

Jurnal Informatika

Pembuatan Aplikasi Saat Teduh "Makanan Rohani" di Windows 8 Metro
Erico Damawan Handoyo

Sistem Pendukung Keputusan untuk Seleksi Administrasi Penerimaan Pegawai
dengan Pohon Keputusan ID3
(Studi Kasus PT Jasamedika Saranotama)
Ady Andreas Prihatna, Hapnes Toba

Aplikasi Web untuk Pencarian Minat, Bakat, dan Kompetensi
dengan Metode Self-Directed Search untuk Alumni
Deddy Taufiq Rachmanto, Teddy Marcus Zakaria

Pengembangan Aplikasi Hypermedia Menggunakan Pendekatan
Relationship Management Methodology (RMM)
Sri Lestari, Falisahan

Pembuatan Aplikasi Datamining Facebook dan Twitter dengan *Naive Bayes Classifier*
Yosef Ganasaputra, Robby Tan

Sistem Pendukung Keputusan Penerimaan Pegawai
dengan Algoritma *Simple Additive Weighing* dan *Fuzzy Logic*
M.Rizki Fahrurrozi, Tjatur Kandangga Gautama

ISSN 0216-4280



9 770216 428004

UNIVERSITAS KRISTEN MARANATHA - BANDUNG

j. informatika

Vol. 9

No. 2

Hlm. 99 - 204

Bandung, Des 2013

ISSN 0216-4280

Jurnal Informatika

Volume 9 Nomor 2 Desember 2013

Pelindung:

Rektor Universitas Kristen Maranatha

Penasehat:

Pembantu Rektor Universitas Kristen Maranatha

Pembina:

Dekan Fakultas Teknologi Informasi Universitas Kristen Maranatha

Ketua Tim Redaksi:

Diana Trivena Yulianti, S.Kom., M.T.

Penyunting:

Wilfridus Bambang, S.T., M.Cs.

Robby Tan, S.T., M.Kom.

Niko Ibrahim, S.Kom., MIT

Diana Trivena Yulianti, S.Kom., M.T.

Penyunting Ahli:

Dr. Ir. Bambang SP. Abednego

Prof. Dr. Richardus Eko Indrajit

Perapih:

Dr. Andi Wahyu R. E., MSSE

Pelaksana Teknis:

Teddy Yusnandar

PENERBIT (PUBLISHER)

Maranatha University Press

ALAMAT PENYUNTING (EDITORIAL ADDRESS)

Sekretariat Jurnal Informatika

Jurusan Teknik Informatika

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri, MPH, No. 65 Bandung. 40164

Telp (022) 70753665

Fax (022) 2005915

Email: jurnal.informatika@itmaranatha.org

Homepage: <http://www.itmaranatha.org/jurnal/jurnal.informatika>

Jurnal Informatika terbit sejak 2005 merupakan jurnal ilmiah sebagai bentuk pengabdian dalam hal pengembangan bidang Teknik Informatika dan bidang terkait lainnya.

Jurnal Informatika diterbitkan oleh Jurusan Teknik Informatika Universitas Kristen Maranatha. **Redaksi** mengundang para professional dari dunia usaha, pendidikan dan peneliti untuk menulis mengenai perkembangan ilmu di bidang yang berkaitan dengan **Teknik Informatika**. **Jurnal Informatika** diterbitkan 2 (dua) kali dalam 1 tahun pada bulan **Juni** dan **Desember**. Harga berlangganan Rp 50.000.- / eksemplar.

Jurnal Informatika

Volume 9 Nomor 2 Desember 2013

DAFTAR ISI

Volume 9 Nomor 1

- | | | |
|---|---|---------|
| 1 | Penerapan Algoritma <i>Artificial Bee Colony</i> dalam Aplikasi Penjadwalan Pelajaran untuk Sekolah Menengah Pertama
Rakhmad Fajar Nugroho, Mewati Ayub | 1 - 17 |
| 2 | Aplikasi Pembelajaran Elektronik Tugas Online Menggunakan ASP.NET
Binsar Fofu M., Meliana Christianti J. | 19 - 27 |
| 3 | BossBook <i>Social Trading Business</i>
Yosep Agustinus Coatandi, Teddy Marcus Zakaria | 29 - 45 |
| 4 | Aplikasi Penghasil Video dari <i>Capture</i> Layar Komputer Menggunakan Bahasa Pemrograman Java
Tjatur Kandaga, Febrian Berthanio | 47 - 63 |
| 5 | Sistem HRD Perekrutan, Penggajian, dan Penjadwalan Menggunakan Algoritma Genetika pada Hotel Nirwana
Radiant Victor Imbar, Kevin Septiano | 65 - 80 |
| 6 | Aplikasi Jejaring Bisnis Berbasis Web
Wahyu Hidayat, Timotius Witono | 81 - 97 |

Volume 9 Nomor 2 (Akhir Volume)

- | | | |
|---|---|-----------|
| 7 | Pembuatan Aplikasi Saat Teduh “Makanan Rohani” di Windows 8 Metro
Erico Darmawan Handoyo | 99 - 116 |
| 8 | Sistem Pendukung Keputusan untuk Seleksi Administrasi Penerimaan Pegawai dengan Pohon Keputusan ID3 (Studi Kasus PT Jasamedika Saranatama)
Adry Andreas Prathama, Hapnes Toba | 117 - 131 |
| 9 | Aplikasi Web untuk Pencarian Minat Bakat, dan Kompetensi dengan Metode <i>Self-Directed Search</i> untuk Alumni | 133 - 154 |

10	Deddy Taufiq Rachmanto, Teddy Marcus Zakaria Pengembangan Aplikasi <i>Hypermedia</i> Menggunakan Pendekatan <i>Relationship Management Methodology (RMM)</i> Sri Lestari, Falahah	155 - 171
11	Pembuatan Aplikasi Datamining Facebook dan Twitter dengan <i>Naïve Bayes Classifier</i> Yosef Ganisaputra, Robby Tan	173 - 188
12	Sistem Pendukung Keputusan Penerimaan Pegawai dengan Algoritma <i>Simple Additive Weighting dan Fuzzy Logic</i> M.Rizki Fahrurrozi, Tjatur Kandaga Gautama	189 - 204

Ucapan Terima Kasih

Redaksi Jurnal Informatika mengucapkan terima kasih yang sebesar – besarnya kepada mitra bestari yang membantu terwujudnya penerbitan Jurnal Informatika Volume 9 Nomor 2 Desember 2013:

1. Danny Manongga (Fakultas Teknologi Informasi Universitas Kristen Satya Wacana)
2. Eko Sedyono (Fakultas Teknologi Informasi Universitas Kristen Satya Wacana)
3. Mewati Ayub (Fakultas Teknologi Informasi Universitas Kristen Maranatha)

Sistem Pendukung Keputusan untuk Seleksi Administrasi Penerimaan Pegawai dengan Pohon Keputusan ID3 (Studi Kasus PT Jasamedika Saranatama)

Adry Andreas Prathama¹⁾, Hapnes Toba²⁾

Jurusan S1 Teknik Informatika¹⁾, Program Studi D3 Teknik Informatika²⁾
Fakultas Teknologi Informasi, Universitas Kristen Maranatha
Jl. Prof. Drg. Suria Sumantri, MPH, No. 65 Bandung 40164
email:adryhappier@gmail.com¹⁾, hapnestoba@it.maranatha.edu²⁾

Abstract

PT. Jasamedika Saranatama, is a national-wide company which is engaged in the hospital software business. The human resources division of the company conduct the recruitment process for the entire company. Sometimes it is difficult to determine objectively which applicants meet the requirements based on the criteria expected by the company. Based on this problem, a web-based system has been developed which can supports the administrative selection (paper-work) in the recruitment process by using a data mining approach. The administrative rules are constructed based on historical data by using the ID3 algorithm. In the system, the rules are implemented in a MySQL database management system, and activated from a PHP web-based system. The system is tested by comparing some real data submissions between the activated system rules and the judgment from the human resources division. Statistic evaluation shows that the rules embedded in the system has a comparable performance as the human judgment.

Keywords: data mining, decision support system, ID3 algorithm

1. PENDAHULUAN

Sistem Pendukung Keputusan (*Decision Support System*) adalah sistem informasi berbasis komputer yang menyediakan dukungan informasi yang interaktif bagi manajer dan praktisi bisnis selama proses pengambilan keputusan [4]. Aplikasi DSS menggunakan CBIS (*Computer Based Information Systems*) yang fleksibel, interaktif, dan dapat diadaptasi, dikembangkan untuk mendukung solusi atau masalah manajemen spesifik yang tidak terstruktur.

Pada PT. Jasamedika Saranatama seleksi penerimaan pegawai dilaksanakan melalui beberapa tahapan, yaitu: seleksi administrasi, tes kemampuan, dan wawancara. Kendala yang sering ditemukan dalam proses penerimaan pegawai yaitu sulitnya menentukan pelamar mana yang memenuhi kriteria untuk menjadi pegawai dengan berdasarkan pada kriteria-kriteria yang ditetapkan

oleh perusahaan. Oleh sebab itu perusahaan membutuhkan suatu sistem pendukung keputusan (*Decision Support System*) yang dapat membantu bagian personalia untuk memutuskan pelamar mana yang akan diterima menjadi pegawai di PT. Jasamedika Saranatama. Dengan adanya sistem pendukung keputusan, dapat membantu perusahaan dalam penerimaan pegawai berdasarkan kriteria-kriteria yang sesuai dengan kebutuhan perusahaan, terutama untuk hal-hal yang bersifat obyektif, seperti penilaian administratif dan kelengkapan surat lamaran.

Penelitian akan memanfaatkan data-data historis pelamar yang telah diarsipkan pada bagian sumber daya manusia (HRD). Data historis dipakai untuk membentuk pohon keputusan saat seleksi administrasi dilakukan. Diharapkan bahwa pembelajaran akan dapat menghasilkan aturan-aturan yang bermanfaat dalam pengambilan keputusan.

Berdasarkan pada berbagai latar belakang sebagaimana dituliskan di atas, maka permasalahan pokok yang akan dibahas dalam penelitian ini adalah : 'bagaimana memanfaatkan data historis pelamar untuk pengambilan keputusan penerimaan pegawai dengan pendekatan *data mining*?'. Adapun tujuan dari perancangan sistem ini adalah membuat sistem pendukung keputusan untuk membantu perusahaan dalam menangani proses penerimaan pegawai baru sesuai kriteria yang dibutuhkan memanfaatkan data historis dengan pendekatan *data mining*.

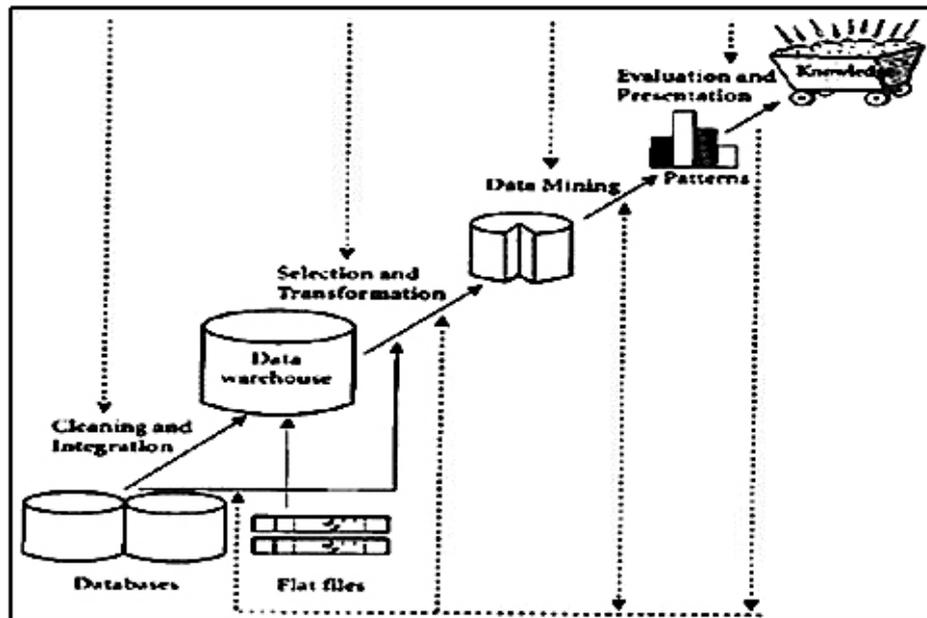
2. KAJIAN TEORI

Dalam bagian ini akan dipaparkan beberapa ulasan terkait data mining dan pembelajaran pohon keputusan yang digunakan dalam penelitian.

2.1 *Data Mining*

Data mining adalah suatu konsep yang digunakan untuk menemukan pengetahuan yang tersembunyi di dalam sekumpulan data. *Data mining* merupakan proses semi otomatis yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstrasi dan mengidentifikasi informasi pengetahuan potensial dan berguna yang tersimpan di dalam *database* besar [4]. *Data mining* adalah bagian dari proses KDD (*Knowledge Discovery in Databases*) yang terdiri dari beberapa tahapan seperti pemilihan data, pra-pengolahan, transformasi, *data mining*, dan evaluasi hasil [2].

Adapun tahapan-tahapan dalam proses *data mining* dapat dilihat pada Gambar 1 [1] :



Gambar 1. Tahap-tahap dalam *Datamining* [1]

2.2 *Decision Tree Learning* (Pembelajaran Pohon Keputusan)

Decision Tree Learning adalah salah satu metode belajar yang sangat populer karena mudah untuk diinterpretasi oleh manusia. Metode ini merupakan metode yang berusaha menemukan fungsi-fungsi pendekatan yang bernilai diskrit dan tahan terhadap data-data yang terdapat kesalahan (*noisy data*) serta mampu mempelajari ekspresi-ekspresi disjungsi (ekspresi OR) [3].

Decision tree merupakan salah satu metode klasifikasi yang menggunakan representasi struktur pohon (*tree*) di mana setiap node merepresentasikan atribut, cabangnya merepresentasikan nilai dari atribut, dan daun merepresentasikan kelas. *Node* yang paling atas dari *decision tree* disebut sebagai *root*. *Decision tree* merupakan metode klasifikasi yang paling populer digunakan. Selain karena pembangunannya relatif cepat, hasil dari model yang dibangun mudah untuk dipahami. Pada *decision tree* terdapat 3 jenis *node*, yaitu [3] :

1. *Root Node*, merupakan *node* paling atas, pada *node* ini tidak ada *input* dan bisa tidak mempunyai *output* atau mempunyai *output* lebih dari satu.
2. *Internal Node*, merupakan *node* percabangan, pada *node* ini hanya terdapat satu *input* dan mempunyai *output* minimal dua.
3. *Leaf node* atau *terminal node*, merupakan *node* akhir, pada *node* ini hanya terdapat satu *input* dan tidak mempunyai *output*.

2.3 Algoritma ID3

Iterative Dichotomizer version 3 (ID3) adalah algoritma *decision tree learning* (algoritma pembelajaran pohon keputusan) yang paling dasar. Algoritma ini melakukan pencarian secara menyeluruh (*greedy*) pada semua kemungkinan pohon keputusan [5]. Salah satu keunggulan algoritma ID3 adalah dapat diimplementasikan menggunakan fungsi *rekursif* sehingga mengurangi kompleksitas pembelajaran [3]. Algoritma ID3 membangun pohon keputusan (*decision tree*) secara *top-down* (dari atas ke bawah), dimulai dengan prosedur untuk menentukan “atribut yang pertama kali harus dievaluasi dan diletakkan sebagai *root*”. Strategi utama dalam ID3 adalah dengan mengevaluasi semua atribut dalam data dengan menggunakan suatu ukuran statistik (misalnya *information gain*) untuk mengukur efektifitas suatu atribut dalam mengklasifikasikan kumpulan sampel data.

Secara ringkas, cara kerja Algoritma ID3 dapat digambarkan sebagai berikut:

- Pemilihan atribut dengan menggunakan *information gain*.
- Pilih atribut dimana nilai *information gain*-nya terbesar.
- Buat simpul yang berisi atribut tersebut.
- Proses perhitungan *information gain* akan terus dilaksanakan sampai semua data telah termasuk dalam kelas yang sama. Atribut yang telah dipilih tidak diikuti lagi dalam perhitungan nilai *information gain*.

2.3.1 Entropy

Untuk menghitung *information gain*, terlebih dahulu harus dipahami suatu perhitungan yang disebut *entropy*. Dalam bidang teori informasi, *entropy* seringkali digunakan sebagai suatu parameter untuk mengukur heterogenitas (keberagaman) dari suatu kumpulan sampel data. Jika kumpulan sampel data semakin heterogen, maka nilai *entropy*-nya semakin besar. Secara matematis, *entropy* dirumuskan pada Formula nomor 1 berikut [3] :

$$\text{Entropy}(S) = - p+ \log_2 p+ - p- \log_2 p- \quad \dots (1)$$

dimana : $p+$: jumlah sample (+)
 $p-$: jumlah sample (-)

2.3.2 Information Gain

Setelah mendapatkan nilai *entropy* untuk suatu kumpulan *sampel* data, maka efektifitas suatu atribut dapat diukur untuk mengklasifikasikan data. Ukuran efektifitas ini disebut sebagai *information gain*. Secara matematis, *information gain* dari suatu atribut A, dirumuskan pada formula nomor 2 berikut [3]:

$$\text{Gain} (S, A) = \text{Entropy}(S) - \sum_{V \in \text{Value}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \quad \dots (2)$$

dimana :

A : atribut
V : menyatakan suatu nilai yang mungkin untuk atribut A
Values(A) : himpunan nilai-nilai yang mungkin untuk atribut A
|S_v| : jumlah sampel untuk nilai v
|S| : jumlah seluruh data
Entropy (S_v) : *entropy* untuk sampel-sampel yang memiliki nilai v

3. METODOLOGI, PERANCANGAN DAN IMPLEMENTASI SISTEM

Dalam bagian ini disampaikan metodologi dan perancangan sistem pengambilan keputusan yang dilakukan dalam penelitian, mulai dari proses penggalian data sampai dengan perancangan implementasi aturan dalam sistem pengambilan keputusan.

3.1 Proses Data Mining

Tahapan-tahapan yang diperlukan untuk *data mining* antara lain :

1. *Data cleaning*, membersihkan data dari *noise* dan data yang tidak konsiten. Proses pembersihan mencakup memperbaiki maupun membuang data yang kosong atau pun yang tidak diperlukan, contoh pada atribut: *name*, *status*, *email* maupun *address*. Juga dilakukan proses *enrichment*, yaitu proses "memperkaya" data yang sudah ada

dengan data atau informasi lain yang relevan, misalnya dengan menambahkan atribut *point* untuk menilai kelengkapan *Curriculum Vitae* (VC).

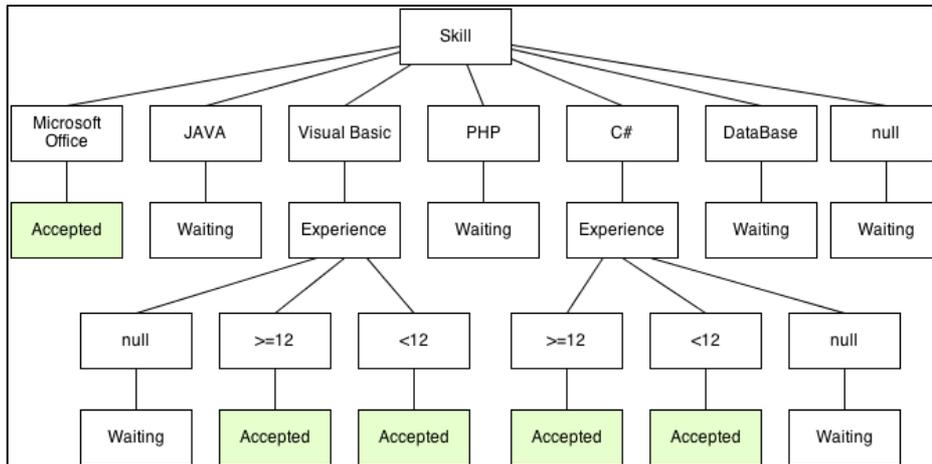
2. *Data transformation*, mentransformasikan data *summary* ataupun operasi agregasi. Pada tahap transformasi ini format data diubah ke dalam bentuk yang sesuai, contohnya data yang berupa angka numerik seperti pada atribut Indeks Prestasi Kumulatif (IPK) perlu dibagi-bagi menjadi beberapa interval. Proses ini sering disebut *binning* atau diskretisasi.
3. *Data mining*, merupakan proses yang esensial dimana metode digunakan untuk mengekstrak pola data yang tersembunyi. Dalam melakukan proses *data mining* ini dibantu dengan salah satu perangkat lunak populer, yaitu Weka. Pada perangkat ini data diproses menggunakan teknik *classification* berupa metode pohon keputusan (*Decision Tree*) dengan algoritma ID3. Algoritma ini digunakan karena setiap atribut diproses dan diuji kemungkinan yang ada, tanpa adanya pembuangan aturan. Hasil dari proses data tersebut berupa aturan-aturan berbentuk pohon keputusan, kemudian dirubah menjadi kumpulan *record* dan disimpan dalam *database* untuk menjadi suatu acuan.
4. *Evaluation and Presentation*, mengidentifikasi pola sehingga merepresentasikan pengetahuan berdasarkan nilai-nilai yang menarik. Pola informasi yang dihasilkan dari proses *data mining* perlu ditampilkan dalam bentuk yang mudah dimengerti oleh pihak yang berkepentingan. Tahap ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan bertentangan dengan fakta atau hipotesa yang ada sebelumnya.

3.2 Model Keputusan

Untuk membuat suatu model keputusan dari data historis pelamar dibuatlah beberapa skenario transformasi data. Untuk data pelatihan digunakan 100 data lamaran pekerjaan untuk beberapa posisi: analis sistem, *programmer*, dan teknisi. Dengan menggunakan bantuan perangkat lunak Weka beberapa skenario tersebut dicoba untuk menghasilkan aturan-aturan yang sesuai dengan fakta atau hipotesa yang dipakai sebelumnya untuk penerimaan pegawai, sebagai berikut:

1. *Non-Leveling (Single-Tree)*
Skenario *non-leveling*, dengan *single-tree* dari data history CV pelamar diproses dan hasil akhir hanya melihat atribut *result*, apakah CV tersebut diterima, atau menunggu. Aturan pohon keputusan yang dihasilkan belum memenuhi harapan, karena belum terlihat penentuan *job position* sesuai

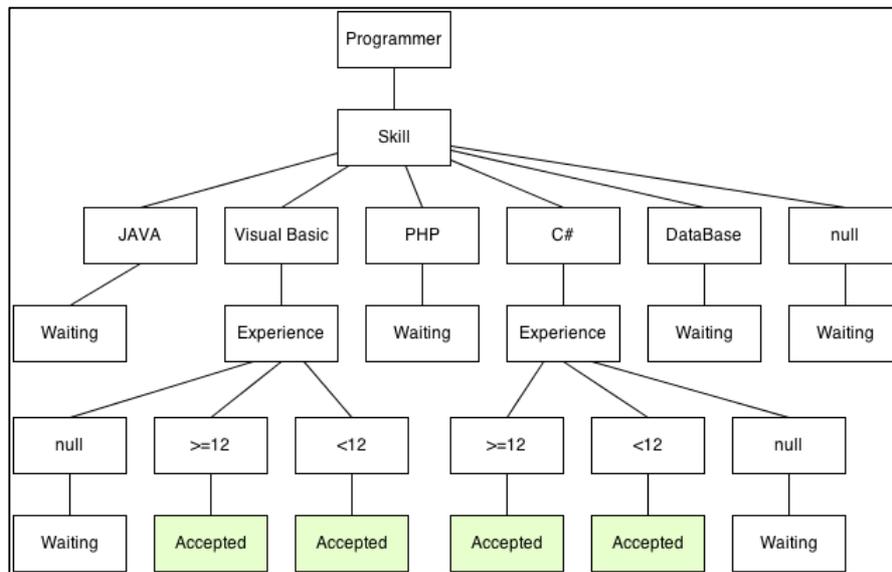
dengan yang dilamar. Hasil *rules* dari Weka yang menjadi *root node* adalah *skill* karena atribut yang paling berpengaruh atau memiliki nilai *information gain* yang besar. Gambar 2 merupakan contoh bentuk pohon keputusan *non-leveling* implementasi dari hasil Weka.



Gambar 2. Pohon Keputusan Non-Leveling

2. *Leveling (Sub-Tree)*

Skenario *leveling*, dengan *sub-tree* per posisi lamaran dari data historis CV pelamar yang diproses. Dengan menambahkan informasi tambahan yaitu atribut *point* untuk menilai kelengkapan CV dan pendekatan dengan persyaratan posisi yang dilamar. Hasil akhir tetap berupa keputusan, apakah CV tersebut diterima, atau menunggu, namun lebih spesifik untuk posisi pekerjaan tertentu. Pada skenario pohon keputusan ini menghasilkan aturan yang lebih detail untuk setiap posisi lamaran, dan tidak dipengaruhi oleh atribut diluar posisi lamaran. Maka pada sistem ini digunakan skenario *leveling* dalam pembuatan *rules*. Dengan metode *leveling* ini, yang menjadi *root node* tetap adalah *skill* karena atribut yang paling berpengaruh atau memiliki nilai *information gain* yang besar. Gambar 3 memberikan contoh bentuk pohon keputusan *leveling* untuk posisi programmer implementasi dari hasil Weka.



Gambar 3. Pohon Keputusan *Leveling* (untuk posisi *programmer*)

3.3 Algoritma Pencocokan

Dengan menggunakan ekstraksi aturan dari pohon keputusan membuat pengambilan keputusan lebih mudah dipahami dibandingkan dengan menggunakan *tree* secara langsung. Sebuah aturan diciptakan untuk setiap jalur dari akar ke daun dan setiap pasang *attribute-value* dalam suatu jalur membentuk suatu *conjunction* (aturan 'dan'), dengan daun pada pohon keputusan sebagai suatu kelas prediksi yang dituju. Contoh ekstraksi aturan untuk posisi *programmer* dengan pohon keputusan *Leveling* (gambar 3), adalah sebagai berikut:

```

if position = programmer and skill = java then result = waiting
if position = programmer and skill = visual basic
and experience = null then result = waiting
if position = programmer and skill = visual basic
and experience = ">=12" then result = accepted
if position = programmer and skill = visual basic
and experience = "<12" then result = accepted
  
```

Adapun *pseudocode* dari algoritma pencocokan adalah sebagai berikut:

```
Program PencocokanRules {menentukan kecocokan antara dua nilai yang dimasukkan dan hasilnya berupa status boolean}
Kamus
    hasil : boolean {hasil merupakan variable untuk menampung nilai keluaran}
    atributCV, aturan : array {adalah variable untuk menampung nilai masukan}
Proses
    input(atributCV, aturan) {membaca nilai dari piranti masukan}
    foreach (atributCV, aturan)
        if (atributCV == aturan)
            then hasil = true
            else hasil = false
        end for
    output (hasil)
```

4. PENGEMBANGAN PERANGKAT LUNAK

4.1 Implementasi *Rules* dalam *Database*

Pada bagian ini akan dijelaskan implementasi *rules* dalam *database* Sistem Pendukung Keputusan Seleksi Administrasi Penerimaan Pegawai. Hasil *rules leveling* yang telah terbentuk dengan bantuan Weka untuk posisi lamaran *programmer* membentuk pohon keputusan yang terdapat pada Gambar 3. *Rules* yang dihasilkan merupakan representasi dari cabang-cabang pohon keputusan tersebut, satu *rule* diciptakan untuk setiap jalur dari akar ke daun. Dari jalur-jalur cabang aturan pohon keputusan kemudian dirubah menjadi kumpulan *record* aturan dan disimpan dalam *table requirement* pada *database* untuk menjadi suatu acuan. Contoh implementasi *rules* dalam *table requirement* yang dapat dilihat pada Gambar 4.

jobPosition	gender	degree	major	rangeGPA	skill	proficiency	experience	traveling	relocated	point	result
PROGRAMMER	Male	S1	IT	GPA > 3.00	C#	Intermediate	experience >= 12	Y	N	4	Accepted
PROGRAMMER	Male	S1	IT	2.75 >= GPA < 3.00	C#	Beginner	experience < 12	Y	N	4	Accepted
PROGRAMMER	Male	S1	IT	GPA > 3.00	VISUAL BASIC	Intermediate	experience >= 12	Y	N	4	Accepted
PROGRAMMER	Male	D3	IT	GPA > 3.00	C#	Intermediate	experience < 12	Y	N	4	Accepted
PROGRAMMER	Female	S1	IT	GPA > 3.00	VISUAL BASIC	Intermediate	experience < 12	Y	N	4	Accepted
PROGRAMMER	Female	S1	IT	GPA > 3.00	VISUAL BASIC	Advance	experience >= 12	Y	N	4	Accepted
PROGRAMMER	Female	S1	IT	GPA > 3.00	VISUAL BASIC	Advance	experience < 12	Y	N	4	Accepted
PROGRAMMER	Male	S1	IT	2.75 >= GPA < 3.00	C#	Intermediate	experience < 12	Y	N	4	Accepted
PROGRAMMER	Male	S1	IT	GPA > 3.00	VISUAL BASIC	Advance	experience >= 12	Y	N	4	Accepted
PROGRAMMER	Male	D3	IT	2.75 >= GPA < 3.00	VISUAL BASIC	Beginner	experience >= 12	Y	N	4	Accepted
PROGRAMMER	Male	S1	IT	2.75 >= GPA < 3.00	VISUAL BASIC	Beginner	experience < 12	Y	N	4	Accepted
PROGRAMMER	Male	D3	IT	2.75 >= GPA < 3.00	C#	Intermediate	experience >= 12	Y	N	4	Accepted
PROGRAMMER	Male	S1	IT	2.75 >= GPA < 3.00	VISUAL BASIC	Intermediate	experience < 12	Y	N	4	Accepted
PROGRAMMER	Male	D3	IT	2.75 >= GPA < 3.00	C#	Beginner	experience >= 12	Y	N	4	Accepted

Gambar 4. Contoh Implementasi Rules pada Table Requirement

4.2 Implementasi Pemanggilan Rules

Pada bagian ini akan dijelaskan *script* proses pemanggilan *rules*. Program 1 memberikan prosedur untuk mencocokkan tiap atribut antara data CV pelamar baru dengan data dari *tabel requirement*, seperti dapat dilihat pada contoh potongan program di baris 3 – 12. Pencocokan aturan akan menghasilkan sebuah keputusan untuk menentukan apakah sebuah dokumen lamaran kerja akan diproses lebih lanjut atau tidak.

```

1  foreach($data->result() as $cv){
2      $sum=0;
3          $match['job'] = $cv->jobPosition;
4          $match['gender'] = $cv->gender;
5          $match['degree'] = $cv->degree;
6          $match['major'] = $cv->major;
7          $match['range'] = $cv->rangeGPA;
8          $match['skill'] = $cv->skill;
9          $match['proficiency'] = $cv->proficiency;
10         $match['experience'] = $cv->experience;
11         $match['traveling'] = $cv->traveling;
12         $match['relocated'] = $cv->relocated;
13         $matching = $this->cvm->matching($match);
14         $sum = count($matching->result());
15         if($sum > 0) {
16             $result = $matching->row()->result;
17             $point += $matching->row()->point;
18             if($result == 'Accepted') {
19                 $count += 1; $point += 10;
20             }
16         }

```

```

21             }else $count+=0;
22             }
23 }
```

Program 1. Kode Program untuk Pencocokan Rules

5. TESTING DAN EVALUASI SISTEM

Pengujian sistem merupakan hal penting yang bertujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang diuji. Pengujian sistem ini menggunakan 2 metode yaitu metode *white box* dan *black box*. Di samping itu dilakukan juga pengujian statistik untuk mengetahui signifikansi data pengujian antara sistem dan penilaian manual dari divisi HRD.

5.1.1 White Box

Merupakan metode perancangan *test case* yang menggunakan struktur kontrol dari perancangan prosedural untuk mendapatkan *test case*. Pengujian *white box* ini menggunakan *library unit_test* yang ada pada *Framework CodeIgniter* dan dilakukan pada modul *job vacancy*. Contoh pengujian pada model '*job vacancy*' untuk *function 'apply job'* dapat dilihat pada Gambar 5.

Test Name	Cek Data Job Vacancy
Test Datatype	Array
Expected Datatype	Array
Result	Passed
File Name	C:\xampp\htdocs\Cv\application\controllers\testing.php
Line Number	32
Notes	

Gambar 5. Pengujian white box untuk *function apply job*

5.1.2 Black Box

Pengujian dengan *black box* dilakukan untuk mencari kesalahan atau *error* pada aplikasi dari sisi masukan yang didapatkan dari pengguna untuk sebuah fungsionalitas tertentu. Contoh pengujian pada proses pembuatan data *personal* dapat dilihat pada Tabel 1. Data personal ini digunakan sebagai identifikasi utama dalam pengelolaan data pelamar dan proses seleksi.

Tabel 1. Pengujian *black box* untuk pembuatan data personal

Nama Masukan	Tipe Field	Tipe Input	Keterangan
Full Name	TextBox	Angka	Nama harus di isi apabila tidak di isi akan muncul pesan “ <i>This field is required.</i> ”
		Karakter	
		Karakter khusus	
		Kosong	X
Place of Birth	TextBox	Angka	Tempat kelahiran harus di isi apabila tidak di isi akan muncul pesan “ <i>This field is required.</i> ”
		Karakter	
		Karakter khusus	
		Kosong	X
Date of Birth	Date	Angka	Tanggal kelahiran harus di isi apabila tidak di isi akan muncul pesan “ <i>please fill out this field</i> ”
		Karakter	
		Karakter khusus	
		Kosong	X
Gender	Combobox	Angka	Jenis kelamin harus diisi terdiri dari ‘ <i>Male</i> ’ dan ‘ <i>Female</i> ’
		Karakter	
		Karakter khusus	
		Kosong	X
Marital Status	Combobox	Angka	<i>Marital status</i> harus diisi terdiri dari ‘ <i>Single</i> ’, ‘ <i>Married</i> ’ dan ‘ <i>Divorce</i> ’
		Karakter	
		Karakter khusus	
		Kosong	X
Email Address	Textbox	Angka	<i>Email address</i> harus di isi apabila tidak di isi akan muncul pesan “ <i>This field is required.</i> ”
		Karakter	
		Karakter	

Nama Masukan	Type Field	Type Input	Keterangan	
		khusus		
		Kosong	X	Fieldemail address tidak boleh kosong

5.1.3 Pengujian Statistik

Untuk evaluasi statistik digunakan metode uji kesamaan rerata *t-Test*. *T-Test* digunakan untuk menguji rerata beberapa populasi. Terdapat beberapa pilihan untuk melakukan uji-t, sesuai data yang diperoleh. Gambar 6 memberikan hasil pengujian *t-Test* antara skenario *rules leveling* dan pengujian oleh staf HRD dengan menggunakan angka signifikansi sebesar 0,05 untuk 30 data lamaran. Hasil pengujian menunjukkan nilai absolut *t-Stat* (yaitu: -1,558174245), dan masih berada dalam rentang atau lebih kecil dari nilai *t-Critical* (yaitu: 1,671552762).

t-Test: Two-Sample Assuming Equal Variances

	<i>Rules Leveling</i>	<i>Modified HRD</i>
Mean	0,366666667	0,566666667
Variance	0,240229885	0,254022989
Observations	30	30
Pooled Variance	0,247126437	
Hypothesized Mean Difference	0	
Df	58	
t Stat	-1,558174245	
P(T<=t) one-tail	0,062316132	
t Critical one-tail	1,671552762	
P(T<=t) two-tail	0,124632264	
t Critical two-tail	2,001717484	

Gambar 6. Daftar Hasil *T-Test*

Dengan hasil tersebut dapat disimpulkan bahwa aturan yang dibuat menggunakan pendekatan *data mining*, “tidak signifikan berbeda” dan cukup mewakili apa yang diharapkan bagian personalia. Selain itu dapat membantu bagian personalia dalam tahap awal untuk menentukan calon pegawai yang sesuai dengan posisi pekerjaan. Sedangkan untuk beberapa kasus khusus *rules* perlu ditambahkan secara manual oleh bagian personalia di *table requirement* dalam sistem ini.

6. KESIMPULAN DAN SARAN

Kesimpulan yang dapat diambil berdasarkan hasil analisis dan pembahasan yang sedang berjalan dan rancangan sistem adalah sebagai berikut:

1. Salah satu hal penting yang sangat menentukan performa sistem adalah pada saat pengolahan data, yaitu pada proses pembersihan data (*cleaning*) dan pada saat pembuatan *rules* (bandingkan dengan tahap-tahap data mining pada Gambar 1).
2. Dalam membuat sistem dengan pendekatan *data mining*, *rules* dibuat dari kumpulan data historis pelamar diproses dengan bantuan perangkat lunak *data mining* Weka dengan *algoritma decision tree learning* ID3. *Rules* hasil proses tersebut disimpan ke dalam *database*, untuk dijadikan acuan dalam proses seleksi administrasi penerimaan pegawai selanjutnya.
3. Pengujian sistem seleksi penerimaan pegawai untuk memastikan bahwa sistem telah sesuai dengan yang diharapkan berhasil dilakukan dengan menggunakan metode *whitebox* dan *blackbox* serta pengujian algoritma. Dari hasil pengujian dengan metode *whitebox* dan *blackbox* dapat disimpulkan bahwa sistem ini secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan. Sedangkan hasil dari pengujian algoritma, *rules* yang diimplementasikan dalam sistem ini “tidak signifikan berbeda” dengan harapan dari bagian personalia (lihat hasil pengujian pada bagian 5.1.3).
4. Sistem pendukung keputusan yang dikembangkan untuk proses seleksi administrasi penerimaan pegawai baru telah dapat membantu bagian personalia untuk memberikan gambaran dalam melakukan penyeleksian dengan membandingkan CV pelamar dengan *rules leveling* yang telah dibuat sebelumnya. *Rules leveling* dipilih karena dari hasil pengujian memiliki performa yang setara dengan keputusan bagian personalia (lihat hasil pengujian pada bagian 5.1.3).

Berdasarkan evaluasi yang telah dilakukan untuk sistem yang telah dibangun, terdapat beberapa hal yang perlu dilakukan untuk pengembangan sistem ini, antara lain:

1. Dalam sistem ini perlu adanya pengembangan pada penyajian informasi lowongan kerja yang ditampilkan sesuai dengan kriteria-kriteria CV pelamar, sehingga dapat membantu pengguna mendapatkan referensi informasi lamaran yang dibutuhkan.
2. Proses sistem pendukung keputusannya dapat diperluas dengan menggunakan teknik-teknik *data mining* ataupun algoritma lainnya, contohnya menggunakan algoritma *naive bayesian classification* dengan pembentukan model.

DAFTAR PUSTAKA

- [1] Han, J., and Kamber, M. (2001). *Data Mining : Concepts and Techniques*. Morgan Kaufmann.
- [2] Maimon, O., and Last, M. (2000). *Knowledge Discovery and Data Mining*. The Info-Fuzzy Network (IFN) Methodology. Dordrecht: Kluwer Academic.
- [3] Suyanto ST., M. (2007). *Artificial Intelligence*. Bandung: Penerbit Informatika.
- [4] Turban, E. (2001). *Decision Support Systems and Expert Systems and Intelligent Systems, 6th Ed*. New Jersey: Prentice Hall Internasional, Inc.
- [5] Witten H.I., Frank, E., and Hall, M.A. (2011). *Data Mining Practical Machine Learning Tools and Techniques, 3rd Ed*. Morgan Kaufmann Publishers