-	2-	2+	3-	2-	ya	0.00011524	
29	...																	

Gambar 6.1 Hasil perhitungan nilai WAVF

6.2.1 Pelatihan Achievement Motivation Training

Pengujian pada pelatihan AMT (*Achievement Motivation Training*) dilakukan dengan 451 data. Pada data yang digunakan tidak dilakukan *over-sampling* karena kelas datanya cukup seimbang dengan jumlah data untuk kelas Ya adalah 200 data dan jumlah data untuk kelas Tdk adalah 251 data.

Data yang berjumlah 451 data tersebut apabila diproses dengan algoritma EBD dengan kriteria pemberhentian jumlah interval akan menghasilkan 6 buah interval untuk masing-masing variabel. Jika data tersebut diproses dengan algoritma EBD dengan kriteria pemberhentian MDLP, jumlah intervalnya belum tentu sama dengan 6. Bahkan jumlah interval variabel satu dan variabel yang lain dimungkinkan berbeda.

Hasil pemrosesan data tersebut dengan 2 sub sistem, sub sistem PCA dan C4.5 serta sub sistem PCA, diskritisasi, dan C4.5 untuk penentuan rekomendasi pelatihan AMT dapat dilihat pada Lampiran 2. Lampiran 2 menunjukkan bahwa nilai akurasi tertinggi saat menggunakan metode PCA dan C4.5 diperoleh saat menggunakan 10 PC. Nilai akurasi tertinggi saat menggunakan metode PCA, diskritisasi dengan kriteria pemberhentian interval dan C4.5 diperoleh saat menggunakan 9 PC. Nilai akurasi tertinggi saat menggunakan metode PCA, diskritisasi dengan kriteria pemberhentian MDLP dan C4.5 diperoleh saat menggunakan 9 PC. Hasil pemrosesan terbaik tersebut kemudian dibandingkan dengan sub sistem yang lain.

Performa terbaik dari masing-masing sub sistem dibandingkan dan terlihat pada Tabel 6.1 bahwa performa terbaik didapatkan dengan metode ke-4 yaitu pada sub sistem PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian MDLP. Nilai akurasi, presisi, *recall*, dan *F-Measure* tertinggi juga dihasilkan oleh sub sistem tersebut. Sub sistem tersebut memberikan hasil terbaik saat menggunakan 9 PC. Hasil ini menunjukkan bahwa pada saat penentuan pelatihan AMT, metode yang diusulkan memberikan performa yang lebih baik dibanding metode C4.5 dan kombinasi metode PCA dan C4.5.

Tabel 6.1 Perbandingan metode untuk rekomendasi pelatihan AMT

NO	Metode	Akurasi	Presisi	Recall	F-measure	Jml Atribut
1	C4.5	0.639	0.602	0.495	0.543	14
2	PCA dan C4.5	0.518	0.457	0.502	0.479	10
3	PCA, diskritisasi, dan C4.5	0.631	0.594	0.512	0.550	9
4	**PCA, diskritisasi dg MDLP, dan C4.5**	**0.692**	0.676	0.597	0.634	9

Kombinasi metode PCA dan C4.5 memiliki performa terendah dengan selisih akurasi terhadap metode C4.5 sekitar 12%. Kombinasi metode PCA, diskritisasi dengan kriteria MDLP, dan C4.5 memiliki performa tertinggi dengan selisih nilai akurasi terhadap metode C4.5 sekitar 5% dan selisih nilai F-Measure sekitar 8%. Hal tersebut menunjukkan bahwa metode yang diusulkan pada penelitian ini dapat meningkatkan performa dari metode C4.5 untuk kasus penentuan rekomendasi Pelatihan AMT.

Tabel 6.2 Nilai eigen hasil analisis PCA untuk data pelatihan AMT

Principal Component	Nilai awal eigen		
	Total	% dari variansi	*Cumulative %*
1	7.493	53.541	53.541
2	1.672	11.947	65.488
3	0.842	6.016	71.504
4	0.642	4.587	76.091
5	0.531	3.794	79.886
6	0.452	3.230	83.115
7	0.422	3.015	86.131
8	0.406	2.901	89.032
9	**0.363**	**2.594**	**91.626**
10	0.329	2.351	93.976
11	0.283	2.022	95.999
12	0.227	1.622	97.621
13	0.205	1.465	99.085
14	0.128	0.915	100.000

Pohon keputusan yang diperoleh dari konfigurasi terbaik sub sistem PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian MDLP dapat dilihat pada Lampiran 3. Pohon keputusan tersebut memiliki root node PC1. Jika diubah menjadi aturan, pohon keputusan tersebut akan menghasilkan 33 aturan.

Konfigurasi untuk merekomendasikan pelatihan AMT diperoleh dengan menggunakan 9 PC. Nilai eigen untuk PC tersebut dapat dilihat pada Tabel 6.2. Tabel 6.2 menunjukkan bahwa ketika yang digunakan adalah 9 PC maka PC tersebut dapat menjelaskan 91,626 % dari total variansi yang ada seperti terlihat pada kolom *cumulative* pada PC ke-9.

Tabel 6.3 Nilai hubungan variabel asli dan PC terkait pelatihan AMT

Vari-	Principal Component								
abel	1	2	3	4	5	6	7	8	9
dk	**0,832**	0,139	0,179	0,186	0,159	0,055	0,257	0,063	0,125
da	**0,787**	0,110	0,167	0,179	0,102	0,116	0,346	0,083	0,206
pk	**0,776**	0,043	0,105	0,170	0,168	0,029	-0,061	0,115	0,440
fb	0,683	0,125	0,194	0,029	0,299	0,206	0,414	0,123	0,042
se	0,169	**0,895**	0,069	0,162	0,135	0,203	0,076	0,255	0,069
kp	0,272	0,064	**0,923**	0,135	0,113	0,092	0,155	0,016	0,057
kd	0,265	0,148	0,136	**0,848**	0,133	0,172	0,213	0,131	0,052
sk	0,343	0,121	0,117	0,140	**0,811**	0,174	0,188	0,150	0,140
pd	0,127	0,204	0,139	0,477	0,501	0,150	0,240	0,400	0,088
ts	0,141	0,225	0,110	0,195	0,184	**0,864**	0,113	0,265	0,085
if	0,449	0,065	0,248	0,264	0,191	0,118	**0,715**	0,158	0,161
hb	0,436	0,088	0,114	0,310	0,233	0,090	0,585	0,081	0,261
ks	0,127	0,230	0,005	0,137	0,152	0,219	0,091	**0,899**	0,040
kn	0,441	0,079	0,059	0,048	0,128	0,092	0,244	0,040	**0,811**

Variabel asli memiliki hubungan dengan PC (*principal component*) yang nilai hubungannya dapat diketahui pada hasil rotasi faktor yang terlihat pada Tabel 6.3. Variabel asli yang memiliki hubungan yang kuat dengan PC adalah yang nilainya di

atas 0,71 (lihat area yang dihitamkan). Daftar variabel asli yang memiliki pengaruh kuat terhadap masing-masing PC dapat dilihat pada Tabel 6.4.

Tabel 6.4 Hubungan variabel asli dan PC terkait pelatihan AMT

PC	Variabel Asli	Nilai Loading
PC1	dk (Daya Konseptual) da (Daya Analisis) pk (Potensi Kecerdasan)	0,832 0,787 0,776
PC2	se (Stabilitas Emosi)	0,895
PC3	kp (Kepemimpinan)	0,923
PC4	kd (Kepercayaan Diri)	0,848
PC5	sk (Sistematika Kerja)	0,811
PC6	ts (Toleransi terhadap Stress)	0,864
PC7	if (Inisiatif)	0,715
PC8	ks (Kerjasama)	0,899
PC9	kn (Kemampuan Numerikal)	0,811

Variabel asli memiliki hubungan dengan PC (*principal component*) yang nilai hubungannya dapat diketahui pada hasil rotasi faktor yang terlihat pada Tabel 6.3. Variabel asli yang memiliki hubungan yang kuat dengan PC adalah yang nilainya di atas 0,71 (lihat area yang dihitamkan). Daftar variabel asli yang memiliki pengaruh kuat terhadap masing-masing PC dapat dilihat pada Tabel 6.4.

Tabel 6.4 menunjukkan variabel asli yang mempunyai hubungan kuat dengan PC. Penentuan variabel asli yang mempunyai hubungan kuat dengan PC dilakukan dengan melihat nilai hubungan (nilai *loading*) variabel terhadap PC seperti terlihat pada Tabel 6.3. Kolom 1 Principal Component (PC) pada Tabel 6.3 menunjukkan nilai *loading* PC1 dengan variabel dk, da, pk, dan seterusnya. Kolom 1 PC pada baris 1 menunjukkan bahwa nilai *loading* variabel dk terhadap PC1 adalah 0,832. Nilai tersebut berada di atas 0,71 sehingga dapat dikatakan bahwa variabel dk mempunyai hubungan yang sangat kuat dengan PC1. Variabel-variabel yang mempunyai hubungan kuat dengan PC1 dapat diketahui dengan melihat pada kolom 1 PC yang memiliki nilai *loading* di atas 0,71.

Dengan cara di atas dapat diketahui bahwa PC1 sangat dipengaruhi oleh variabel daya konseptual, daya analisis dan potensi kecerdasan. PC2 sangat dipengaruhi oleh variabel stabilitas emosi, sedangkan PC3 sangat dipengaruhi oleh variabel kepemimpinan. PC4 sangat dipengaruhi oleh variabel kepercayaan diri dan PC5 sangat dipengaruhi oleh variabel sistematika kerja. PC6 sangat dipengaruhi oleh variabel toleransi terhadap stress, sedangkan PC7 sangat dipengaruhi oleh variabel inisiatif. PC8 sangat dipengaruhi oleh variabel kerjasama, dan PC9 sangat dipengaruhi oleh variabel kemampuan numerik. Berdasar tabel tersebut, ada 11 variabel yang berpengaruh kuat terhadap penentuan rekomendasi pelatihan AMT seperti terlihat pada Variabel asli memiliki hubungan dengan PC (*principal component*) yang nilai hubungannya dapat diketahui pada hasil rotasi faktor yang terlihat pada Tabel 6.3. Variabel asli yang memiliki hubungan yang kuat dengan PC adalah yang nilainya di atas 0,71 (lihat area yang dihitamkan). Daftar variabel asli yang memiliki pengaruh kuat terhadap masing-masing PC dapat dilihat pada Tabel 6.4.

Tabel 6.4 dan ada 3 variabel yang tidak memiliki pengaruh kuat dengan proses tersebut yaitu variabel fleksibilitas berpikir, penyesuaian diri, dan hasrat berprestasi.

6.2.2 Pelatihan Effective Communication Skill

Pengujian pada pelatihan *Effective Communication Skill* dilakukan dengan 731 data. Jumlah tersebut adalah jumlah data setelah dilakukan *over-sampling*. Data tersebut terdiri dari 336 data dengan kelas Ya dan 391 data dengan kelas Tdk.

Hasil pemrosesan data tersebut dengan 2 sub sistem, sub sistem PCA dan C4.5 serta sub sistem PCA, diskritisasi, dan C4.5 untuk penentuan rekomendasi pelatihan *Effective Communication Skill* dapat dilihat pada Lampiran 4. Lampiran 4 menunjukkan bahwa nilai total tertinggi saat menggunakan metode PCA dan C4.5 diperoleh saat menggunakan 10 PC. Nilai total tertinggi saat menggunakan metode PCA, diskritisasi dengan kriteria pemberhentian interval dan C4.5 diperoleh saat menggunakan 9 PC. Nilai total tertinggi saat menggunakan metode PCA, diskritisasi

dengan kriteria pemberhentian MDLP dan C4.5 diperoleh saat menggunakan 9 PC. Hasil pemrosesan terbaik tersebut kemudian dibandingkan dengan sub sistem yang lain.

Performa terbaik dari masing-masing sub sistem dibandingkan dan terlihat pada Tabel 6.5 bahwa performa terbaik didapatkan dengan metode ke-3 yaitu pada sub sistem PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian MDLP. Nilai akurasi, presisi, *recall*, dan F-Measure tertinggi juga dihasilkan oleh sub sistem tersebut. Sub sistem tersebut memberikan hasil terbaik saat menggunakan 12 PC. Hasil ini menunjukkan bahwa pada saat penentuan pelatihan Effective Communication Skill, metode yang diusulkan memberikan performa yang lebih baik dibanding metode C4.5 dan kombinasi metode PCA dan C4.5.

Kombinasi metode PCA dan C4.5 memiliki performa terendah dengan selisih akurasi terhadap metode C4.5 sekitar 12%. Kombinasi metode PCA, diskritisasi dengan kriteria pemberhentian berdasar jumlah interval, dan C4.5 memiliki performa tertinggi dengan selisih nilai akurasi terhadap metode C4.5 sekitar 13% dan selisih nilai *F-Measure* sekitar 19%. Hal tersebut menunjukkan bahwa metode yang diusulkan pada penelitian ini dapat meningkatkan performa dari metode C4.5 untuk kasus penentuan rekomendasi Pelatihan *Effective Communication Skill*.

Tabel 6.5 Perbandingan metode untuk rekomendasi pelatihan Effective Communication Skill

NO	Metode	Akurasi (0,3)	Presisi (0,2)	Recall (0,3)	F-Measure (0,2)	Nilai	Jml Atribut
1	C4.5	0,707	0,736	0,565	0,640	0,657	14
2	PCA dan C4.5	0,555	0,535	0,224	0,315	0,404	13
3	PCA, diskritisasi, dan C4.5	**0.842**	**0.796**	**0.882**	**0.837**	**0.844**	12
4	PCA, diskritisasi dg MDLP, dan C4.5	0.795	0.745	0.841	0.790	0.798	13

Pohon keputusan yang diperoleh dari konfigurasi terbaik sub sistem PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian jumlah interval dapat dilihat pada Lampiran 5. Pohon keputusan tersebut memiliki root node PC3. Jika diubah menjadi aturan, pohon keputusan tersebut akan menghasilkan 165 aturan.

Tabel 6.6 Nilai eigen hasil analisis PCA untuk data pelatihan Effective Communication Skill

Principal Component	**Nilai awal eigen**		
	Total	**% dari variansi**	***Cumulative* %**
1	7,330	52,36	52,36
2	1,695	12,11	64,47
3	0,806	5,76	70,22
4	0,692	4,95	75,17
5	0,524	3,74	78,91
6	0,455	3,25	82,17
7	0,441	3,15	85,31
8	0,404	2,89	88,20
9	0,379	2,71	90,91
10	0,344	2,45	93,36
11	0,303	2,16	95,53
12	**0,279**	**1,99**	**97,52**
13	0,188	1,34	98,86
14	0,159	1,14	100,00

Tabel 6.7 Nilai hubungan variabel dan PC terkait pelatihan Effective Communication Skill

Vari-	Principal Component											
abel	1	2	3	4	5	6	7	8	9	10	11	12
dk	**0.821**	0.103	0.189	0.127	0.082	0.024	0.216	0.105	0.128	0.201	0.211	0.200
da	0.691	0.055	0.164	0.117	0.087	0.163	0.131	0.105	0.123	0.199	0.447	0.252
se	0.101	**0.878**	0.064	0.159	0.235	0.225	0.080	0.168	0.139	0.062	0.134	0.056
kp	0.185	0.055	**0.943**	0.083	0.034	0.086	0.097	0.077	0.074	0.068	0.151	0.081
kd	0.185	0.204	0.124	**0.807**	0.138	0.149	0.089	0.226	0.178	0.094	0.327	0.093
ks	0.879	0.212	0.036	0.100	**0.907**	0.198	0.058	0.198	0.113	0.475	0.096	0.057
ts	0.092	0.239	0.108	0.125	0.239	**0.863**	0.096	0.188	0.141	0.088	0.168	0.063
fb	0.394	0.121	0.177	0.103	0.090	0.135	**0.740**	0.130	0.177	0.218	0.279	0.166
pd	0.133	0.179	0.098	0.188	0.236	0.191	0.094	**0.861**	0.138	0.029	0.199	0.049
sk	0.211	0.193	0.119	0.194	0.172	0.185	0.159	0.179	**0.782**	0.159	0.276	0.167
kn	0.284	0.067	0.087	0.083	0.055	0.091	0.155	0.029	0.128	**0.847**	0.240	0.269
if	0.214	0.056	0.212	0.219	0.149	0.131	0.254	0.142	0.115	0.165	**0.766**	0.178
hb	0.353	0.185	0.079	0.186	0.032	0.126	0.052	0.169	0.217	0.183	**0.750**	0.136
pk	0.319	0.061	0.107	0.083	0.067	0.064	0.122	0.051	0.136	0.276	0.223	**0.838**

Konfigurasi untuk merekomendasikan pelatihan *Effective Communication Skill* diperoleh dengan menggunakan 12 PC. Nilai eigen untuk PC tersebut dapat dilihat pada Tabel 6.6. Tabel tersebut menunjukkan bahwa ketika yang digunakan adalah 12 PC maka PC tersebut dapat menjelaskan 97,52% dari total variansi yang ada seperti terlihat pada kolom *cumulative* pada PC ke-12.

Variabel asli memiliki hubungan dengan PC (*principal component*) yang nilai hubungannya dapat diketahui pada hasil rotasi faktor yang terlihat pada Tabel 6.7. Variabel asli yang memiliki hubungan yang kuat dengan PC adalah yang nilainya di atas 0,71 (lihat area yang dihitamkan). Daftar variabel asli yang memiliki pengaruh kuat terhadap masing-masing PC dapat dilihat pada Tabel 6.8.

Tabel 6.8 menunjukkan variabel asli yang memiliki hubungan kuat dengan *principal component*. PC1 sangat dipengaruhi oleh variabel daya konseptual yang memiliki nilai loading di atas 0,71. PC2 sangat dipengaruhi oleh variabel stabilitas emosi, sedangkan PC3 sangat dipengaruhi oleh variabel kepemimpinan. PC4 sangat dipengaruhi oleh variabel kepercayaan diri dan PC5 sangat dipengaruhi oleh variabel kerjasama. PC6 sangat dipengaruhi oleh variabel toleransi terhadap stress, sedangkan

PC7 sangat dipengaruhi oleh variabel fleksibilitas berpikir. PC8 sangat dipengaruhi oleh variabel penyesuaian diri, dan PC9 sangat dipengaruhi oleh variabel sistematika kerja. PC10 sangat dipengaruhi oleh variabel kemampuan numerikal dan PC11 sangat dipengaruhi oleh variabel potensi kecerdasan. PC12 sangat dipengaruhi oleh 2 variabel yaitu variabel inisiatif dan hasrat berprestasi. PC13 sangat dipengaruhi oleh varabel potensi kecerdasan. Berdasar tabel tersebut, ada 13 variabel yang berpengaruh kuat terhadap penentuan rekomendasi pelatihan *Effective Communication Skill* dan ada 1 variabel yang tidak memiliki pengaruh kuat dengan proses tersebut yaitu variabel daya analisis.

Tabel 6.8 Hubungan variabel asli dan PC terkait pelatihan Effective Communication Skill

PC	Variabel Asli	Nilai Loading
PC1	dk (Daya Konseptual)	0,821
PC2	se (Stabilitas Emosi)	0,878
PC3	kp (Kepemimpinan)	0,943
PC4	kd (Kepercayaan Diri)	0,807
PC5	ks (Kerjasama)	0,907
PC6	ts (Toleransi terhadap Stress)	0,863
PC7	fb (Fleksibilitas Berpikir)	0,740
PC8	pd (Penyesuaian Diri)	0,861
PC9	sk (Sistematika Kerja)	0,782
PC10	kn (Kemampuan Numerikal)	0,847
PC11	pk (Potensi Kecerdasan)	0,785
PC12	if (Inisiatif) hb (Hasrat Berprestasi)	0,766 0,750
PC13	pk (Potensi Kecerdasan)	0,838

6.2.3 Pelatihan Human Skill Improvement

Pengujian pada pelatihan Human Skill Improvement dilakukan dengan 746 data. Jumlah tersebut adalah jumlah data setelah dilakukan *over-sampling*. Data tersebut terdiri dari 354 data dengan kelas Ya dan 392 data dengan kelas Tdk.

Hasil pemrosesan data tersebut dengan 2 sub sistem, sub sistem PCA dan C4.5 serta sub sistem PCA, diskritisasi, dan C4.5 untuk penentuan rekomendasi pelatihan *Human Skill Imprvement* dapat dilihat pada Lampiran 6. Lampiran 6 menunjukkan bahwa nilai total tertinggi saat menggunakan metode PCA dan C4.5 diperoleh saat menggunakan 13 PC. Nilai total tertinggi saat menggunakan metode PCA, diskritisasi dengan kriteria pemberhentian interval dan C4.5 diperoleh saat menggunakan 11 PC. Nilai total tertinggi saat menggunakan metode PCA, diskritisasi dengan kriteria pemberhentian MDLP dan C4.5 diperoleh saat menggunakan 13 PC. Hasil pemrosesan terbaik tersebut kemudian dibandingkan dengan sub sistem yang lain.

Performa terbaik dari masing-masing sub sistem dibandingkan dan terlihat pada Tabel 6.9 bahwa performa terbaik didapatkan dengan metode ke-3 yaitu pada sub sistem PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian jumlah interval. Nilai akurasi, presisi, dan *F-Measure* tertinggi juga dihasilkan oleh sub sistem tersebut. Sub sistem tersebut memberikan hasil terbaik saat menggunakan 11 PC. Hasil ini menunjukkan bahwa pada saat penentuan pelatihan *Human Skill Improvement*, metode yang diusulkan memberikan performa yang lebih baik dibanding metode C4.5 dan kombinasi metode PCA dan C4.5.

Kombinasi metode PCA dan C4.5 memiliki performa terendah dengan selisih akurasi terhadap metode C4.5 sekitar 20%. Kombinasi metode PCA, diskritisasi dengan kriteria pemberhentian berdasar jumlah interval, dan C4.5 memiliki performa tertinggi dengan selisih nilai akurasi terhadap metode C4.5 sekitar 8% dan selisih nilai *F-Measure* sekitar 9%. Hal tersebut menunjukkan bahwa metode yang diusulkan pada penelitian ini dapat meningkatkan performa dari metode C4.5 untuk kasus penentuan rekomendasi Pelatihan *Human Skill Improvement*.

Tabel 6.9 Perbandingan metode untuk rekomendasi pelatihan Human Skill Improvement

NO	Metode	Akurasi (0,3)	Presisi (0,2)	Recall (0,3)	F-Measure (0,2)	Nilai	Jml Atribut
1	C4.5	0.773	0.774	0.737	0.755	0.759	14
2	PCA dan C4.5	0.573	0.597	0.294	0.394	0.458	13
3	PCA, diskritisasi, dan C4.5	**0.860**	**0.852**	0.852	**0.852**	**0.854**	11
4	PCA, diskritisasi dg MDLP, dan C4.5	0.851	0.819	**0.877**	0.847	0.851	13

Pohon keputusan yang diperoleh dari konfigurasi terbaik sub sistem PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian jumlah interval memiliki root node PC8. Jika diubah menjadi aturan, pohon keputusan tersebut akan menghasilkan 197 aturan.

Tabel 6.10 Nilai eigen hasil analisis PCA untuk data pelatihan Effective Communication Skill

Principal Component	Nilai awal eigen		
	Total	% dari variansi	*Cumulative %*
1	7,771	55,51	55,51
2	1,567	11,19	66,70
3	0,813	5,81	72,51
4	0,619	4,42	76,93
5	0,483	3,45	80,38
6	0,464	3,32	83,70
7	0,429	3,07	86,77
8	0,381	2,72	89,49
9	0,359	2,56	92,06
10	0,322	2,30	94,36
11	**0,271**	**1,94**	**96,30**
12	0,202	1,44	97,74
13	0,286	1,33	99,07
14	0,129	0,93	100

Tabel 6.11 Nilai hubungan variabel dan PC terkait pelatihan Human Skill Improvement

Vari-abel	Principal Component										
	1	2	3	4	5	6	7	8	9	10	11
dk	**0.836**	0.119	0.144	0.189	0.193	0.105	0.106	0.206	0.025	0.032	0.215
fb	**0.749**	0.119	0.187	0.309	0.034	0.225	0.196	0.130	0.172	0.183	0.079
da	**0.733**	0.111	0.171	0.326	0.187	0.085	0.066	0.284	0.126	0.120	0.227
se	0.174	**0.869**	0.074	0.118	0.134	0.126	0.174	0.091	0.248	0.229	0.075
kp	0.230	0.063	**0.931**	0.153	0.099	0.086	0.091	0.070	0.021	0.096	0.086
hb	0.388	0.140	0.095	**0.766**	0.167	0.175	0.178	0.184	0.032	0.138	0.130
if	0.432	0.062	0.262	0.682	0.228	0.159	0.119	0.121	0.196	0.111	0.125
kd	0.236	0.146	0.127	0.235	**0.843**	1.147	0.176	0.062	0.183	0.172	0.101
sk	0.264	0.160	0.128	0.242	0.174	**0.798**	0.179	0.118	0.193	0.225	0.156
pd	0.218	0.216	0.132	0.209	0.200	0.172	**0.809**	0.094	0.254	0.181	0.099
kn	0.386	0.093	0.083	0.188	0.057	0.098	0.082	**0.848**	0.069	0.079	0.216
ks	0.137	0.256	0.022	0.109	0.172	0.154	0.209	0.068	**0.858**	0.223	0.082
ts	0.151	0.251	0.124	0.154	0.172	0.190	0.161	0.080	0.237	**0.845**	0.059
pk	0.419	0.092	0.131	0.183	0.119	0.159	0.105	0.278	0.103	0.069	**0.785**

Konfigurasi untuk merekomendasikan pelatihan *Human Skill Improvement* diperoleh dengan menggunakan 11 PC. Nilai eigen untuk PC tersebut dapat dilihat pada Tabel 6.10. Tabel tersebut menunjukkan bahwa ketika yang digunakan adalah 11 PC maka PC tersebut dapat menjelaskan 96,30 % dari total variansi yang ada seperti terlihat pada kolom *cumulative* pada PC ke-11.

Variabel asli memiliki hubungan dengan PC (*principal component*) yang nilai hubungannya dapat diketahui pada hasil rotasi faktor yang terlihat pada Tabel 6.11. Variabel asli yang memiliki hubungan yang kuat dengan PC adalah yang nilainya di atas 0,71 (lihat area yang dihitamkan). Daftar variabel asli yang memiliki pengaruh kuat terhadap masing-masing PC dapat dilihat pada Tabel 6.12q A.

Tabel 6.12 menunjukkan variabel asli yang memiliki hubungan kuat dengan *principal component*. PC1 sangat dipengaruhi oleh variabel daya konseptual, daya analisis dan fleksibiltas berpikir yang memiliki nilai loading di atas 0,71. PC2 sangat dipengaruhi oleh variabel stabilitas emosi, sedangkan PC3 sangat dipengaruhi oleh variabel kepemimpinan. PC4 sangat dipengaruhi oleh variabel hasrat berprestasi dan

PC5 sangat dipengaruhi oleh variabel kepercayaan diri. PC6 sangat dipengaruhi oleh variabel sistematika kerja, sedangkan PC7 sangat dipengaruhi oleh variabel penyesuaian diri. PC8 sangat dipengaruhi oleh variabel kemampuan numerikal, dan PC9 sangat dipengaruhi oleh variabel kemampuan kerjasama. PC10 sangat dipengaruhi oleh variabel toleransi terhadap stress dan PC11 sangat dipengaruhi oleh variabel potensi kecerdasan. Berdasar tabel tersebut, ada 13 variabel yang berpengaruh kuat terhadap penentuan rekomendasi pelatihan *Human Skill Improvement* dan ada 1 variabel yang tidak memiliki pengaruh kuat dengan proses tersebut yaitu variabel inisiatif.

Tabel 6.12 Hubungan variabel asli dan PC terkait pelatihan Human Skill Improvement

PC	Variabel Asli	Nilai Loading
PC1	dk (Daya Konseptual) fb (Fleksibilitas Berpikir) da (Daya Analisis)	0,836 0,749 0,733
PC2	se (Stabilitas Emosi)	0,869
PC3	kp (Kepemimpinan)	0,931
PC4	hb (Hasrat Berprestasi)	0,766
PC5	kd (Kepercayaan Diri)	0,843
PC6	sk (Sistematika Kerja)	0,798
PC7	pd (Penyesuaian Diri)	0,809
PC8	kn (Kemampuan Numerikal)	0,848
PC9	ks (Kerjasama)	0,858
PC10	ts (Toleransi terhadap Stress)	0,845
PC11	pk (Potensi Kecerdasan)	0,785

6.2.4 Pelatihan Personnel Effectiveness

Pengujian pada pelatihan *Personnel Effectiveness* dilakukan dengan 585 data. Jumlah tersebut adalah jumlah data setelah dilakukan *over-sampling*. Data tersebut terdiri dari 317 data dengan kelas Ya dan 268 data dengan kelas Tdk.

Hasil pemrosesan data tersebut dengan 2 sub sistem, sub sistem PCA dan C4.5 serta sub sistem PCA, diskritisasi, dan C4.5 untuk penentuan rekomendasi pelatihan

Personnel Effectiveness dapat dilihat pada Lampiran 7. Lampiran 7 menunjukkan bahwa nilai total tertinggi saat menggunakan metode PCA dan C4.5 diperoleh saat menggunakan 2 PC. Nilai total tertinggi saat menggunakan metode PCA, diskritisasi dengan kriteria pemberhentian interval dan C4.5 diperoleh saat menggunakan 10 PC. Nilai total tertinggi saat menggunakan metode PCA, diskritisasi dengan kriteria pemberhentian MDLP dan C4.5 diperoleh saat menggunakan 2 PC. Hasil pemrosesan terbaik tersebut kemudian dibandingkan dengan sub sistem yang lain.

Performa terbaik dari masing-masing sub sistem dibandingkan dan terlihat pada Tabel 6.13 bahwa performa terbaik didapatkan dengan metode ke-3 yaitu pada sub sistem PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian jumlah interval. Nilai akurasi dan nilai *F-Measure* tertinggi didapatkan dengan menggunakan metode ke-3. Nilai presisi tertinggi didapatkan dengan menggunakan metode ke-1, sedangkan nilai *recall* tertinggi didapatkan dengan menggunakan metode ke-4. Nilai total tertinggi didapatkan dengan menggunakan metode ke-3, yaitu PCA, diskritisasi, dan C4.5. Hasil ini menunjukkan bahwa pada saat penentuan pelatihan *Personnel Effectiveness*, metode yang diusulkan memberikan performa yang lebih baik dibanding metode C4.5 dan kombinasi metode PCA dan C4.5.

Tabel 6.13 Perbandingan metode untuk rekomendasi pelatihan Personnel Efectiveness

NO	Metode	Akurasi (0,3)	Presisi (0,2)	Recall (0,3)	F-Measure (0,2)	Nilai	Jml Atribut
1	C4.5	0.636	**0.666**	0.659	0.662	0.654	14
2	PCA dan C4.5	0.550	0.565	0.733	0.638	0.625	2
3	PCA, diskritisasi, dan C4.5	**0.645**	0.650	0.745	**0.695**	**0.686**	10
4	PCA, diskritisasi dg MDLP, dan C4.5	0.593	0.593	**0.806**	0.683	0.675	2

Kombinasi metode PCA dan C4.5 memiliki performa terendah dengan selisih akurasi terhadap metode C4.5 sekitar 8%. Kombinasi metode PCA, diskritisasi dengan

kriteria pemberhentian berdasar jumlah interval, dan C4.5 memiliki performa tertinggi dengan selisih nilai akurasi terhadap metode C4.5 sekitar 0,9% dan selisih nilai *F-Measure* sekitar 3%. Hal tersebut menunjukkan bahwa metode yang diusulkan pada penelitian ini dapat meningkatkan performa dari metode C4.5 untuk kasus penentuan rekomendasi pelatihan *Personnel Effectiveness* meskipun peningkatannya tidak terlalu signifikan.

Tabel 6.14 Nilai eigen hasil analisis PCA untuk data pelatihan Personnel Effectiveness

Principal Component	**Nilai awal eigen**		
	Total	**% dari variansi**	***Cumulative %***
1	7,668	54,77	54,77
2	1,576	11,26	66,03
3	0,815	5,82	71,85
4	0,579	4,14	75,99
5	0,492	3,52	79,51
6	0,472	3,37	82,88
7	0,437	3,12	86,00
8	0,424	3,03	89,03
9	0,362	2,59	91,62
10	**0,323**	**2,31**	**93,92**
11	0,301	2,15	96,07
12	0,216	1,54	97,62
13	0,204	1,46	99,07
14	0,129	0,93	100

Pohon keputusan yang diperoleh dari konfigurasi terbaik sub sistem PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian jumlah interval memiliki root node yaitu PC6. Jika diubah menjadi aturan, pohon keputusan tersebut akan menghasilkan 275 aturan.

Konfigurasi untuk merekomendasikan pelatihan *Personnel Effectiveness* diperoleh dengan menggunakan 10 PC. Nilai eigen untuk PC tersebut dapat dilihat pada Tabel 6.14.Tabel tersebut menunjukkan bahwa ketika yang digunakan adalah 10

PC maka PC tersebut dapat menjelaskan 93,92 % dari total variansi yang ada seperti terlihat pada kolom *cumulative* pada PC ke-10.

Tabel 6.15 Nilai hubungan variabel dan PC terkait pelatihan Personnel Effectiveness

Vari-abel	Principal Component									
	1	2	3	4	5	6	7	8	9	10
fb	**0.802**	0.135	0.201	0.019	0.202	0.191	0.178	0.081	0.065	0.166
dk	**0.792**	0.097	0.164	0.155	0.391	0.099	0.069	0.151	0.121	0.029
da	**0.787**	0.107	0.157	0.189	0.332	0.103	0.108	0.124	0.218	0.012
if	**0.737**	0.164	0.257	0.251	(0.095)	0.185	0.101	0.055	0.255	0.213
hb	0.705	0.005	0.080	0.271	0.007	0.196	0.108	0.146	0.314	0.319
ks	0.162	**0.901**	0.006	0.122	0.055	0.116	0.200	0.234	0.053	0.154
kp	0.336	(0.008)	**0.898**	0.143	0.117	0.114	0.096	0.062	0.075	0.091
kd	0.302	0.142	0.160	**0.857**	0.101	0.132	0.162	0.129	0.074	0.191
pk	0.435	0.059	0.133	0.093	**0.751**	0.146	0.058	0.078	0.289	0.127
sk	0.380	0.164	0.156	0.166	0.159	**0.800**	0.167	0.153	0.144	0.185
ts	0.197	0.227	0.106	0.159	0.056	0.133	**0.869**	0.226	0.097	0.177
se	0.177	0.244	0.064	0.119	0.074	0.113	0.208	**0.893**	0.071	0.150
kn	0.429	0.066	0.087	0.074	0.282	0.121	0.104	0.078	**0.804**	0.054
pd	0.278	0.245	0.128	0.249	0.135	0.190	0.242	0.216	0.061	**0.768**

Tabel 6.16 Hubungan variabel asli dan PC terkait pelatihan Personnel Effectiveness

PC	Variabel Asli	Nilai Loading
PC1	fb (Fleksibilitas Berpikir) dk (Daya Konseptual) da (Daya Analisis) if (Inisiatif)	0,802 0,792 0,787 0.737
PC2	ks (kerjasama)	0,901
PC3	kp (Kepemimpinan)	0,898
PC4	kd (Kepercayaan Diri)	0,857
PC5	pk (Potensi Kecerdasan)	0,751
PC6	sk (Sistematika Kerja)	0,800
PC7	ts (Toleransi terhadap Stress)	0,869
PC8	se (Stabilitas Emosi)	0,893
PC9	kn (Kemampuan Numerikal)	0,804
PC10	pd (Penyesuaian Diri)	0,768

Variabel asli memiliki hubungan dengan PC (*principal component*) yang nilai hubungannya dapat diketahui pada hasil rotasi faktor yang terlihat pada Tabel 6.15. Variabel asli yang memiliki hubungan yang kuat dengan PC adalah yang nilainya di atas 0,71 (lihat area yang dihitamkan). Daftar variabel asli yang memiliki pengaruh kuat terhadap masing-masing PC dapat dilihat pada Tabel 6.16.

Tabel 6.16 menunjukkan variabel asli yang memiliki hubungan kuat dengan principal component. PC1 sangat dipengaruhi oleh variabel fleksbilitas berpikir, daya konseptual, daya analisis dan inisiatif yang memiliki nilai loading di atas 0,71. PC2 sangat dipengaruhi oleh variabel kerjasama, sedangkan PC3 sangat dipengaruhi oleh variabel kepemimpinan. PC4 sangat dipengaruhi oleh variabel kepercayaan diri dan PC5 sangat dipengaruhi oleh variabel potensi kecerdasan. PC6 sangat dipengaruhi oleh variabel sistematika kerja, sedangkan PC7 sangat dipengaruhi oleh variabel toleransi terhadap stress. PC8 sangat dipengaruhi oleh variabel stabilitas emosi, dan PC9 sangat dipengaruhi oleh variabel kemampuan numerikal. PC10 sangat dipengaruhi oleh variabel penyesuaian diri. Ada 13 variabel yang berpengaruh kuat terhadap penentuan rekomendasi pelatihan *Personnel Effectiveness* dan ada 1 variabel yang tidak memiliki pengaruh kuat dengan proses tersebut yaitu variabel hasrat berprestasi.

6.2.5 Pelatihan Readiness to Change

Pengujian pada pelatihan *Readiness to Change* dilakukan dengan 811 data. Jumlah tersebut adalah jumlah data setelah dilakukan *over-sampling*. Data tersebut terdiri dari 400 data dengan kelas Ya dan 411 data dengan kelas Tdk.

Hasil pemrosesan data tersebut dengan 2 sub sistem, sub sistem PCA dan C4.5 serta sub sistem PCA, diskritisasi, dan C4.5 untuk penentuan rekomendasi pelatihan *Readiness to Change* dapat dilihat pada Lampiran 8. Lampiran 8 menunjukkan bahwa nilai total tertinggi saat menggunakan metode PCA dan C4.5 diperoleh saat menggunakan 2 PC. Nilai total tertinggi saat menggunakan metode PCA, diskritisasi

dengan kriteria pemberhentian interval dan C4.5 diperoleh saat menggunakan 13 PC. Nilai total tertinggi saat menggunakan metode PCA, diskritisasi dengan kriteria pemberhentian MDLP dan C4.5 diperoleh saat menggunakan 7 PC. Hasil pemrosesan terbaik tersebut kemudian dibandingkan dengan sub sistem yang lain.

Performa terbaik dari masing-masing sub sistem dibandingkan dan terlihat pada Tabel 6.17 bahwa performa terbaik didapatkan dengan metode ke-4 yaitu pada sub sistem PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian MDLP. Nilai akurasi, *recall*, dan nilai *F-Measure* tertinggi didapatkan dengan menggunakan metode ke-4. Nilai presisi tertinggi didapatkan dengan menggunakan metode ke-3, Nilai total tertinggi didapatkan dengan menggunakan metode ke-4. Hasil ini menunjukkan bahwa pada saat penentuan pelatihan Readiness to Change, metode yang diusulkan memberikan performa yang lebih baik dibanding metode C4.5 dan kombinasi metode PCA dan C4.5.

Kombinasi metode PCA dan C4.5 memiliki performa terendah dengan selisih akurasi terhadap metode C4.5 sekitar 33%. Kombinasi metode PCA, diskritisasi dengan kriteria pemberhentian berdasar MDLP, dan C4.5 memiliki performa tertinggi dengan selisih nilai akurasi terhadap metode C4.5 sekitar 7% dan selisih nilai *F-Measure* sekitar 7%. Hal tersebut menunjukkan bahwa metode yang diusulkan pada penelitian ini dapat meningkatkan performa dari metode C4.5 untuk kasus penentuan rekomendasi Pelatihan *Readiness to Change*.

Pohon keputusan yang diperoleh dari konfigurasi terbaik sub sistem PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian MDLP dapat dilihat pada Lampiran 9. Pohon keputusan tersebut memiliki root node PC2. Jika diubah menjadi aturan, pohon keputusan tersebut akan menghasilkan 129 aturan.

Tabel 6.17 Perbandingan metode untuk rekomendasi pelatihan Readiness to Change

NO	Metode	Akurasi (0,3)	Presisi (0,2)	Recall (0,3)	F-Measure (0,2)	Nilai	Jml Atribut
1	C4.5	0.829	0.813	0.848	0.830	0.831	14
2	PCA dan C4.5	0.499	0.497	0.229	0.314	0.381	2
3	PCA, diskritisasi, dan C4.5	0.903	**0.902**	0.902	0.902	0.902	13
4	PCA, diskritisasi dg MDLP, dan C4.5	**0.904**	0.895	**0.915**	**0.905**	**0.905**	7

Tabel 6.18 Nilai eigen hasil analisis PCA untuk data pelatihan Readiness to Change

Principal Component	Nilai awal eigen		
	Total	% dari variansi	Cumulative %
1	7,544	53,89	53,89
2	1,804	12,89	66,78
3	0,772	5,52	72,30
4	0,651	4,65	76,95
5	0,492	3,51	80,46
6	0,474	3,38	83,85
7	**0,428**	**3,06**	**86,91**
8	0,380	2,72	89,62
9	0,354	2,53	92,15
10	0,289	2,06	94,21
11	0,257	1,83	96,05
12	0,230	1,65	97,69
13	0,192	1,37	99,06
14	0,131	0,94	100

Konfigurasi untuk merekomendasikan pelatihan *Readiness to Change* diperoleh dengan menggunakan 7 PC. Nilai eigen untuk PC tersebut dapat dilihat pada Tabel 6.18. Tabel tersebut menunjukkan bahwa ketika yang digunakan adalah 7 PC maka PC tersebut dapat menjelaskan 86,91 % dari total variansi yang ada seperti terlihat pada kolom *cumulative* pada PC ke-7.

Tabel 6.19 Nilai hubungan variabel dan PC terkait pelatihan Readiness to Change

Vari-	Principal Component						
abel	1	2	3	4	5	6	7
pk	**0.670**	0.232	0.098	0.361	0.163	0.296	0.334
dk	0.521	0.184	0.201	0.718	0.011	0.008	0.154
ks	0.095	**0.845**	0.001	0.060	0.241	0.089	-0.005
pd	-0.033	**0.776**	0.171	0.326	0.098	0.279	0.172
sk	0.213	**0.708**	0.191	0.344	0.117	0.122	0.077
se	0.319	0.615	0.123	0.116	0.535	0.021	-0.007
kp	0.110	0.172	**0.931**	0.238	0.097	0.107	0.051
if	-0.005	0.171	0.140	**0.823**	0.179	0.321	0.123
hb	0.042	0.163	0.063	**0.811**	0.115	0.284	0.253
fb	0.242	0.273	0.143	**0.781**	0.147	-0.031	0.179
da	0.470	0.207	0.209	**0.715**	0.092	0.095	0.179
ts	0.038	0.406	0.093	0.228	**0.809**	0.180	0.101
kd	0.208	0.372	0.150	0.330	0.186	**0.758**	0.053
kn	0.219	0.072	0.058	0.439	0.071	0.052	**0.846**

Variabel asli memiliki hubungan dengan PC (*principal component*) yang nilai hubungannya dapat diketahui pada hasil rotasi faktor yang terlihat pada Tabel 6.19. Variabel asli yang memiliki hubungan yang kuat dengan PC adalah yang nilainya di atas 0,71 (lihat area yang dihitamkan), tetapi jika menggunakan kriteria nilai *loading* di atas 0,71 akan ada 1 PC yang tidak diketahui variabel aslinya. Oleh karena itu, untuk pelatihan ini digunakan kriteria nilai loading yaitu 0,63 yang oleh Tabachnick dan Fidell (2013) dinyatakan menggambarkan hubungan yang sangat baik. Daftar variabel asli yang memiliki pengaruh kuat terhadap masing-masing PC dapat dilihat pada Tabel 6.20.

Tabel 6.20 menunjukkan variabel asli yang memiliki hubungan kuat dengan *principal component*. PC1 dipengaruhi oleh variabel potensi kecerdasan dengan nilai *loading* 0,670. PC2 sangat dipengaruhi oleh variabel kerjasama dan penyesuaian diri. PC2 dipengaruhi juga oleh variabel sistematika kerja dengan nilai *loading* 0,708. PC3 sangat dipengaruhi oleh variabel kepemimpinan. PC4 sangat dipengaruhi oleh variabel inisiatif, hasrat berprestasi, fleksibilitas berpikir, dan daya analisis. PC5 sangat

dipengaruhi oleh variabel toleransi terhadap stress, sedangkan PC6 sangat dipengaruhi oleh variabel kepercayaan diri. PC7 sangat dipengaruhi oleh variabel kemampuan numerik. Ada 10 variabel yang sangat berpengaruh kuat terhadap penentuan rekomendasi pelatihan *Readiness to Change*, 2 variabel yang mempunyai pengaruh kuat dan ada 2 variabel yang tidak memiliki pengaruh kuat dengan proses tersebut yaitu variabel daya konseptual dan stabilitas emosi.

Tabel 6.20 Hubungan variabel asli dan PC terkait pelatihan Readiness to Change

PC	Variabel Asli	Nilai Loading
PC1	pk (Potensi Kecerdasan)	0,670
PC2	ks (Kerjasama) pd (Penyesuaian Diri) sk (Sistematika Kerja)	0,845 0,776 0,708
PC3	kp (Kepemimpinan)	0,931
PC4	if (Inisiatif) hb (Hasrat Berprestasi) fb (Fleksibilitas Berpikir) da (Daya Analisis)	0,823 0,811 0,781 0,715
PC5	ts (Toleransi terhadap Stress)	0,809
PC6	kd (Kepercayaan Diri)	0,758
PC7	kn (Kemampuan Numerik)	0,846

6.2.6 Pelatihan Team Building

Pengujian pada pelatihan *Team Building* dilakukan dengan 451 data. Pada data yang digunakan tidak dilakukan *over-sampling* karena kelas datanya cukup seimbang dengan jumlah data untuk kelas Ya adalah 234 data dan jumlah data untuk kelas Tdk adalah 217 data.

Hasil pemrosesan data tersebut dengan 2 sub sistem, sub sistem PCA dan C4.5 serta sub sistem PCA, diskritisasi, dan C4.5 untuk penentuan rekomendasi pelatihan *Team Building* dapat dilihat pada Lampiran 10. Lampiran 10 menunjukkan bahwa nilai total tertinggi saat menggunakan metode PCA dan C4.5 diperoleh saat menggunakan 9 PC. Nilai total tertinggi saat menggunakan metode PCA, diskritisasi dengan kriteria

pemberhentian interval dan C4.5 diperoleh saat menggunakan 12 PC. Nilai total tertinggi saat menggunakan metode PCA, diskritisasi dengan kriteria pemberhentian MDLP dan C4.5 diperoleh saat menggunakan 6 dan 7 PC. Hasil pemrosesan terbaik tersebut kemudian dibandingkan dengan sub sistem yang lain.

Performa terbaik dari masing-masing sub sistem dibandingkan dan terlihat pada Tabel 6.21 bahwa performa terbaik diberikan oleh sub sistem PCA dan C4.5. Nilai *recall* dan *F-Measure* tertinggi dihasilkan oleh sub sistem tersebut. Sub sistem tersebut memberikan hasil terbaik saat menggunakan 9 PC.

Tabel 6.21 Hasil pembandingan performa antar sub sistem untuk pelatihan Team Building

Sub Sistem	Akurasi	Presisi	Recall	F–measure	Nilai	Jml Atribut
C4.5	0.563	0.579	0.398	0.472	0.499	14
PCA dan C4.5	0.536	0.529	**0.667**	**0.590**	**0.585**	9
PCA, diskritisasi, dan C4.5	**0.566**	**0.596**	0.416	0.490	0.512	12
PCA, diskritisasi dg MDLP, dan C4.5	0.523	0.533	0.385	0.447	0.468	8

Pohon keputusan yang diperoleh dari konfigurasi terbaik sub sistem PCA dan C4.5 dapat dilihat pada Lampiran 11. Pohon keputusan tersebut memiliki root node PC8. Jika diubah menjadi aturan, pohon keputusan tersebut akan menghasilkan 39 aturan.

Konfigurasi untuk merekomendasikan pelatihan *Team Building* diperoleh dengan menggunakan 9 PC. Nilai eigen untuk PC tersebut dapat dilihat pada Tabel 6.22. Tabel 6.22 menunjukkan bahwa ketika yang digunakan adalah 9 PC maka PC tersebut dapat menjelaskan 91,62 % dari total variansi yang ada seperti terlihat pada kolom *cumulative* pada PC ke-9.

Tabel 6.22 Nilai eigen hasil analisis PCA untuk data pelatihan Team Building

Principal Component	Nilai awal eigen		
	Total	% dari variansi	Cumulative %
1	7,493	53,52	53,52
2	1,672	11,59	65,47
3	0,842	6,01	71,48
4	0,643	4,59	76,08
5	0,531	3,80	79,87
6	0,452	3,23	83,10
7	0,422	3,02	86,12
8	0,407	2,91	89,03
9	**0,363**	**2,60**	**91,62**
10	0,329	2,35	93,97
11	0,283	2,02	95,99
12	0,227	1,62	97,62
13	0,205	1,47	99,09
14	0,127	0,91	100,00

Tabel 6.23 Nilai hubungan variabel asli dan PC terkait pelatihan Team Building

Vari-abel	Principal Component							
	1	2	3	4	5	6	7	8
fb	**0,769**	0,146	0,200	0,034	0,275	0,158	0,245	0,144
da	**0,752**	0,141	0,184	0,165	0,071	0,063	0,456	0,104
if	**0,740**	0,012	0,183	0,360	0,198	0,176	0,182	0,115
dk	**0,746**	0,194	0,220	0,141	0,117	0,033	0,432	0,104
hb	0,695	0,030	0,046	0,410	0,241	0,160	0,266	0,032
se	0,162	**0,884**	0,056	0,184	0,139	0,213	0,102	0,240
kp	0,307	0,052	**0,907**	0,156	0,112	0,103	0,114	0,003
kd	0,298	0,175	0,154	**0,834**	0,105	0,128	0,143	0,144
pd	0,234	0,188	0,118	0,518	0,500	0,165	0,089	0,380
sk	0,361	0,133	0,123	0,147	**0,797**	0,149	0,238	0,156
ts	0,182	0,251	0,126	0,185	0,171	**0,830**	0,107	0,286
kn	0,343	0,024	0,003	0,120	0,148	0,193	**0,825**	-0,022
pk	0,417	0,113	0,168	0,094	0,128	-0,055	**0,758**	0,156
ks	0,140	0,222	-0,005	0,166	0,158	0,223	0,074	**0,888**

Variabel asli memiliki hubungan dengan PC (*principal component*) yang nilai hubungannya dapat diketahui pada hasil rotasi faktor yang terlihat pada Tabel 6.23. Variabel asli yang memiliki hubungan yang kuat dengan PC adalah yang nilainya di atas 0,71 (lihat area yang dihitamkan). Daftar variabel asli yang memiliki pengaruh kuat terhadap masing-masing PC dapat dilihat pada Tabel 6.24.

Tabel 6.24 Hubungan variabel asli dan PC terkait pelatihan Team Bulding

PC	Variabel Asli	Nilai loading
PC1	fb (Fleksibilitas Berpikir) da (Daya Analisis) if (Inisiatif) dk (Daya Konseptual)	0,769 0,752 0,740 0,746
PC2	se (Stabilitas Emosi)	0,884
PC3	kp (Kepemimpinan)	0,907
PC4	kd (Kepercayaan Diri)	0,834
PC5	sk (Sistematika Kerja)	0,797
PC6	ts (Toleransi terhadap Stress)	0,830
PC7	kn (Kemampuan Numerikal)	0,825
PC8	pk (Potensi Kecerdasan)	0,758
PC9	ks (Kerjasama)	0,888

Tabel 6.24 menunjukkan hubungan antara *principal component* dan variabel aslinya. PC1 sangat dipengaruhi oleh variabel fleksibilitas berpikir, daya analisis, inisiatif, dan daya konseptual yang memiliki nilai loading di atas 0,7. PC2 sangat dipengaruhi oleh variabel stabilitas emosi, sedangkan PC3 sangat dipengaruhi oleh variabel kepemimpinan. PC4 sangat dipengaruhi oleh variabel kepercayaan diri dan PC5 sangat dipengaruhi oleh variabel sistematika kerja. PC6 sangat dipengaruhi oleh variabel toleransi terhadap stress, sedangkan PC7 sangat dipengaruhi oleh variabel kemampuan numerikal. PC8 sangat dipengaruhi oleh variabel potensi kecerdasan, dan PC9 sangat dipengaruhi oleh variabel kerjasama. Berdasar tabel tersebut, ada 12 variabel yang berpengaruh kuat terhadap penentuan rekomendasi pelatihan *Team*

Building dan ada 2 variabel yang tidak memiliki pengaruh kuat dengan proses tersebut yaitu variabel penyesuaian diri dan hasrat berprestasi.

6.3 Hasil Pengujian Keseluruhan Model

Pengujian keseluruhan model ini dilakukan pada bagian *Front-End*. Pengujian ini dilakukan untuk menguji kemampuan model yang telah dibangun pada bagian *Back-end* secara menyeluruh, tidak hanya per pelatihan saja. Pengujian dilakukan dengan membandingkan hasil rekomendasi pelatihan yang diberikan oleh assessor dan rekomendasi yang diberikan oleh model. Pengukuran performanya dilakukan sesuai dengan aturan yang telah dijelaskan pada sub bab 4.4.2.

Data yang akan digunakan untuk pengujian diambil dari data pemetaan pegawai yang sudah dibersihkan dari *outlier*. Dari data yang keseluruhannya berjumlah 451 tersebut kemudian diambil 240 data secara acak. Selanjutnya 240 data tersebut dibagi secara acak menjadi 3 bagian sehingga dihasilkan 3 buah dataset. Hasil pengujian menggunakan 3 dataset ini dapat dilihat pada Tabel 6.25.

Tabel 6.25 Hasil pengujian keseluruhan model

NO	Dataset	Akurasi (%)	Presisi (%)	Recall (%)	F-Measure(%)
1	Dataset 1	87.5	86.52	74.85	80.26
2	Dataset 2	87.71	84.46	77.64	80.91
3	Dataset 3	90.63	88.08	83.13	85.53

Tabel 6.25 menunjukkan bahwa performa terbaik didapatkan dari pengujian pada *dataset* 3 dengan nilai akurasi 90,63% persen, presisi 88,08%, recal 83,03% dan F-Measure 85,53%, sedangkan performa terburuk didapatkan dari pengujian pada *dataset* 1 dengan nilai akurasi 87,5%, presisi 86,52%, recall 74,85% dan F-Measure 80,26%. Jika dirata-rata, nilai akurasi, presisi, recall, dan *F-measure* dari pengujian terhadap keseluruhan *dataset* tersebut adalah 88,61%, 86,35%, 78,54%, dan 82,23%. Contoh hasil pengujian keseluruhan model dapat dilihat pada Tabel 6.26.

Tabel 6.26 Hasil rekomendasi pelatihan

NO	ID	Rekomendasi Pelatihan		Akurasi	F-Meas
		Assessor	Model		
1	JKTP047	Effective Comm. Skill Personnel Effective. Team Building **AMT**	Effective Comm. Skill Personnel Effective. Team Building	83,3%	85,7%
2	JKTB141	Personnel Effective.	Personnel Effective.	100%	100%
3	JKTP011	Personnel Effective. **AMT**	Personnel Effective.	83,3%	66,7%
4	JKTB068	Personnel Effective. **Team Building**	Personnel Effective.	83,3%	66,7%
5	JOGJA049	Human Skill Improve. Personnel Effective. **Team Building**	Human Skill Improve. Personnel Effective.	83,3%	80%
6	JOGJA114	Effective Comm. Skill Personnel Effective. **Team Building**	Effective Comm. Skill Personnel Effective.	83,3%	80%
7	JKTM043	Effective Comm. Skill	Effective Comm. Skill	100%	100%
8	JOGJA140	Effective Comm. Skill Personnel Effective. **Team Building**	Effective Comm. Skill Personnel Effective. AMT	66,7%	66,7%
9	JKTM028	AMT **Team Building**	AMT Personnel Effective.	66,7%	50%
10	JKTB014	Personnel Effective. **Team Building**	Personnel Effective.	83,3%	66,7%
11	JOGJA041	Personnel Effective. **Team Building**	Personnel Effective.	83,3%	66,7%

Banyak kesalahan dalam rekomendasi pelatihan Team Building seperti terlihat pada Tabel 6.26. Pada nomor 4 sampai dengan nomor 11 terlihat sistem tidak memberikan rekomendasi pelatihan Team Building yang seharusnya diberikan. Hal tersebut terjadi karena rendahnya akurasi dari sistem untuk merekomendasikan pelatihan Team Building seperti terlihat pada Tabel 6.21.

Penyebab rendahnya akurasi rekomendasi Team Building adalah ketidakkonsistenan data pelatihan Team Building. Contoh ketidakkonsistenan yang terjadi dapat dilihat pada Tabel 6.27 Untuk melihat ketidakkonsistenan yang terjadi, nilai aspek psikologis yang perlu diperhatikan adalah nilai kerjasama (KS). Nilai kerjasama menjadi pertimbangan utama dalam penentuan pelatihan Team Building. Data nomor 4 pada Tabel 6.27 memperlihatkan rekomendasi yang berbeda untuk pegawai dengan nilai kerjasama 3-. Ada 47 pegawai dengan nilai kerjasama sebesar 3- yang direkomendasikan untuk mengikuti pelatihan Team Building dan ada 36 pegawai nilai yang sama tetapi tidak direkomendasikan mengikuti pelatihan Team Building. Jika yang dilihat adalah nilai kerjasama dan juga pembulatan dari rerata keseluruhan nilai maka akan terlihat ketidakkonsistenan juga. Misal pada data nomor 3, ada 32 pegawai dengan nilai aspek kerjasama 2+ dan rerata nilai aspek psikologis 3 yang direkomendasikan mengikuti pelatihan Team Building dan ada 14 pegawai dengan nilai yang sama tetapi tidak direkomendasikan mengikuti pelatihan Team Building.

Tabel 6.27 Inkonsistensi data pelatihan Team Building

NO	Nilai Aspek KS	Kelas Ya			Kelas Tdk		
		Rerata Nilai		Jumlah	Rerata Nilai		Jumlah
		2	3		2	3	
1	2-	2	1	3	1	0	1
2	2	15	7	22	9	5	14
3	2+	14	32	46	4	14	18
4	3-	6	41	47	5	31	36
5	3	15	67	82	10	113	123

6.4 Pembahasan Hasil Pengujian

Dari pengujian performa model yang telah lakukan dapat didapatkan kesimpulan seperti yang terlihat pada Tabel 6.28. Pada tabel tersebut terlihat bahwa metode PCA, diskritisasi, dan C4.5 memberikan performa terbaik untuk kelima jenis

pelatihan. Metode PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian diskritisasi menggunakan kriteria jumlah interval memberikan performa yang lebih baik daripada ketiga metode lainnya saat digunakan untuk penentuan rekomendasi pelatihan *Effective Communication Skill, Human Skil Improvement, dan Personnel Effectiveness*. Metode PCA, diskritisasi, dan C4.5 dengan kriteria pemberhentian diskritisasi menggunakan kriteria MDLP memberikan performa yang lebih baik daripada ketiga metode lainnya saat digunakan untuk penentuan rekomendasi pelatihan AMT, dan *Readiness to Change*. Metode PCA dan C4.5 memberkan performa yang lebih baik daripada ketiga metode lainnya saat digunakan untuk penentuan rekomendasi pelatihan *Team Building*.

Tabel 6.28 Jenis-jenis pelatihan dan metode penentuan rekomendasinya

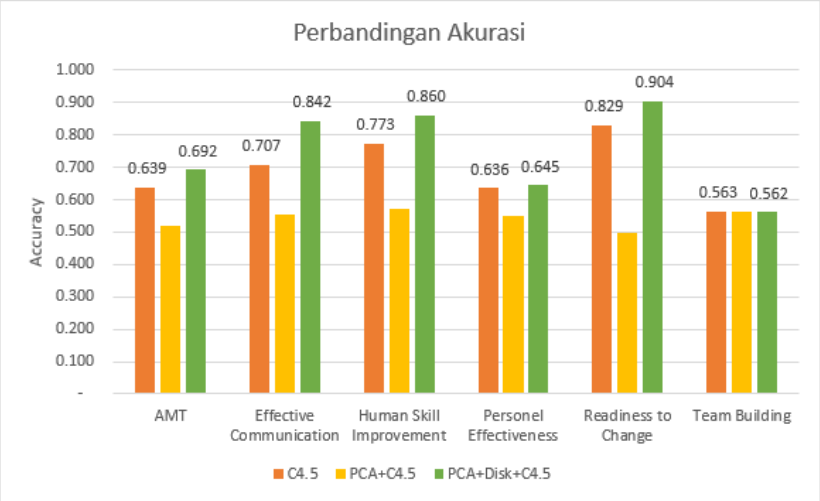
Pelatihan	Metode	Akurasi	Presisi	Recall	F-Measure	Jml Atribut
AMT	PDC.b	0.692	0.676	0.597	0.649	9
Effective Communication Skill	PDC.a	0.845	0.813	0.859	0.835	13
Human Skill Improvement	PDC.a	0.860	0.852	0.852	0.852	11
Personnel Effectivenes	PDC.a	0.645	0.650	0.745	0.695	10
Readiness To Change	PDC.b	0.904	0.895	0.915	0.905	7
Team Building	PDC.a	0.566	0.596	0.416	0.490	12

Keterangan :

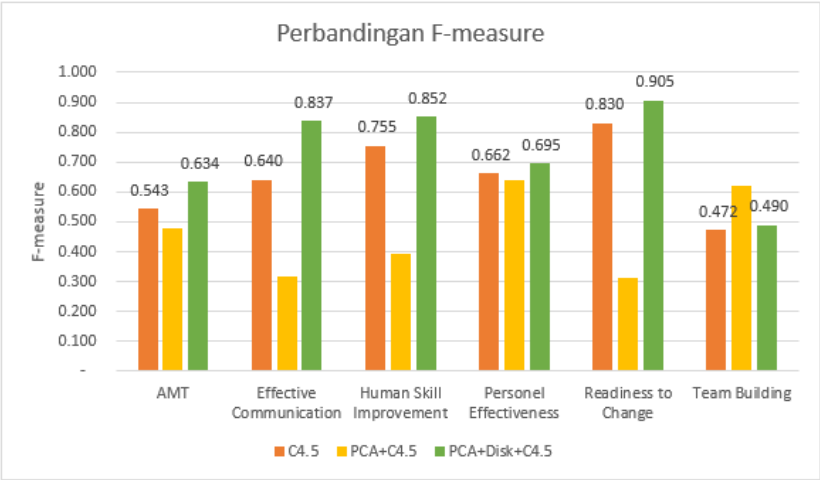
*PDC.a : PCA, diskritisasi dengan kriteria pemberhentian jumlah interval, dan C4.5

*PDC.b : PCA, diskritisasi dengan kriteria pemberhentian MDLP, dan C4.5

Gambar 6.2 menunjukkan bahwa metode PCA, diskritisasi, dan C4.5 menunjukkan nilai akurasi yang lebih baik dibanding kedua metode lainnya kecuali di pelatihan Team Building. Terlihat juga bahwa nilai akurasi metode PCA dan C4.5 berada di bawah 60% untuk semua pelatihan.



Gambar 6.2 Grafik perbandingan nilai akurasi



Gambar 6.3 Grafik perbandingan nilai F-Measure

Gambar 6.3 menunjukkan bahwa metode PCA, diskritisasi, dan C4.5 menunjukkan nilai *F-Measure* yang lebih tinggi dibanding kedua metode lainnya kecuali pada data pelatihan *Team Building*. Pada data pelatihan *Team building*, nilai *F-Measure* metode tersebut kalah tinggi dengan nilai *F-Measure* yang dihasilkan oleh metode PCA dan C4.5. Tabel perbandingan nilai akurasi dan *F-Measure* secara lengkap dapat dilihat pada Tabel 6.29.

Tabel 6.29 Rincian nilai perbandingan performa

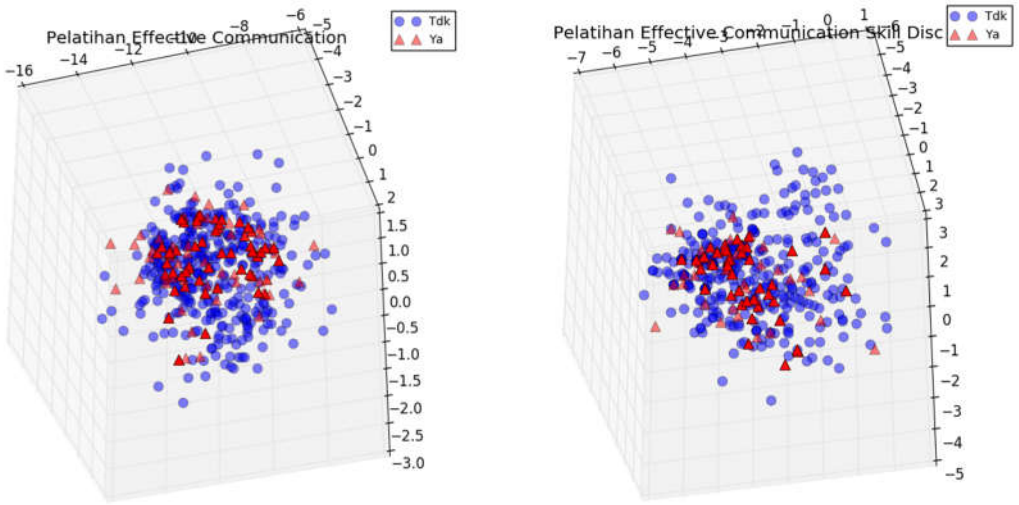
NO	Pelatihan	Akurasi			F-measure			Selisih Akurasi
		A	B	C	A	B	C	A vs C
1	AMT	0.639	0.518	0.692	0.543	0.479	0.634	5.339
2	Effective Communication	0.707	0.555	0.842	0.640	0.315	0.837	13.485
3	Human Skill Improvement	0.773	0.573	0.860	0.755	0.394	0.852	8.633
4	Personel Effectiveness	0.636	0.550	0.645	0.662	0.638	0.695	0.948
5	Readiness to Change	0.829	0.497	0.904	0.830	0.314	0.905	7.500
6	Team Building	0.563	0.564	0.562	0.472	0.623	0.490	(0.137)

[A] Metode C4.5

[B] Metode PCA dan C4.5

[C] Metode PCA, diskritisasi, dan C4.5

Tabel 6.29 menunjukkan selisih akurasi antara metode C4.5 dan metode PCA, diskritisasi, dan C4.5 yang paling besar terjadi pada pelatihan *Effective Communication Skill* dengan selisih akurasi 13,485%. Selain nilai akurasi, selisih nilai F-Measure juga besar dengan nilai 19,7%.



a. Tanpa ekstraksi fitur dan diskritisasi b. Dengan ekstraksi fitur dan diskritisasi

Gambar 6.4 Visualisasi data pelatihan Effective Communication Skill

Gambar 6.4 pada bagian a menunjukkan visualisasi data pelatihan *Effective Communication Skill* tanpa dilakukan ekstraksi fitur dan diskritisasi. Pada Gambar 6.4

bagian a terlihat datanya mengumpul tetapi tercampur antara kelas Ya dan kelas Tidak. Gambar 6.4 bagian b menunjukkan visualisasi data pelatihan *Effectiveness Communication Skill* dengan dilakukan ekstraksi fitur dan diskritisasi. Gambar grafik tersebut memperlihatkan data yang lebih tersebar, ketercampuran kelas Ya dan kelas Tidak sudah agak terurai. Kelas Ya pun cenderung lebih memgumpul. Kondisi tersebut akan mempermudah pemisahan kelas Ya dan kelas Tidak sehingga berimbas pada meningkatkan performa dari *classifier* yang digunakan.

Hasil pengujian metode PCA, diskritisasi, dan C4.5 juga dibandingkan terhadap penggunaan metode C4.5 yang dilakukan dengan aplikasi *data mining*. Aplikasi *data* mining yang digunakan adalah WEKA versi 3.8 dan TANAGRA. Algoritma C4.5 pada WEKA yang disebut algoritma J4.8 digunakan untuk membentuk pohon keputusan dengan *pruning* dan tanpa *pruning*. Aplikasi TANAGRA juga digunakan untuk membuat pohon keputusan dengan algoritma C4.5 dengan *pruning*. Nilai performa hasil pengujian menggunakan kedua aplikasi tersebut beserta pembandingannya dengan metode PCA, diskritisasi, dan C4.5 dapat dilihat pada Tabel 6.30, sedangkan pembandingan kompleksitas pohon keputusan berupa jumlah *leaf node* dan jumlah *node* yang terbentuk dapat dilihat pada Tabel 6.31.

Tabel 6.30 Perbandingan performa metode PCA, diskritisasi, dan C4.5

NO	Pelatihan	C4.5/J4.8 WEKA				C4.5 TANAGRA		PCA, diskritisasi, & C4.5	
		Tanpa Pruning		Dengan Pruning		Dengan Pruning			
		Akurasi	F-measure	Akurasi	F-measure	Akurasi	F-measure	Akurasi	F-measure
1	AMT	0.663	0.612	0.707	0.663	0.698	0.692	0.692	0.634
2	Effective Communication	0.841	0.836	0.817	0.817	0.849	0.848	0.842	0.837
3	Human Skill Improvement	0.861	0.862	0.853	0.855	0.929	0.929	0.860	0.852
4	Personel Effectiveness	0.696	0.718	0.668	0.703	0.679	0.674	0.645	0.695
5	Readiness to Change	0.899	0.902	0.891	0.894	0.894	0.837	0.904	0.905
6	Team Building	0.557	0.559	0.585	0.547	0.544	0.545	0.562	0.490

Tabel 6.30 menunjukkan bahwa performa metode PCA, diskritisasi, dan C4.5 hampir sama dengan performa algoritma C4.5 yang dilakukan dengan *pruning*. Misal pada pelatihan AMT, terlihat bahwa akurasi yang dihasilkan algoritma C4.5 dengan *pruning* menggunakan program WEKA adalah 70,7% dan menggunakan program

TANAGRA adalah 69,8%. Akurasi yang dihasilkan dengan metode PCA, diskritisasi, dan C4.5 adalah 69,2%, mendekati akurasi algoritma C4.5 dengan *pruning*.

Tabel 6.31 Perbandingan kompleksitas metode PCA, diskritisasi, dan C4.5

NO	Pelatihan	C4.5/J4.8 WEKA				C4.5 TANAGRA Dengan Pruning		PCA, diskritisasi, & C4.5	
		Tanpa Pruning		Dengan Pruning					
		Jml Leaf	Jml Node	Jml Leaf	Jml Node	Jml Leaf	Jml Node	Jml Leaf	Jml Node
1	AMT	676	721	76	81	72	80	33	54
2	Effective Communication	586	625	226	241	148	164	165	183
3	Human Skill Improvement	601	641	196	209	164	181	197	217
4	Personel Effectiveness	811	865	196	209	177	196	275	298
5	Readiness to Change	421	449	181	193	126	139	129	180
6	Team Building	796	849	151	161	132	146	218	236

Selain menghasilkan performa yang hampir sama, kompleksitas pohon keputusan yang dihasilkan pun pada beberapa pelatihan hampir sama seperti terlihat pada Tabel 6.31. Misal pada pelatihan *Readiness to Change*, jumlah *leaf node* yang dihasilkan algoritma C4.5 dengan *pruning* menggunakan WEKA adalah 181 dan menggunakan TANAGRA adalah 126. Jumlah *leaf node* yang dihasilkan metode PCA, diskritisasi, dan C4.5 adalah 129, selisih 3 *leaf node* jika dibandingkan dengan pohon keputusan hasil program TANAGRA. Contoh lain pada pelatihan *Human Skill Improvement*. Jumlah *leaf node* yang dihasilkan algoritma C4.5 dengan *pruning* menggunakan WEKA adalah 196 dan menggunakan TANAGRA adalah 209. Jumlah *leaf node* yang dihasilkan metode PCA, diskritisasi, dan C4.5 adalah 197, selisih 1 *leaf node* jika dibandingkan dengan pohon keputusan hasil program WEKA dan selisih 12 *node* jika dibandingkan dengan hasil program TANAGRA. Berdasar hasil tersebut, metode PCA, diskritisasi, dan C4.5 dapat dikatakan sebagai sebuah cara alternatif untuk melakukan *pruning* terhadap pohon keputusan.

BAB VII
PENUTUP

7.1 Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah

1. Penentuan rekomendasi pelatihan pengembangan diri bagi pegawai negeri sipil berdasar data pemetaan pegawai dapat dilakukan dengan menggunakan algoritma C4.5 yang dikombinasikan dengan PCA dan diskritisasi dengan rerata nilai akurasi adalah 86,61%, rerata nilai presisi 86,35%, rerata nilai recall 78,54% dan rerata nilai *F-measure* 82,23%.
2. Pengujian untuk penentuan rekomendasi pelatihan AMT menunjukkan nilai akurasi 66,6% dan *F-Measure* 59,9%, pelatihan *Effective Communication Skill* menunjukkan nilai akurasi 84,2% dan nilai *F-Measure* 83,7%, pelatihan *Human Skill Improvement* menunjukkan akurasi 85% dan nilai *F-Measure* 84,6%, pelatihan *Personnel Effectiveness* menunjukkan nilai akurasi 63,9% dan nilai *F-Measure* 47,1%, pelatihan *Readiness to Change* menunjukkan akurasi 88,5% dan nilai *F-Measure* 88,8%, dan pelatihan *Team Building* menunjukkan nilai akurasi 56,3% dan F-Measure 47,2%.
3. Pendekatan yang diusulkan dengan menggabungkan algoritma PCA, diskritisasi, dan algoritma C4.5 menunjukkan performa yang lebih baik daripada menggunakan algoritma C4.5 dan PCA, serta algoritma C4.5 saja untuk kasus penentuan rekomendasi pelatihan pengembangan diri. Hal tersebut terbukti dengan semua jenis pelatihan yang mendapatkan rekomendasi terbaik dari metode PCA, diskritisasi, dan C4.5.
4. Metode PCA, diskritisasi, dan C4.5 dapat menjadi alternatif untuk melakukan *pruning* terhadap pohon keputusan terlihat dari performa dan kompleksitas

pohon keputusan yang terbentuk yang tidak jauh berbeda dengan hasil algoritma C4.5 dengan *pruning*.

7.2 Saran

Beberapa saran pengembangan yang dapat diberikan untuk penelitian selanjutnya adalah

1. Proses klasifikasi dapat dilakukan dengan classifier lain misal SVM, KNN, atau JST dan mengkombinasikannya dengan PCA dan diskritisasi.
2. Proses *data cleaning* perlu dioptimalkan lagi untuk mendapatkan performa yang lebih baik.

DAFTAR PUSTAKA

Alvarez, M.A., Carrasco, J.A. & Martinez, J.F., 2013. Combining Techniques to Find the Number of Bins for Discretization. In *32nd International Conference of the Chilean Computer Science Society*. Temuco, pp. 54–57.

Amin, A. et al., 2016. Comparing Oversampling Techniques to Handle the Class Imbalance Problem : A Customer Churn Prediction Case Study. , 4(Ml).

Badan Kepegawaian Negara, 2011, Peraturan Kepala BKN Nomor 23 Tahun 2011 Tentang Pedoman Penilaian Kompetensi PNS, Jakarta.

Ben-gal, I., 2010. Outlier Detection. In O. Maimon & L. Rokach, eds. *Data Mining and Knowledge Discovery Handbook*. New York: Springer, pp. 117–127.

Chawla, N. V, 2003. C4.5 and Imbalanced Datasets : Investigating The Effect Of Sampling Method , Probabilistic Estimate , and Decision Tree Structure. In Washington.

Chawla, N. V et al., 2002. SMOTE : Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, pp.321–357.

Chen, K.K. et al., 2007. Constructing A Web-Based Employee Training Expert System With Data Mining Approach. *Proceedings - The 9th IEEE International Conference on E-Commerce Technology; The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services, CEC/EEE 2007*, pp.659–664.

Chu, X. et al., 2016. Data Cleaning : Overview and Emerging Challenges. In *Proceedings of the 2016 International Conference on Management of Data*. San Francisco, pp. 16–21.

Dash, R., Paramguru, R.L. & Dash, R., 2011. Comparative Analysis of Supervised and Unsupervised Discretization Techniques. *International Journal of Advances in Science and Technology*, 2(3), pp.29–37.

Doja, M.N., Jain, S., Alam, M.A., 2012. SORA: An Application Of Scaled K-Means To Remove Outliers On Multidimensional Dataset. *International Journal of Computer Application and Engineering Technology*, 1(3), pp.77–84.

Fayyad, U.M. & Irani, K.B., 1993. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In *Proceddings of 13th International*

Conference on Artificial Inatelligence. pp. 1022–1027. Available at: http://www.decom.ufop.br/luiz/site_media/uploads/arquivos/bcc444_pcc142/multiintervaldiscretizationofcontinuousvaluedattributesforclassificationlearning1993.pdf.

Hacibeyoglu, M., Arslan, A. & Kahramanli, S., 2011. Improving Classification Accuracy with Discretization on Datasets Including Continuous Valued Features. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 5(6), pp.555–558.

Han, J., Kamber, M. & Pei, J., 2012. *Data Mining: Concepts and Techniques* 3rd ed., San Francisco: Morgan Kaufmann Publishers.

Hussain, A., Rao, M.K. & Mahmood, A.M., 2013. An Optimized Approach To Generate Simplified Decision Trees. In *IEEE International Conference on Computational Intelligence and Computing Research*. Tamilnadu: IEEE.

Jantan, H., Hamdan, A.R. & Othman, Z.A., 2011. Talent Knowledge Acquisition using Data Mining Classification Techniques. In *Conference on Data Mining and Optimization*. Selangor, pp. 32–37.

Julie Grisanti, 2016. Decision Trees: An Overview. *Aunalytics*, pp.1–3. Available at: http://www.aunalytics.com/decision-trees-an-overview/ [Accessed October 10, 2016].

Kantor Regional I BKN, 2011, Laporan Pemetaan Pegawai Kantor Regional I BKN, Yogyakarta.

Kareem, I.A. & Duaimi, M.G., 2014. Improved Accuracy for Decision Tree Algorithm Based on Unsupervised Discretization. *International Journal of Computer Science and Mobile Computing*, 3(6), pp.176–183.

Khalid, S., Khalil, T. & Nasreen, S., 2014. A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning. In *Science and Information Conference*. London, pp. 372–378.

Krishnan, S. et al., 2015. SampleClean : Fast and Reliable Analytics on Dirty Data. *IEEE Computer Society Technical Comittee on Data Engineering*, pp.59–75.

Larose, D.T. & Larose, C.D., 2015. *Data Mining And Predictive Analytics* 2nd ed., New Jersey: John Wiley & Sons, Inc.

Larose, D.T. & Larose, C.D., 2014. *Discovering Knowledge in data An Introduction to Data Mining* 2nd ed., New Jersey: John Wiley & Sons, Inc.

Last, M. & Kandel, A., 2001. Automated Detection of Outliers in Real-World Data. In *The Second International Conference on Intelligent Technologies*. pp. 292–301.

Li, L. et al., 2014. The Application of Decision Tree Algorithm in the Employment Management System. *Applied Mechanics and Materials*, 543–547, pp.1639–1642.

Maimon, O. & Rokach, L., 2010. *Data Mining and Knowledge Discovery Handbook* 2nd ed., New York: Springer. Available at: http://dx.doi.org/10.1007/0-387-25465-x_2%5Cnhttp://link.springer.com/content/pdf/10.1007/978-0-387-09823-4.pdf.

Martinez, A.M. & Kak, A.C., 2001. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), pp.228–233.

Motoda, H. & Liu, H., 1998. Feature selection, extraction and construction. *Communication of IICM*, 5, pp.67–72.

Noe, R.A., 2009. *Employee Training and Development* 5th ed., New York: McGraw-Hill.

Rokach, L. & Maimon, O., 2014. *Data Mining with Decision Trees: Theory and Applications* 2nd ed., Singapore: World Scientific Publishing.

Rokhman, N., Winarko, E. & Subanar, 2016. Improving the Performance of Outlier Detection Methods for Categorical Data By Using Weighting Function. *Journal of Theoretical and Applied Information Technology*, 83(3), pp.327–336.

Santoso, B., Wijayanto, H., Notodipuro, K.A., Sartono, B., 2017. Synthetic Over Sampling Methods for Handling Class Imbalanced Problems : A Review. In *58 th IOP Conference Series: Earth and Environmental Science*. IOP Publishing, pp. 1–8.

Saptarini, N.G.A.P.H., 2012. *Penggunaan Algoritma C4.5 Dan Logika Fuzzy Untuk Klasifikasi Talenta Karyawan(Studi Kasus : Politeknik Negeri Bali)*. Universitas Gadjah Mada.

Sharma, M., Goyal, A., 2015. An Application of Data Mining to Improve Personnel Performance Evaluation in Higher Education Sector In. In *International Conference on Advances in Computer Engineering and Applications (ICACEA)*. Ghaziabad, pp. 559–564.

Smith, L.I., 2002. A tutorial on Principal Components Analysis Introduction. *Statistics*, 51, p.52.

Strohmeier, S. & Piazza, F., 2013. Domain Driven Data Mining in Human Resource Management: A Review of Current Research. *Expert Systems with Applications*, 40(7), pp.2410–2420. Available at: http://dx.doi.org/10.1016/j.eswa.2012.10.059.

Tabachnick, B.G. & Fidell, L.S., 2013. *Using Multivariate Statistics* 6th ed., Boston: Pearson.

Tan, P.N., Steinbach, M. & Kumar, V., 2005. *Introduction To Data Mining*, Boston: Addison-Wesley.

Thornton, G.C. & Rupp, D.E., 2006. *Assessment Centers In Human Resource*, New Jersey: Lawrence Erlbaum Associates.

Wu, X. & Kumar, V., 2009. *The Top Ten Algorithms In Data Mining*, Boca Raton: CRC Press.

Ye, N., 2014. *Data mining*, Boca Raton: CRC Press.

LAMPIRAN

Lampiran 1 Data pemetaan pegawai

REKAPITULASI NILAI
PEMETAAN POTENSI BKN KANREG I YOGYAKARTA

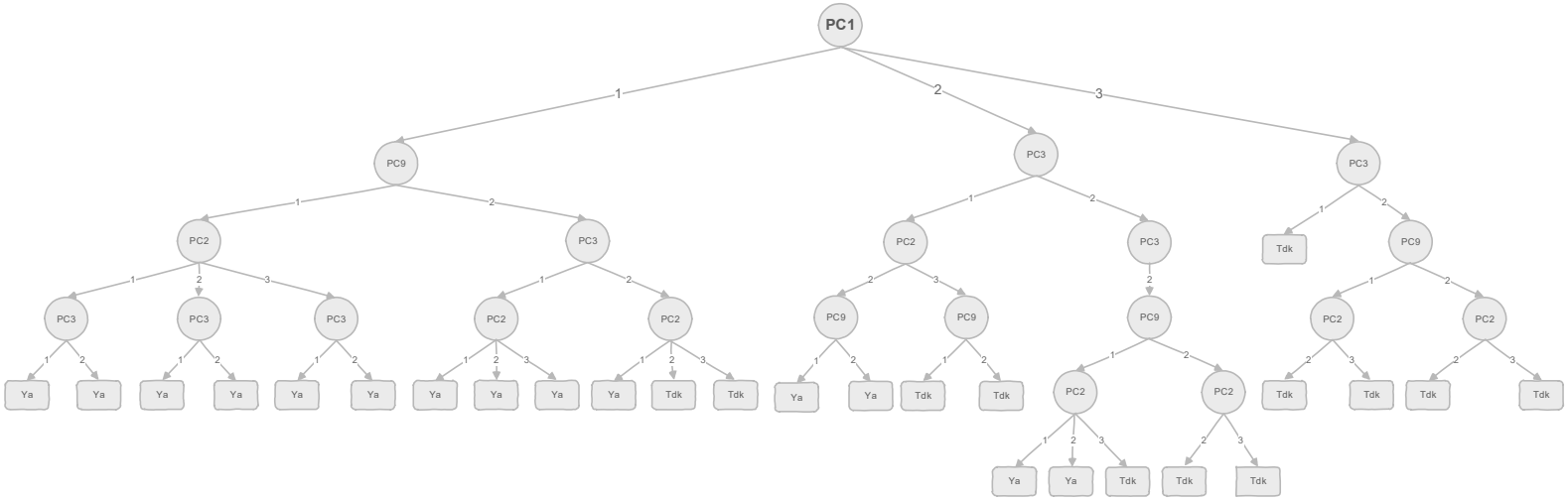
NO	NO. TEST PESERTA	NAMA	KOMPETENSI														PENEMPATAN	PELATIHAN
			Pot. Kec.	Daya Kons.	Daya Anls.	FB	K. Num	Sist. Kerja	Hsrt. Pres	Ins	Stab. Emo	Kpcy Diri	Peny. Diri	KS	Tol stres	Kpp		
1	001	KU	2	2	2+	3-	2	3	3-	3-	3	3-	3	3	3	2+	- Administrasi Rutin - Teknis Kepegawaian	- Personal Eff - Human Skill Improvement - Team Building
2	002	EM	3	2	2	2+	3	3-	2	2	3	2	3	3-	3	1	- Administrasi - Keuangan	- Personal Eff - Achievement Motivation Training - Readiness to Change
3	003	SU	3	2-	2-	2	2	3-	2-	2-	3	2-	2-	2	2	-	- Administrasi Rutin - Tata Naskah	- Personal Eff - Achievement Motivation Training - Readiness to Change
4	004	AG	2	2-	2-	2	2	2	2-	2-	2+	2+	2+	2+	2	-	Pengelola Data Kepegawaian	- Personal Eff - Achievement Motivation Training
5	005	SAF	3	2	3	2+	3-	3	2	3	3	3-	3-	3	3-	2-	- Administrasi Rutin - Teknis Kepegawaian	- Personal Eff - Achievement Motivation Training
6	006	SUN	3	2	3	2+	3-	3	2	3	3	3-	2+	3	3	2+	- Pekerjaan Konseptual - Bimtek	- Personal Eff - Achievement Motivation Training
7	007	SUN	2	1+	2-	2	2-	2+	2-	2-	3	2+	2+	3	3	1	- Administrasi Rutin - Teknis Kepegawaian	- Personal Eff - Achievement Motivation Training - Readiness to Change
8	008	NAV	2	2	2+	2+	2+	3	2	2+	3	3-	3	3	3	1+	Pelayanan Teknis	- Personal Eff - Team Building - Readiness to Change
9	009	RR.	2-	1	1	2	1	2	1	1	1+	1	2	3	1	1-	- Administrasi Rutin Sederhana - Entry Data	- Personal Eff - Achievement Motivation Training - Team Building
10	010	HAR	2-	1	1	2-	2	2+	1	1	3	2	3	3	3-	-	Administrasi Rutin	- Personal Eff - Achievement Motivation Training - Team Building

Lampiran 2 Hasil pengujian untuk pelatihan AMT

PC	PCA dan C4.5					PCA, Diskritisasi (Interval), dan C4.5					PCA, Diskritisasi (MDLP), dan C4.5				
	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total
2 PC	**0.562**	**0.504**	0.291	0.369	0.430	0.605	0.559	0.488	0.521	0.544	0.612	0.573	0.463	0.512	0.540
3 PC	0.555	0.493	0.340	0.402	0.448	0.631	0.595	0.507	0.548	0.570	0.659	0.651	0.488	0.558	0.586
4 PC	0.551	0.486	0.330	0.393	0.440	0.601	0.554	0.483	0.516	0.539	0.651	0.636	0.483	0.549	0.577
5 PC	0.525	0.449	0.350	0.393	0.431	0.610	0.568	0.473	0.516	0.542	0.644	0.626	0.478	0.542	0.570
6 PC	0.514	0.439	0.374	0.404	0.435	0.579	0.528	0.424	0.470	0.500	0.636	0.607	0.488	0.541	0.567
7 PC	0.514	0.443	0.404	0.423	0.449	0.581	0.531	0.424	0.471	0.502	0.644	0.621	0.493	0.549	0.575
8 PC	0.529	0.460	0.399	0.427	0.456	0.586	0.536	0.438	0.482	0.511	0.644	0.621	0.493	0.549	0.575
9 PC	0.512	0.443	0.419	0.430	0.454	0.631	0.594	0.512	0.550	0.572	0.692	0.676	0.597	0.634	0.649
10 PC	0.518	0.457	**0.502**	**0.479**	**0.494**	0.607	0.562	0.493	0.525	0.547	0.659	0.628	0.557	0.590	0.608
11 PC	0.521	0.463	0.461	0.462	0.480	0.612	0.571	0.478	0.520	0.545	0.659	0.628	0.557	0.590	0.608
12 PC	0.516	0.455	0.422	0.438	0.460	0.614	0.571	0.493	0.529	0.552	0.668	0.640	0.562	0.598	0.617
13 PC	0.527	0.471	0.466	0.468	0.486	0.588	0.538	0.448	0.489	0.516	0.649	0.620	0.522	0.567	0.589

Acc	Akurasi
Prec	Presisi
Rec	Recall
F-meas	F-measure
Total	0,3*Akurasi+0,2*Presisi+0,3*Recall+0,2*F-measure

Lampiran 3 Pohon keputusan rekomendasi Pelatihan AMT

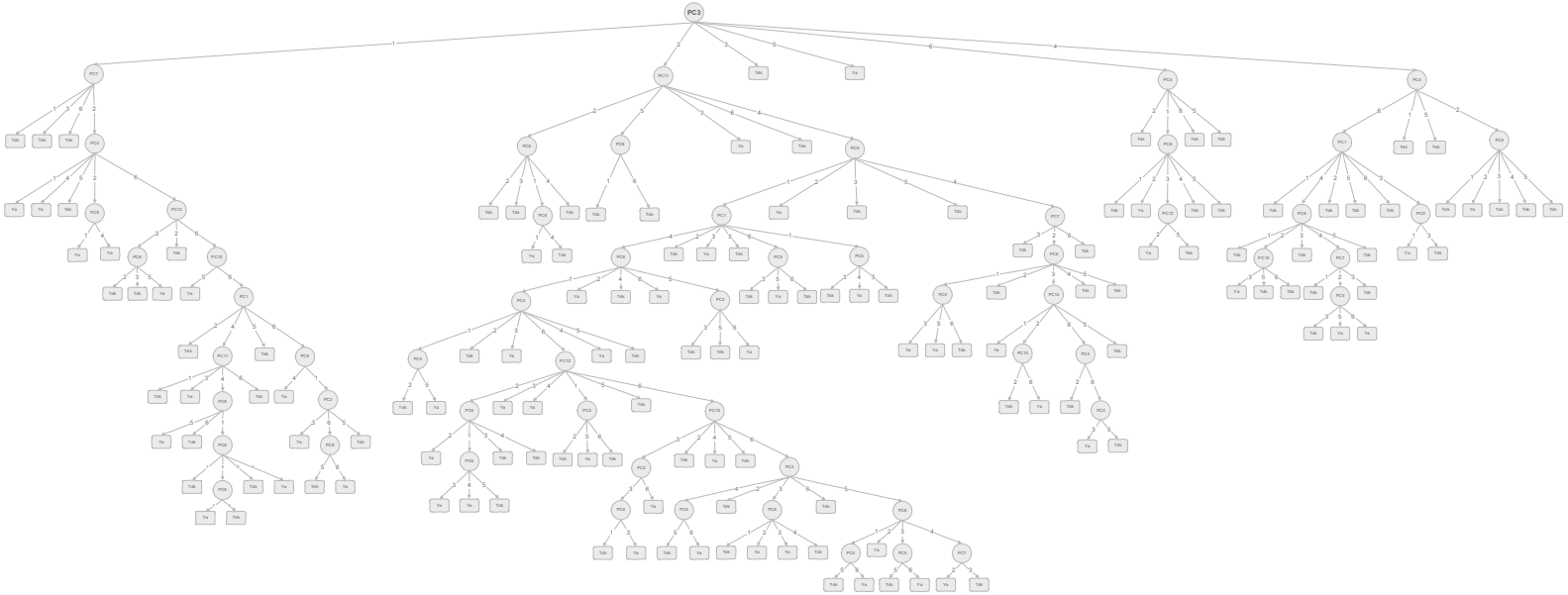


Lampiran 4 Hasil pengujian untuk pelatihan Effective Communication Skill

PC	PCA dan C4.5					PCA, Diskritisasi (Interval), dan C4.5					PCA, Diskritisasi (MDLP), dan C4.5				
	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total
2 PC	0.522	0.427	0.121	0.188	0.316	0.583	0.542	0.594	0.594	0.580	0.564	0.521	0.615	0.564	0.571
3 PC	0.502	0.437	**0.294**	**0.351**	0.396	0.727	0.664	0.821	0.734	0.744	0.710	0.651	0.791	0.714	0.723
4 PC	0.510	0.442	**0.259**	0.327	0.384	0.754	0.681	**0.874**	0.765	0.778	0.718	0.662	0.788	0.719	0.728
5 PC	0.505	0.437	0.274	0.336	0.388	0.754	0.686	0.856	0.762	0.773	0.713	0.658	0.776	0.713	0.721
6 PC	0.510	0.439	0.244	0.314	0.377	0.757	0.700	0.824	0.757	0.766	0.729	0.682	0.765	0.721	0.729
7 PC	0.532	0.481	**0.259**	**0.337**	0.401	0.792	0.742	0.838	0.787	0.795	0.740	0.696	0.768	0.730	0.737
8 PC	0.524	0.461	0.226	0.304	0.378	0.791	0.743	0.832	0.785	0.793	0.741	0.691	0.788	0.736	0.744
9 PC	**0.536**	**0.488**	0.235	0.317	0.392	0.810	0.761	0.853	0.804	0.812	0.750	0.698	0.803	0.747	0.755
10 PC	0.518	0.448	0.218	0.293	0.369	0.821	0.770	0.868	0.816	0.824	0.754	0.704	0.803	0.750	0.758
11 PC	0.530	0.467	0.168	0.247	0.352	0.830	**0.799**	0.841	0.819	0.825	0.776	0.731	0.809	0.768	0.775
12 PC	**0.536**	0.486	0.212	0.295	0.381	**0.842**	0.796	**0.882**	**0.837**	**0.844**	0.773	0.729	0.806	0.765	0.773
13 PC	**0.555**	**0.535**	0.224	0.315	**0.404**	**0.845**	**0.813**	0.859	**0.835**	0.841	**0.795**	**0.745**	**0.841**	**0.790**	**0.798**

PC	PCA dan C4.5					PCA, Diskritisasi (Interval), dan C4.5					PCA, Diskritisasi (MDLP), dan C4.5				
	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total
2 PC						0.564	0.527	0.491	0.508	*0.524*	0.536	0.495	0.632	0.556	0.561
3 PC						0.692	0.648	0.721	0.682	0.690	0.692	0.631	0.791	0.702	0.712
4 PC						0.725	0.671	0.785	0.724	0.732	0.719	0.659	**0.806**	0.725	0.734
5 PC						0.731	0.687	0.762	0.722	0.730	0.715	0.656	0.797	0.720	0.729
6 PC						0.762	0.710	0.815	0.759	0.767	0.740	0.687	0.794	0.737	0.745
7 PC						0.798	0.750	0.838	0.792	0.799	0.745	0.698	0.782	0.738	0.745
8 PC						0.799	0.751	0.841	0.793	0.801	0.746	0.700	0.782	0.739	0.746
9 PC						0.821	0.788	0.832	0.810	0.815	0.746	0.701	0.779	0.738	0.746
10 PC						0.849	0.824	0.853	0.838	0.843	0.748	0.703	0.779	0.739	0.747
11 PC						0.854	0.831	0.856	0.843	0.848	**0.768**	0.723	0.800	**0.760**	**0.767**
12 PC						0.858	0.835	0.862	0.848	0.853	0.767	0.721	0.800	0.759	0.766
13 PC						**0.861**	**0.838**	**0.865**	**0.851**	**0.855**	0.764	**0.724**	0.785	0.753	0.760

Lampiran 5 Pohon keputusan rekomendasi pelatihan Effective Comm. Skill



Lampiran 6 Hasil pengujian untuk pelatihan Human Skill Improvement

PC	PCA dan C4.5					PCA, Diskritisasi (Interval), dan C4.5					PCA, Diskritisasi (MDLP), dan C4.5				
	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total
2 PC	0.499	0.448	0.263	0.332	0.384	0.625	0.644	0.686	0.665	0.655	0.668	0.657	0.622	0.639	0.646
3 PC	0.489	0.416	0.202	0.272	0.345	0.636	0.666	0.659	0.662	0.654	0.680	0.666	0.647	0.656	0.662
4 PC	0.522	0.492	**0.342**	**0.403**	0.438	0.692	0.652	0.745	0.695	0.701	0.717	0.696	0.711	0.704	0.708
5 PC	0.520	0.487	0.317	0.384	0.425	0.698	0.669	0.714	0.691	0.696	0.730	0.712	0.720	0.716	0.721
6 PC	0.515	0.475	0.272	0.346	0.400	0.729	0.697	0.754	0.724	0.729	0.728	0.704	0.731	0.717	0.722
7 PC	0.522	0.489	0.261	0.340	0.401	0.750	0.718	0.776	0.746	0.750	0.739	0.714	0.748	0.731	0.735
8 PC	0.525	0.495	0.275	0.353	0.410	0.794	0.757	0.829	0.791	0.797	0.780	0.755	0.793	0.773	0.778
9 PC	0.525	0.494	0.238	0.321	0.392	0.820	0.784	0.854	0.818	0.823	0.803	0.771	0.829	0.799	0.804
10 PC	0.517	0.476	0.221	0.302	0.377	0.843	0.812	**0.868**	0.839	0.843	0.831	0.814	0.832	0.823	0.826
11 PC	0.528	0.500	0.207	0.293	0.379	**0.860**	**0.852**	0.852	**0.852**	**0.854**	0.776	0.731	0.809	0.768	0.775
12 PC	0.565	0.583	0.275	0.373	0.443	0.849	0.821	**0.871**	**0.845**	0.849	0.843	0.813	0.866	0.839	0.843
13 PC	**0.573**	**0.597**	0.294	0.394	**0.458**	**0.852**	**0.836**	0.854	**0.845**	0.848	**0.851**	**0.819**	**0.877**	**0.847**	**0.851**

Acc	Akurasi
Prec	Presisi
Rec	Recall
F-meas	F-measure
Total	0,3*Akurasi + 0,2*Presisi + 0,3*Recall + 0,2*F-measure

Lampiran 7 Hasil pengujian untuk pelatihan Personnel Effectiveness

PC	PCA dan C4.5					PCA, Diskritisasi (Interval), dan C4.5					PCA, Diskritisasi (MDLP), dan C4.5				
	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total
2 PC	**0.550**	0.565	**0.733**	**0.638**	**0.625**	0.603	0.608	**0.762**	0.677	0.667	**0.593**	0.593	**0.806**	**0.683**	**0.675**
3 PC	0.541	0.564	0.668	0.612	0.598	0.613	0.621	0.747	0.678	0.668	**0.593**	**0.598**	0.775	0.675	0.665
4 PC	0.516	0.545	0.637	0.587	0.572	0.627	0.646	0.698	0.671	0.661	**0.593**	**0.598**	0.775	0.675	0.665
5 PC	0.516	0.542	0.683	0.604	0.589	0.627	0.646	0.698	0.671	0.661	0.587	0.590	0.790	0.675	0.666
6 PC	0.528	0.556	0.630	0.591	0.577	0.625	0.644	0.686	0.665	0.655	0.578	0.590	0.741	0.657	0.645
7 PC	0.521	0.549	0.643	0.592	0.577	0.610	0.629	0.680	0.654	0.644	0.576	0.591	0.722	0.650	0.638
8 PC	0.533	0.558	0.658	0.604	0.590	0.618	0.637	0.686	0.661	0.651	0.576	0.591	0.722	0.650	0.638
9 PC	0.528	0.554	0.655	0.600	0.586	0.630	0.645	0.705	0.674	0.664	0.576	0.591	0.722	0.650	0.638
10 PC	0.524	0.549	0.683	0.609	0.594	**0.645**	**0.650**	0.745	**0.695**	**0.686**	0.576	0.591	0.722	0.650	0.638
11 PC	0.529	0.554	0.665	0.605	0.590	0.635	0.648	0.714	0.679	0.670	0.568	0.577	0.755	0.654	0.643
12 PC	0.536	0.560	0.665	0.608	0.594	0.622	0.633	0.717	0.672	0.663	0.563	0.576	0.727	0.643	0.631
13 PC	0.548	**0.571**	0.661	0.613	0.600	0.608	0.624	0.696	0.658	0.648	0.563	0.576	0.727	0.643	0.631

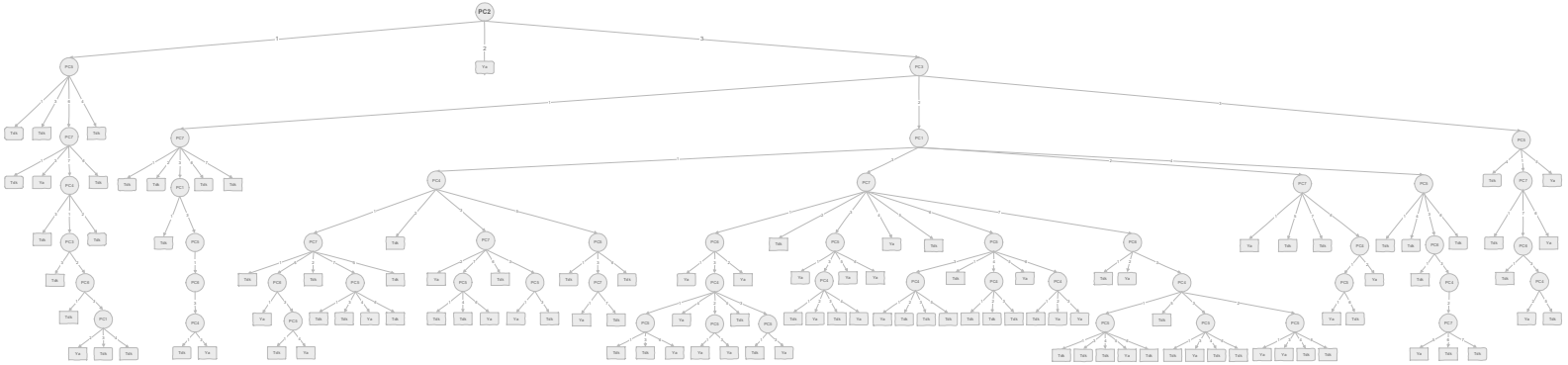
Acc	Akurasi
Prec	Presisi
Rec	Recall
F-meas	F-measure
Total	0,3*Akurasi + 0,2*Presisi + 0,3*Recall + 0,2*F-measure

Lampiran 8 Hasil pengujian untuk pelatihan Readiness to Change

PC	PCA dan C4.5					PCA, Diskritisasi (Interval), dan C4.5					PCA, Diskritisasi (MDLP), dan C4.5				
	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total
2 PC	**0.499**	**0.497**	**0.229**	**0.314**	**0.381**	0.713	0.663	0.863	0.750	0.755	0.726	0.674	0.873	0.761	0.767
3 PC	0.458	0.353	0.102	0.159	0.270	0.781	0.736	0.876	0.800	0.804	0.771	0.747	0.820	0.781	0.783
4 PC	0.477	0.426	0.134	0.204	0.310	0.809	0.765	0.890	0.823	0.827	0.814	0.786	0.861	0.822	0.824
5 PC	0.470	0.409	0.137	0.205	0.305	0.837	0.796	0.905	0.847	0.851	0.848	0.810	0.907	0.856	0.860
6 PC	0.473	0.400	0.112	0.175	0.290	0.884	0.867	0.907	0.887	0.888	0.868	0.833	**0.922**	0.875	0.879
7 PC	0.475	0.417	0.129	0.197	0.304	0.895	0.882	**0.912**	0.897	0.898	**0.904**	**0.895**	**0.915**	**0.905**	**0.905**
8 PC	0.486	0.450	0.132	0.204	0.316	0.898	0.890	0.907	0.899	0.899	0.899	0.888	0.912	**0.900**	0.901
9 PC	0.481	0.422	0.105	0.168	0.294	0.886	0.867	0.910	0.888	0.890	0.899	0.892	0.907	**0.900**	0.900
10 PC	0.474	0.400	0.107	0.169	0.288	0.893	0.880	0.910	0.894	0.896	0.898	0.888	0.910	0.899	0.900
11 PC	0.468	0.358	0.083	0.135	0.264	0.890	0.877	0.907	0.892	0.893	0.898	0.890	0.907	0.899	0.899
12 PC	0.462	0.387	0.134	0.199	0.296	0.898	0.890	0.907	0.899	0.899	0.895	0.888	0.905	0.896	0.897
13 PC	0.469	0.370	0.090	0.145	0.271	**0.903**	**0.902**	0.902	**0.902**	**0.902**	**0.900**	**0.896**	0.905	**0.900**	0.901

Acc	Akurasi
Prec	Presisi
Rec	Recall
F-meas	F-measure
Total	0,3*Akurasi + 0,2*Presisi + 0,3*Recall + 0,2*F-measure

Lampiran 9 Pohon keputusan rekomendasi pelatihan Readiness to Change



Lampiran 10 Hasil pengujian untuk pelatihan Team Building

PC	PCA dan C4.5					PCA, Diskritisasi (Interval), dan C4.5					PCA, Diskritisasi (MDLP), dan C4.5				
	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total	Acc (0,3)	Prec (0,2)	Rec (0,3)	F-meas (0,2)	Total
2 PC	0.488	0.468	0.502	0.485	0.488	0.423	0.426	**0.437**	0.432	0.430	0.499	0.500	0.169	0.252	0.351
3 PC	0.471	0.464	**0.679**	0.551	0.548	0.423	0.426	**0.437**	0.432	0.430	0.499	0.500	0.169	0.252	0.351
4 PC	0.486	0.473	0.638	0.543	0.540	0.447	0.439	0.372	0.403	0.414	0.499	0.500	0.169	0.252	0.351
5 PC	0.486	0.489	0.589	0.534	0.527	0.466	0.463	0.403	0.431	0.439	0.499	0.500	0.169	0.252	0.351
6 PC	0.505	0.487	0.611	0.542	0.541	0.508	0.510	**0.424**	0.463	0.474	**0.523**	0.533	**0.385**	**0.447**	**0.468**
7 PC	0.477	0.464	0.579	0.515	0.513	0.525	0.538	0.364	0.434	0.461	**0.523**	0.533	**0.385**	**0.447**	**0.468**
8 PC	0.447	0.447	0.643	0.527	0.522	0.514	0.521	0.368	0.431	0.455	0.508	**0.537**	0.126	0.204	0.338
9 PC	0.536	0.529	0.667	**0.590**	**0.585**	0.536	0.556	0.368	0.443	0.471	0.508	**0.537**	0.126	0.204	0.338
10 PC	0.534	0.528	0.658	0.586	0.580	0.542	0.563	0.385	0.458	0.482	0.508	**0.537**	0.126	0.204	0.338
11 PC	**0.540**	**0.535**	0.632	0.579	0.574	**0.555**	**0.584**	0.390	**0.468**	0.494	0.508	**0.537**	0.126	0.204	0.338
12 PC	0.458	0.451	0.597	0.514	0.509	**0.566**	**0.596**	0.416	**0.490**	**0.512**	0.508	**0.537**	0.126	0.204	0.338
13 PC	0.514	0.512	0.632	0.566	0.559	0.521	0.529	0.398	0.454	0.472	0.508	**0.537**	0.126	0.204	0.338

Acc	Akurasi
Prec	Presisi
Rec	Recall
F-meas	F-measure
Total	0,3*Akurasi + 0,2*Presisi + 0,3*Recall + 0,2*F-measure

Lampiran 11 Pohon keputusan rekomendasi pelatihan Team Building