

**SISTEM PAKAR PENENTUAN PASAL DAN LAMA HUKUMAN
UNTUK PERMASALAHAN TINDAK PIDANA PENGANIAYAAN**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Nama Achmad Dwi Ardian

NIM: 105060800111016



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016**

PENGESAHAN

SISTEM PAKAR PENENTUAN PASAL DAN LAMA HUKUMAN UNTUK
PERMASALAHAN TINDAK PIDANA PENGANIAYAAN

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Achmad Dwi Ardian
NIM: 105060800111016

Skripsi ini telah diuji dan dinyatakan lulus pada
21 Juli 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Ir.Sutrisno, M.T
NIP: 195703251987011001

M. Tanzil Furgon, S.Kom, M.Comp.Sc
NIP: 19820930 200801 1 004

Mengetahui
Ketua Jurusan Teknik Informatika

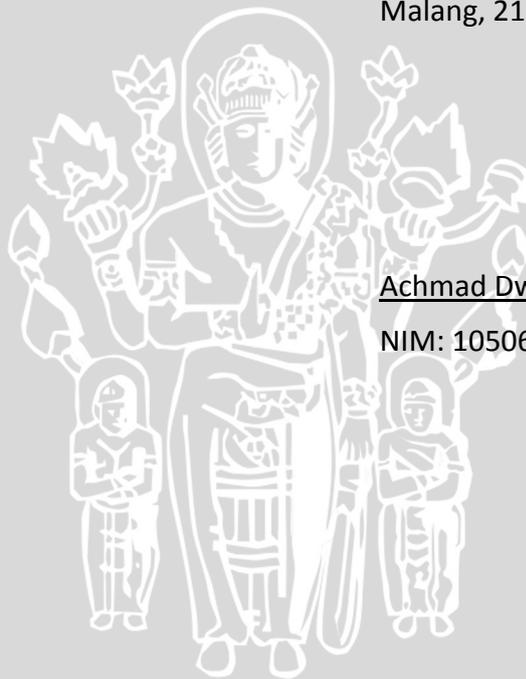
Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 21 Juli 2016



Achmad Dwi Ardian

NIM: 105060800111016

KATA PENGANTAR

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT yang telah melimpahkan segala rahmat serta karunia-Nya, sehingga penulis dapat menyelesaikan penulisan skripsi dengan judul “SISTEM PAKAR PENENTUAN PASAL DAN LAMA HUKUMAN UNTUK PERMASALAHAN TINDAK PIDANA PENGANIAYAAN”, yang diajukan untuk menempuh Ujian Akhir Sarjana Program Strata Satu Jurusan Informatika.

Selesainya penulisan skripsi ini tidak terlepas dari peran serta berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, selaku Dekan Fakultas Ilmu Komputer.
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika di Fakultas Ilmu Komputer
3. Bapak Agus Wahyu Widodo, ST, M.Cs selaku Ketua Program Studi Teknik Informatika di Fakultas Ilmu Komputer.
4. Bapak Ir.Sutrisno, M.T dan bapak M. Tanzil Furqon, S.Kom, M.Comp.Sc, selaku dosen pembimbing penulis yang dengan sabar memberikan kritik, saran, serta arahan yang baik dalam proses pengerjaan skripsi ini.
5. Bapak dan Ibu Dosen yang mengajar beserta *staff* akademik di Program Studi Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Bapak Bondan Supodo, S.H selaku pakar (hakim) yang telah meluangkan waktunya untuk membantu proses pengumpulan data skripsi.
7. Ayahanda Abu Amin, Ibunda Nur Aini, Kakak M.Denny A.S. dan seluruh keluarga besar atas nasehat, kasih sayang, perhatian, dan kesabarannya dalam mendidik penulis, serta tiada hentinya selalu memberikan dukungan dan doa demi terselesaikannya skripsi ini.
8. Sarah Wahyu Susanti yang selalu memberikan semangat, membantu serta menemani penulis dalam proses pengerjaan skripsi ini.
9. Teman-teman seperjuangan, Arie Desprianto, Agung Dwi Wicaksono, Arif Rahman, M Fahmi Yusuf, Yulis Raga, M Rifqi Aditya, Rifqi Kurniawan, Maftuhriza Afrul Yumida, Rizki Bangkit yang selalu berbagi semangatnya dan saling berbagi ilmu untuk menyelesaikan skripsi.
10. Teman–teman seluruh Angkatan 2010 Teknik Informatika, terima kasih

atas segala bantuannya selama menempuh studi di Teknik Informatika Universitas Brawijaya.

11. Teman-teman Lembaga Semi Otonom Optiik, yang selalu memberikan tempat, membantu, berbagi ilmu, dan bertukar semangat dengan penulis selama menyelesaikan tugas akhir ini.
12. Teman-teman Sambung Bayar , Zubad Adam, Febrian Arifianto, Trio kusuma putra, M. Zakiya Nafik, M. Bachtiar Affandi, Dani Al Faruq, Syaiful Kurniawan, Angga Kurniawan, Rachman Wicaksono, Tegar Ardianto, M. adhimun T yang selalu berbagi semangatnya dan selalu memberi dukungan untuk menyelesaikan skripsi.
13. Semua pihak lain yang telah membantu terselesaikannya skripsi ini yang tidak bisa penulis sebutkan satu-persatu.

Semoga Allah SWT melimpahkan rahmat dan karunia-Nya kepada semua pihak yang telah membantu penulis dalam menyelesaikan penulisan skripsi ini. Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak terdapat kekurangan. Untuk itu penulis menyampaikan permohonan maaf sebelumnya, serta sangat diharapkan kritik dan saran yang bersifat membangun.

Malang, 21 juni 2016

Penulis

Achmad Dwi Ardian
adwardian@gmail.com

ABSTRAK

Indonesia merupakan sebuah negara hukum, sehingga memiliki peraturan-peraturan hukum yang mengatur, salah satunya adalah Kitab Undang-undang Hukum Pidana (KUHP) dan Kitab Undang-Undang Perdata (KUH PERDATA). Ketika suatu norma hukum dibentuk atau ditetapkan maka pada saat itu juga setiap orang di Indonesia diharuskan dan dianggap tahu hukum. Bidang hukum juga merupakan bidang yang jarang disentuh oleh teknologi informasi. Dikarenakan masih banyaknya masyarakat yang buta akan undang-undang diperlukan adanya sebuah program komputer sederhana untuk membantu memahami dan memilah-milah pasal-pasal yang terlibat dalam suatu kasus hukum. Pada penelitian ini masalah tersebut diselesaikan dengan membuat system pakar yang dapat menentukan pasal sesuai kasus yang dilanggar beserta berapa lama hukuman yang didapatkan khusus bab penganiayaan saja, sehingga dapat membantu masyarakat agar lebih tahu tentang hukum khususnya bab penganiayaan. minimal masyarakat awam tahu hukuman maksimal dari pasal yang dilanggar, sehingga masyarakat dapat sedikit lebih aktif dalam mengkritisi sekaligus mengawasi kasus yang ada ketika sang hakim sedikit melenceng dari ketetapan KUHP yang ada. Sistem ini menggunakan metode Iterative Dichotomiser *Tree* (ID3). Sehingga diharapkan sistem ini dapat membantu memberikan solusi untuk mengatasi masalah yang sering dihadapi oleh masyarakat.

Pengujian dilakukan dengan cara membandingkan hasil keluaran dari sistem dengan data aktual yang diperoleh dari pakar. Hasil pengujian didapatkan akurasi terbaik sebesar 92% dari beberapa kali pengujian, baik 25 data latih ataupun pada 40 data latih, menunjukkan bahwa sistem bekerja dengan baik sesuai dengan hasil aktual pakar.

Kata kunci: Sistem Pakar, Hukum, Pasal, Penganiayaan, Iterative Dichotomiser *Tree*

ABSTRACT

Indonesia is a state of law, and so have the *Rules* governing law, one of them is a Kitab Undang-undang Hukum Pidana (KUHP) and Kitab Undang-Undang Perdata (KUH PERDATA). When a legal norms established or set then at that moment as well as any people in Indonesia are required to know the law and are considered. The law is also the field that are rarely touched by information technologies. Because there are still many people who are blind to the law , it is necessary to have a simple computer program to help understand and sort out the articles involved in a law case. In this research the problem is solved by making experts system who can determine the appropriate case that article are violated along with how long the punishment it brings special chapters of persecution only, so that it can help more people to know about the law in particular chapters of persecution. at least lay community know the maximum punishment from article being violated. so that the community can be a little more active in critiquing while overseeing the cases when the judge is slightly deviated from the provision of the CRIMINAL CODE. This system uses the method of Iterative Dichotomiser *Tree* (ID3). So expect this system can help provide solutions to solve the problem often encountered by people.

Testing is done by comparing the output of the system with the actual data obtained from experts. The testing results obtained the best accuracy of 92% of the test several times, either 25 or 40 trainer data training, It shows that the system works well and suits with the experts actual Renault

Keywords: expert system , law , article , persecution , iterative dichotomiser *tree*.

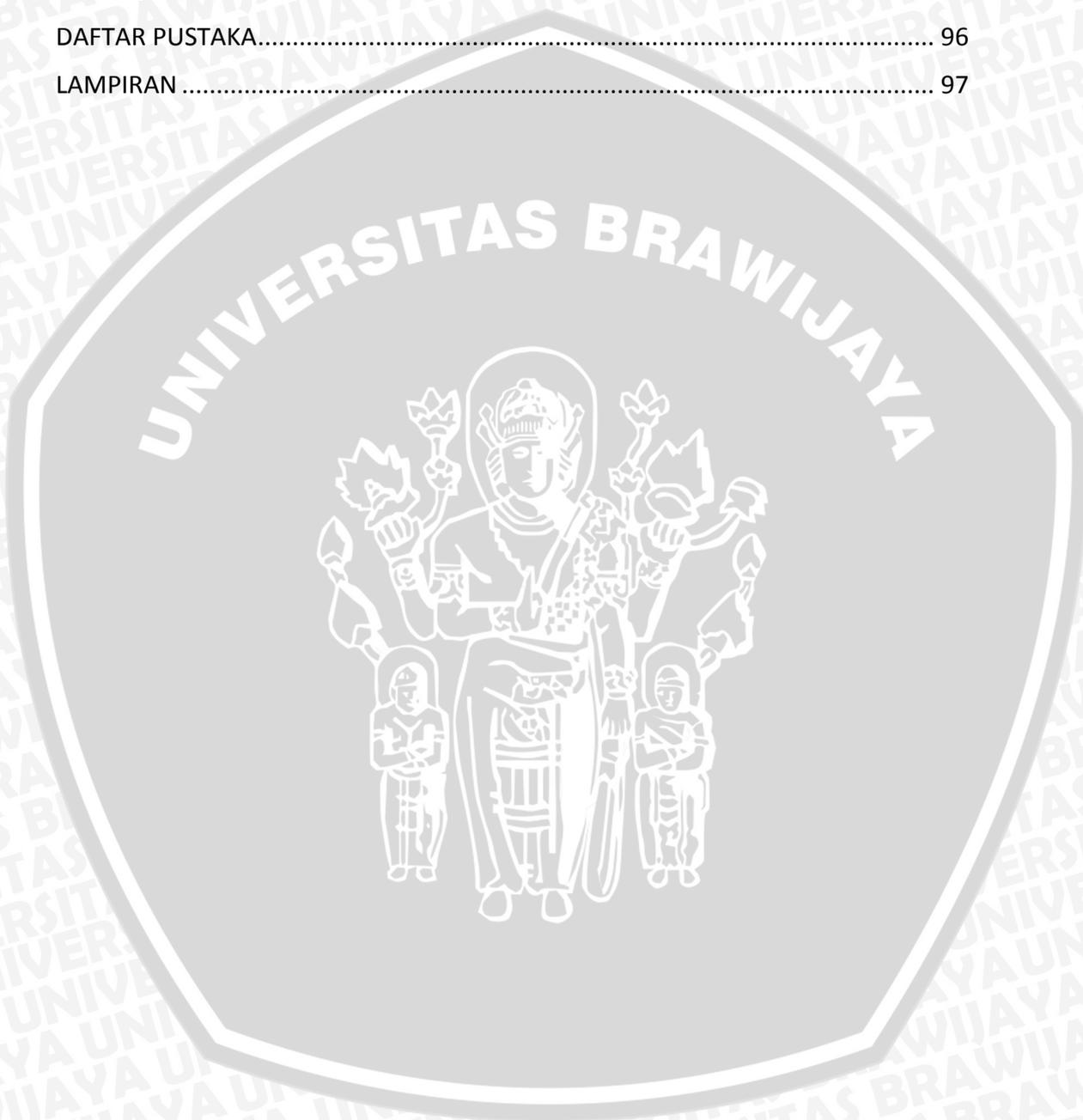
DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Sistem Pakar	7
2.2.1 Pengertian Sistem Pakar	7
2.2.2 Ciri-Ciri Sistem Pakar	7
2.3 Struktur Sistem Pakar	9
2.4 Manfaat Sistem Pakar	12
2.5 Tindak Pidana Penganiayaan	12
2.5.1 Tindak Pidana	12
2.5.2 Tindak Pidana Penganiayaan	12
2.6 Decision Tree.....	14
2.7 Algoritma <i>Iterative Dichotomiser 3</i> (ID3).....	15
2.8 <i>Entropy</i> dan <i>Information Gain</i>	15
BAB 3 METODOLOGI DAN PERANCANGAN.....	17

3.1 Metodologi Penelitian	17
3.1.1 Studi Literatur	18
3.1.2 Pengumpulan Data	18
3.1.3 Analisis Kebutuhan	18
3.1.4 Perancangan	19
3.1.5 Implementasi system	20
3.1.6 Pengujian Sistem	20
3.1.7 Pengambilan Kesimpulan	21
3.2 Perancangan	21
3.2.1 Analisis Kebutuhan Perangkat	22
3.2.2 Perancangan Perangkat Lunak	24
3.2.3 Perancangan Sistem Pakar	31
3.3 Perhitungan Manual	42
BAB 4 IMPLEMENTASI	72
4.1 Spesifikasi Perangkat Lunak dan Perangkat Keras	72
4.2 Implementasi Program	72
4.2.1 Pembacaan Data	72
4.2.2 Perhitungan <i>Entropy</i>	74
4.2.3 Perhitungan <i>Information Gain</i>	75
4.2.4 <i>GetRoot</i>	76
4.2.5 Gambar <i>Tree</i>	77
4.2.6 Pencarian PasaI	78
4.2.7 Pencarian Lama Hukuman	84
4.3 Implementasi Antarmuka	86
4.3.1 Login	87
4.3.2 Antarmuka Pakar	87
4.3.3 Antarmuka <i>Training</i>	88
4.3.4 Homepage	89
4.3.5 Proses Interogasi	90
BAB 5 PENGUJIAN DAN ANALISIS	91
5.1 Pengujian Fungsionalitas	91
5.2 Pengujian Akurasi	92



5.2.1 Hasil Uji Coba	92
BAB 6 PENUTUP	95
6.1 Kesimpulan.....	95
6.2 Saran	95
DAFTAR PUSTAKA.....	96
LAMPIRAN	97



DAFTAR TABEL

Tabel 2.1 tabel perbandingan	6
Tabel 3.1 Teknik Pengambilan Data Sampel	18
Tabel 3.2 Deskripsi Aktor	23
Tabel 3.3 Daftar Kebutuhan Fungsional.....	24
Tabel 3.4 <i>Rule</i>	33
Tabel 3.5 Deskripsi Aktor	42
Tabel 3.6 Nilai <i>Entropy</i> atribut perhitungan manual tahap I.....	45
Tabel 3.7 Nilai <i>Information Gain</i> perhitungan manual tahap I.....	46
Tabel 3.8 Data Tindak Pidana kelas Ya atribut P4.....	47
Tabel 3.9 Nilai <i>Entropy</i> atribut perhitungan manual tahap II.....	48
Tabel 3.10 Nilai <i>Information Gain</i> perhitungan manual tahap II.....	48
Tabel 3.11 Data Tindak Pidana kelas Ya atribut P8.....	50
Tabel 3.12 Nilai <i>Entropy</i> atribut perhitungan manual tahap III.....	50
Tabel 3.13 Nilai <i>Information Gain</i> perhitungan manual tahap III.....	51
Tabel 3.14 Data Tindak Pidana kelas Tidak atribut P4.....	52
Tabel 3.15 Nilai <i>Entropy</i> atribut perhitungan manual tahap IV.....	54
Tabel 3.16 Nilai <i>Information Gain</i> perhitungan manual tahap IV.....	55
Tabel 3.17 Data Tindak Pidana kelas Tidak atribut P2.....	56
Tabel 3.18 Nilai <i>Entropy</i> atribut perhitungan manual tahap V.....	57
Tabel 3.19 Nilai <i>Information Gain</i> perhitungan manual tahap V.....	58
Tabel 3.20 Data Tindak Pidana kelas Tidak atribut P7.....	59
Tabel 3.21 Nilai <i>Entropy</i> atribut perhitungan manual tahap VI.....	60
Tabel 3.22 Nilai <i>Information Gain</i> perhitungan manual tahap VI.....	61
Tabel 3.23 Data Tindak Pidana kelas Ya <i>node</i> P9.....	62
Tabel 3.24 Nilai <i>Entropy</i> atribut perhitungan manual tahap VII.....	63
Tabel 3.25 Nilai <i>Information Gain</i> perhitungan manual tahap VII.....	64
Tabel 3.26 Data Tindak Pidana kelas Tidak <i>Node</i> P1.....	65
Tabel 3.27 Nilai <i>Entropy</i> atribut perhitungan manual tahap VIII.....	66
Tabel 3.28 Nilai <i>Information Gain</i> perhitungan manual tahap VIII.....	66

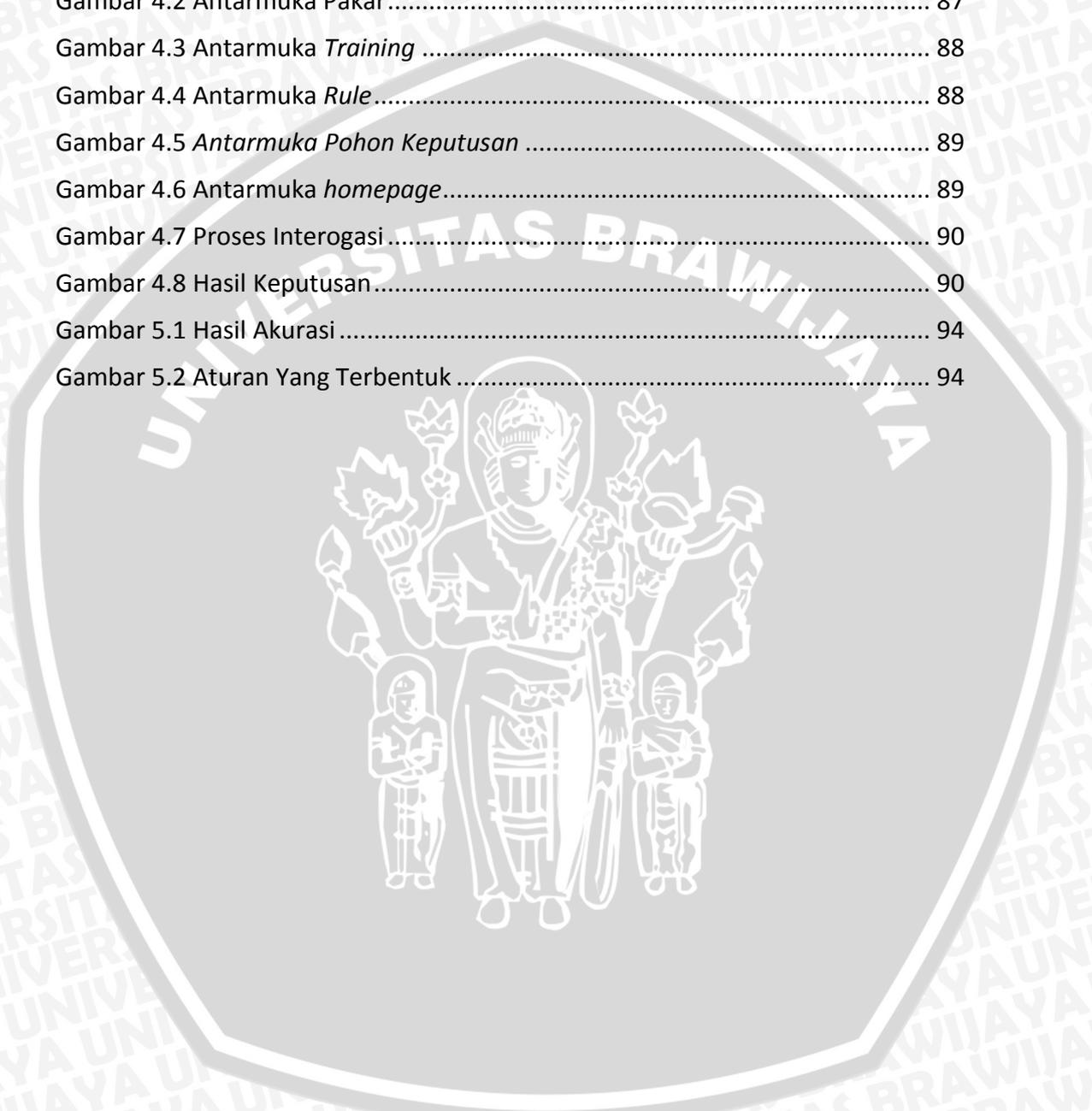
Tabel 5.1 hasil pengujian fungsionalitas 91
Tabel 5.2 Hasil Pengujian Akurasi 92



DAFTAR GAMBAR

Gambar 2.1 Diagram Blok Area Permasalahan.....	5
Gambar 2.2 Struktur sistem pakar	10
Gambar 3.1 <i>Flowchart</i> Perancangan Proses	17
Gambar 3.2 Blok Diagram Perancangan Aplikasi	19
Gambar 3.3 Blok Diagram Implementasi Sistem	20
Gambar 3.4 Blok Diagram Pengujian Sistem.....	21
Gambar 3.5 Pohon Perancangan	22
Gambar 3.6 <i>Flowchart</i> Konsultasi Sistem Pakar	25
Gambar 3.7 <i>Context Diagram</i>	25
Gambar 3.8 DFD Level 0.....	26
Gambar 3.9 DFD Level 1 Proses 2.0	27
Gambar 3.10 DFD Level 1 Proses 3.0	27
Gambar 3.11 DFD Level 1 Proses 4.0	28
Gambar 3.12 DFD Level 1 Proses 5.0	28
Gambar 3.13 <i>Entity Relational Diagram</i> (ERD)	29
Gambar 3.14 <i>Flowchart</i> Sistem Pencarian Pasal.....	35
Gambar 3.15 <i>Flowchart</i> Sistem Pada Perhitungan ID3.....	36
Gambar 3.16 <i>Flowchart</i> Pelatihan	36
Gambar 3.17 <i>Flowchart</i> Pembentukan <i>Tree</i>	37
Gambar 3.18 <i>Flowchart</i> Hitung <i>Entropy</i>	37
Gambar 3.19 <i>Flowchart</i> Perhitungan <i>Entropy</i> total.....	38
Gambar 3.20 <i>Flowchart</i> Perhitungan <i>Entropy</i> Nilai dalam Atribut.....	39
Gambar 3.21 <i>Flowchart</i> Perhitungan <i>Entropy</i> Atribut.....	40
Gambar 3.22 <i>Flowchart</i> <i>GetRoot</i>	41
Gambar 3.23 <i>Tree</i> hasil perhitungan manual tahap I.	46
Gambar 3.24 <i>Tree</i> hasil perhitungan manual tahap II.	49
Gambar 3.25 <i>Tree</i> hasil perhitungan manual tahap III	52
Gambar 3.26 <i>Tree</i> hasil perhitungan manual tahap IV	55
Gambar 3.27 <i>Tree</i> hasil perhitungan manual tahap V	58
Gambar 3.28 <i>Tree</i> hasil perhitungan manual tahap VI.....	61

Gambar 3.29 <i>Tree</i> hasil perhitungan manual tahap VII	64
Gambar 3.30 <i>Tree</i> hasil perhitungan manual tahap VIII	67
Gambar 4.1 Antarmuka <i>Login</i>	87
Gambar 4.2 Antarmuka <i>Pakar</i>	87
Gambar 4.3 Antarmuka <i>Training</i>	88
Gambar 4.4 Antarmuka <i>Rule</i>	88
Gambar 4.5 Antarmuka <i>Pohon Keputusan</i>	89
Gambar 4.6 Antarmuka <i>homepage</i>	89
Gambar 4.7 Proses Interogasi	90
Gambar 4.8 Hasil Keputusan	90
Gambar 5.1 Hasil Akurasi	94
Gambar 5.2 Aturan Yang Terbentuk	94



DAFTAR LAMPIRAN

LAMPIRAN	97
Kasus Hirarki	97
Dataset	99



BAB 1 PENDAHULUAN

1.1 Latar belakang

Indonesia merupakan sebuah negara hukum, sehingga memiliki peraturan-peraturan hukum yang mengatur, salah satunya adalah Kitab Undang-undang Hukum Pidana (KUHP) dan Kitab Undang-Undang Perdata (KUH PERDATA). Kitab Undang-Undang Hukum pidana (KUHP) berisi ratusan pasal yang mengatur tentang hukum pidana di Indonesia, sedangkan Kitab Undang-Undang Perdata (KUH PERDATA) mengatur tentang hukum perdata yang ada di Indonesia.

Semua orang di Indonesia baik tua, muda, laki-laki atau perempuan dianggap mengetahui tentang hukum yang berlaku di Indonesia dan akan dihukum atau dituntut berdasarkan hukum yang berlaku apabila melanggarnya. Begitu suatu norma hukum dibentuk atau ditetapkan maka pada saat itu juga setiap orang di Indonesia diharuskan dan dianggap tahu hukum. Sedangkan ketidaktahuan hukum seseorang bukan menjadi alasan pemberian maaf atau pembebasan orang itu dari hukum.

Masalah hukum ini sangat kompleks sehingga sulit bagi masyarakat awam untuk memahami dan memilah-milah pasal-pasal yang mengatur suatu kasus hukum baik pidana maupun perdata. Masyarakat juga kebanyakan masih buta terhadap pasal-pasal yang ada dalam KUHP dan KUHPerdata, hanya sebagian orang saja yang mengerti akan isi dari kitab undang-undang yang mengatur hukum pidana dan perdata tersebut. Hal ini sering membingungkan bagi masyarakat awam yang buta akan kitab undang-undang saat terlibat dalam suatu kasus hukum sehingga menyebabkan faktor kebingungan bagi masyarakat saat terlibat dalam kasus pidana ataupun perdata, baik sebagai tersangka maupun korban. Adapun salah satu masalah pidana yang sering terjadi di dalam masyarakat adalah tindak pidana terhadap harta kekayaan (tindak pidana materil), seperti: pencurian, pemerasan, penggelapan, penipuan, pengrusakan dan penadahan.

Sedangkan perkembangan teknologi informasi saat ini telah memungkinkan pengambilan keputusan yang dapat dilakukan dengan cepat dan cermat. Selain itu bidang hukum merupakan bidang yang jarang disentuh oleh teknologi informasi. Hal ini dikarenakan bidang ini berkaitan erat dengan masalah yang terjadi di masyarakat, mulai masalah perdata sampai masalah pidana. Dikarenakan masih banyaknya masyarakat yang buta akan undang-undang diperlukan adanya sebuah program komputer sederhana untuk membantu memahami dan memilah-milah pasal-pasal yang terlibat dalam suatu kasus hukum.

Sistem pakar adalah sebuah sistem yang berusaha mengadopsi ilmu dari suatu pakar ke dalam sebuah sistem komputer. Dengan adanya sistem ini diharapkan para pelaku dan penegak hukum dapat mengetahui jenis kategori yang dilanggar, maksimal lama hukuman penjara dan denda maksimal. Sehingga para pelaku hukum tidak buta akan hukum yang ada di Indonesia dan para penegak

hukum tidak salah dalam menentukan keputusan. Selain itu dengan adanya sistem pakar ini diharapkan masyarakat awam bisa lebih tahu tentang pelanggaran dan hukumannya, masyarakat lebih paham tentang pasal berlapis yang biasanya sering menjerat tersangka, minimal masyarakat awam tahu hukuman maksimal dari pasal yang di langgar, sehingga masyarakat dapat sedikit lebih aktif dalam mengkritisi sekaligus mengawasi kasus yang ada ketika sang hakim sedikit melenceng dari ketetapan KUHP yang ada. Dikarenakan bab dan pasal yang ada dalam KUHP sangat banyak dan terlalu umum maka pada skripsi ini lebih difokuskan membahas terkait dengan bab penganiayaan saja. proses penjaringan pasal yang dilanggar beserta maksimal hukuman akan diteliti menggunakan metode *forward chaining* dan metode *Iterative Dichotomiser 3* (ID3).

Beberapa penelitian terkait dengan metode *Iterative Dichotomiser 3* (ID3) diantaranya yaitu penelitian oleh Selly Johansyah Isamuddin tentang Klasifikasi Keluarga Sejahtera Menggunakan Metode *Decision Tree* Dengan Algoritma *Iterative Dichotomiser 3* (ID3) Pada Kecamatan Batu. Sedangkan sistem pakar yang terkait tentang tindak pidana diantaranya yaitu penelitian oleh Fendhy Ongko tentang Perancangan Dan Pembuatan Aplikasi Sistem Pakar Untuk Permasalahan Tindak Pidana Terhadap Harta Kekayaan. Dan juga penelitian oleh Aditya Pranajaya membahas tentang Aplikasi Sistem Pakar Untuk Permasalahan Tindak Pidana Yang Mengakibatkan Kehilangan Nyawa Berbasis Web. Kedua penelitian di atas sama sama menggunakan metode *forward chaining* tetapi dikombinasikan dengan metode lainnya. Untuk itu maka pada skripsi ini proses penjaringan pasal yang dilanggar beserta maksimal hukuman akan diteliti menggunakan metode *Iterative Dichotomiser 3* (ID3).

1.2 Rumusan masalah

Dengan melihat latar belakang masalah yang sudah dijabarkan diatas maka rumusan masalah yang ada adalah :

1. Bagaimana merancang sistem pakar penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan?
2. Bagaimana menerapkan Metode *Rule based* dan *Iterative Dichotomiser 3* (ID3) sebagai salah satu metode dalam membuat sistem pakar penentuan hukuman dan denda pada permasalahan tindak pidana penganiayaan?
3. Bagaimana uji akurasi penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan?

1.3 Tujuan

Tujuan dari penelitian ini adalah :

1. Untuk merancang sistem pakar penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan.

2. Untuk menerapkan Metode *Rule based* dan *Iterative Dichotomiser 3* (ID3) sebagai salah satu metode dalam membuat sistem pakar penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan.
3. Untuk uji akurasi sistem pakar penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan

1.4 Manfaat

Manfaat yang didapat dari penulisan skripsi ini adalah

a. Bagi penulis

1. Menerapkan ilmu yang telah diperoleh dari Teknik Informatika Universitas Brawijaya.
2. Memahami penerapan metode *Iterative Dichotomiser 3* (ID3) dalam perancangan dan pengembangan sistem pakar untuk penentuan kategori pasal yang dilanggar dan lama hukuman
3. Menambah pengetahuan mengenai masalah undang-undang yang berlaku di Indonesia khususnya bab penganiayaan.

b. Bagi pengguna

1. Memudahkan pencarian informasi dalam penentuan pasal yang dilanggar tersangka
2. Memberikan informasi dalam penentuan lama hukuman yang dijatuhkan.

1.5 Batasan masalah

Agar tidak memperluas area pembahasan dalam penelitian skripsi ini, maka perlu adanya batasan-batasan untuk menyederhanakan permasalahan, yaitu :

1. Analisis sistem berdasarkan hanya pada KUHP (kitab undang-undang hukum pidana) bab penganiayaan.
2. Tidak membahas tentang keamanan data dalam basis data yang digunakan.
3. Keluaran sistem yaitu pasal yang dilanggar dalam bab penganiayaan, maksimal lama hukuman yang divoniskan.
4. Pembangunan sistem pakar dengan metode *Rule based* dan *Iterative Dichotomiser 3* (ID3).
5. Parameter yang digunakan dalam metode *Iterative Dichotomiser 3* (ID3) pada penelitian ini ada 11.

1.6 Sistematika pembahasan

Agar dalam penulisan skripsi ini dapat lebih terarah, maka penulis berusaha sedapat mungkin menyusun secara sistematis sehingga diharapkan

tahap-tahap pembahasan akan tampak jelas kaitannya antara bab yang satu dengan bab yang lainnya. Adapun isi dari masing-masing bab tersebut adalah sebagai berikut

Bab I Pendahuluan

Berisi gambaran umum tentang latar belakang permasalahan, rumusan inti masalah yang dihadapi, tujuan dan manfaat penelitian, batasan masalah, metodologi penelitian, keaslian penelitian dan sistematika penulisan.

Bab II Kajian Pustaka dan Dasar Teori

Bab ini membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian. Pada bab ini membahas pengertian sistem pakar, deskripsi tentang hukum di Indonesia, pasal-pasal dalam KUHP bab penganiayaan.

Bab III Metode Penelitian dan Perancangan

Menguraikan tentang metode, membahas analisis kebutuhan dan perancangan sistem pakar penentuan hukuman dan denda untuk permasalahan tindak pidana penganiayaan.

Bab IV Implementasi

Bab ini menjelaskan tentang implementasi dari *Forward chaining* dan *Iterative Dichotomiser 3 (ID3)* dalam perancangan dan pengembangan sistem pendukung keputusan untuk penentuan kategori pasal yang dilanggar dan lama hukuman serta maksimal denda sesuai dengan perancangan sistem yang telah dibuat.

Bab V Pengujian dan Analisis

Berisi proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

Bab VI Penutup

Berisi kesimpulan dan saran atas hasil perancangan sistem terintegrasi yang terkait langsung dengan topik yang dibahas.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Bab ini membahas kajian pustaka dan dasar teori yang digunakan untuk menunjang penulisan skripsi mengenai sistem pakar penentuan hukuman dan denda untuk permasalahan tindak pidana penganiayaan. Sistem Pakar untuk penentuan kategori pasal yang dilanggar dan maksimal lama hukuman. Beberapa dasar teori yang dimaksud adalah Sistem Pakar (SISPAK), KUHP bab penganiayaan, *Rule Based* dan *Iterative Dichotomiser 3* (ID3).

Pada penelitian sebelumnya yaitu dengan judul “perancangan dan pembuatan aplikasi sistem pakar untuk permasalahan tindak pidana terhadap harta kekayaan” telah membuat sistem pakar untuk menentukan jenis tindak pidana dan pasal-pasal yang berhubungan dengan kasus yang menyangkut harta kekayaan. Diagram blok berikut menjelaskan apa saja yang termasuk dalam pidana terhadap harta kekayaan [ONG-04].



Gambar 2.1 Diagram Blok Area Permasalahan

Sumber : [ONG-04]

Pada penelitian lain yang membahas tentang permasalahan tindak pidana yang mengakibatkan kehilangan nyawa berbasis web, lebih menghususkan membahas tentang sistem pakar untuk menentukan tindak pidana yang menyangkut dengan kasus pembunuhan atau menghilangkan nyawa seseorang. Penelitian tersebut menggunakan metode *Forward chaining* dan *Depth First Search* (DFS) [ADP-10]. Sedangkan penelitian yang menggunakan metode ID3 adalah pada penelitian berjudul *Klasifikasi Keluarga Sejahtera Menggunakan Metode Decision Tree Dengan Algoritma Iterative Dichotomiser 3* (ID3) Pada Kecamatan Batu yang dikerjakan oleh Selly Johansyah Isamuddin. Sistem ini dibangun menggunakan metode *Decision Tree* dengan algoritma *Iterative Dichotomiser 3* (ID3) [SEL-15]. Hasil uji coba dari skenario uji coba didapatkan akurasi terbaik sebesar 88% dari 28 data latih seimbang [SEL-15]. Penelitian lain juga menyebutkan bahwa hasil penerapan *Decision Tree* dengan algoritma ID3 dapat berjalan baik dan menghasilkan akurasi yang cukup memuaskan.

Perbedaan yang dibuat penulis pada penelitian ini adalah implementasi metode *forward chaining* dan menggunakan *Iterative Dichotomiser 3* (ID3) untuk menentukan hukuman dan denda yang dikenakan. Sedangkan pada sistem pakar di atas hanya menggunakan metode *forward chaining* saja. Selain itu fokus bab pada KUHP yang dibahas juga berbeda, peneliti di sini lebih memfokuskan ke dalam bab penganiayaan dalam KUHP sedangkan pada penelitian sebelumnya membahas tentang bab yang menyangkut harta kekayaan dan penelitian yang lain lebih fokus terhadap pasal-pasal yang berhubungan dengan perbuatan yang mengakibatkan hilangnya nyawa seseorang. Perbedaan ini mengakibatkan proses penentuan pasal-pasal dan hukuman serta bab dalam hukum yang dibahas berbeda dengan penelitian sebelumnya. Perbedaan yang lain adalah aplikasi yang dihasilkan berupa aplikasi desktop, sehingga dapat lebih mudah digunakan kapan saja dan dimana saja karena tidak diperlukan akses ke jaringan internet. Table perbandingan penelitian dapat dilihat pada tabel 2.1 di bawah ini.

Tabel 2.1 tabel perbandingan

No.	Nama Peneliti	Judul Penelitian	Metode	Output
1.	Fendhy Ongko (2004)	<i>Perancangan Dan Pembuatan Aplikasi Sistem Pakar Untuk Permasalahan Tindak Pidana Terhadap Harta Kekayaan.</i>	- Metode <i>Rule-based</i> dan inferensi <i>forward chaining</i>	- Pasal-pasal apa saja yang terlibat dalam sebuah tindak pidana terhadap harta kekayaan
2.	Aditya Pranajaya (2010)	<i>Aplikasi Sistem Pakar Untuk Permasalahan Tindak Pidana Yang Mengakibatkan Kehilangan Nyawa Berbasis Web.</i>	- Menggunakan metode <i>Forward chaining</i> dan <i>Depth First Search (DFS)</i> .	- Pasal-pasal apa saja yang terlibat dalam sebuah tindak pidana yang mengakibatkan kehilangan nyawa.
3	Selly Johansyah Isamuiddin (2015)	<i>Klasifikasi Keluarga Sejahtera Menggunakan Metode Decision Tree Dengan Algoritma Iterative Dichotomiser 3 (Id3)</i>	Sistem ini dibangun menggunakan metode <i>Decision Tree</i> dengan algoritma <i>Iterative Dichotomiser</i>	- Output berupa status tahapan keluarga sejahtera yang terdapat 5 kategori.

			3 (ID3) [SEL-15]	
4	Achmad Dwi Ardian (2015)	<i>Sistem Pakar Penentuan Pasal dan Lama Hukuman Untuk Permasalahan Tindak Pidana Penganiayaan</i>	Menggunakan metode <i>Forward chaining</i> dan <i>Iterative Dichotomiser 3 (ID3)</i> .	- Pasal yang dilanggar dalam bab penganiayaan, berapa lama hukuman yang divonis.

2.2 Sistem Pakar

Sistem pakar merupakan cabang dari *Artificial Intelligence (AI)* yang cukup tua karena sistem ini mulai dikembangkan pada pertengahan 1960. Sistem pakar yang muncul pertama kali adalah *General-purpose problem solver (GPS)* yang dikembangkan oleh Newel dan Simon. Sampai saat ini sudah banyak sistem pakar yang telah dibuat [SEV-11:159].

2.2.1 Pengertian Sistem Pakar

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud disini adalah orang yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sistem pakar mencoba memecahkan masalah yang biasanya hanya bisa dipecahkan oleh seorang pakar, dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi proses pengambilan keputusan yang diperoleh [KSN-08:3].

Menurut Fegenbum dari universitas Standford "sistem pakar adalah sebuah program komputasi cerdas yang menggunakan pengetahuan dan prosedur inferensi untuk memecahkan masalah yang cukup sulit, yang membutuhkan keahlian manusia yang signifikan sebagai solusi". Sistem pakar berfungsi dengan baik dalam domain terbatas dari mereka [KUM-08:417].

2.2.2 Ciri-Ciri Sistem Pakar

Ciri – ciri dari sistem pakar adalah sebagai berikut :

1. Terbatas pada domain keahlian tertentu.
2. Dapat memberikan penalaran untuk data – data yang tidak lengkap atau tidak pasti.
3. Dapat menjelaskan alasan – alasan dengan cara yang dapat dipahami.
4. Bekerja dengan kaidah atau *Rule* tertentu.
5. Mudah dimodifikasi.
6. Basis pengetahuan dan mekanisme inferensi terpisah.
7. Keluarnya bersifat anjuran.

8. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntut oleh dialog dengan pengguna [SEV-11:162].

2.2.2.1 Konsep Dasar Sistem Pakar

Sistem pakar mencakup enam komponen utama adalah [SEV-11:163-166]:

A. Kepakaran

Kepakaran merupakan suatu pengetahuan yang diperoleh dari pelatihan pengalaman. Kepakaran inilah yang memungkinkan para ahli dapat mengamati keputusan yang lebih cepat dan lebih baik dari pada seseorang yang bukan pakar. Kepakaran meliputi pengetahuan tentang :

1. Fakta –fakta tentang bidang permasalahan tertentu.
2. Teori – teori tentang bidang permasalahan tertentu.
3. Aturan – aturan dan prosedur menurut bidang dan permasalahan tertentu.
4. Aturan *heuristic* yang harus dikerjakan dalam suatu situasi tertentu.
5. Strategi global untuk memecahkan permasalahan.
6. Pengetahuan tentang pengetahuan (*meta knowledge*).

B. Pakar

Pakar adalah seseorang yang mempunyai pengetahuan, pengalaman, dan metode khusus, serta mampu menerapkannya untuk memecahkan masalah atau memberi nasihat. Seorang pakar harus mampu menjelaskan dan mempelajari hal-hal baru yang berkaitan dengan topik permasalahan, jika perlu harus mampu menyusun kembali pengetahuan-pengetahuan yang didapatkan, dan dapat memecahkan masalah aturan-aturan serta menentukan relevansi kepakarannya.

C. Pemindahan Kepakaran (*Transferring Expertise*)

Tujuan dari sistem pakar adalah memindahkan kepakaran dari seorang pakar ke dalam komputer, kemudian ditransfer kepada orang lain yang bukan pakar. Proses ini melibatkan empat kegiatan, yaitu :

1. Akuisisi pengetahuan (dari pakar atau sumber lain).
2. Representasi pengetahuan.
3. Inferensi pengetahuan.
4. Pemindahan pengetahuan ke pengguna.

D. Inferensi (*Inferencing*)

Inferensi adalah sebuah prosedur (program) yang mempunyai kemampuan dalam melakukan penalaran. Inferensi ditampilkan pada suatu komponen yang disebut mesin inferensi yang mencakup prosedur – prosedur mengenai pemecahan masalah. Semua pengetahuan yang dimiliki oleh seorang pakar disimpan pada basis pengetahuan oleh sistem pakar. Tugas mesin inferensi adalah mengambil kesimpulan berdasarkan basis pengetahuan yang dimilikinya.

E. Aturan – aturan (*Rule*).

Kebanyakan *software* sistem pakar komersial adalah sistem yang berbasis pengetahuan *Rule (Rule-based systems)*, yaitu pengetahuan disimpan terutama dalam bentuk *Rule*, sebagai prosedur – prosedur pemecahan masalah.

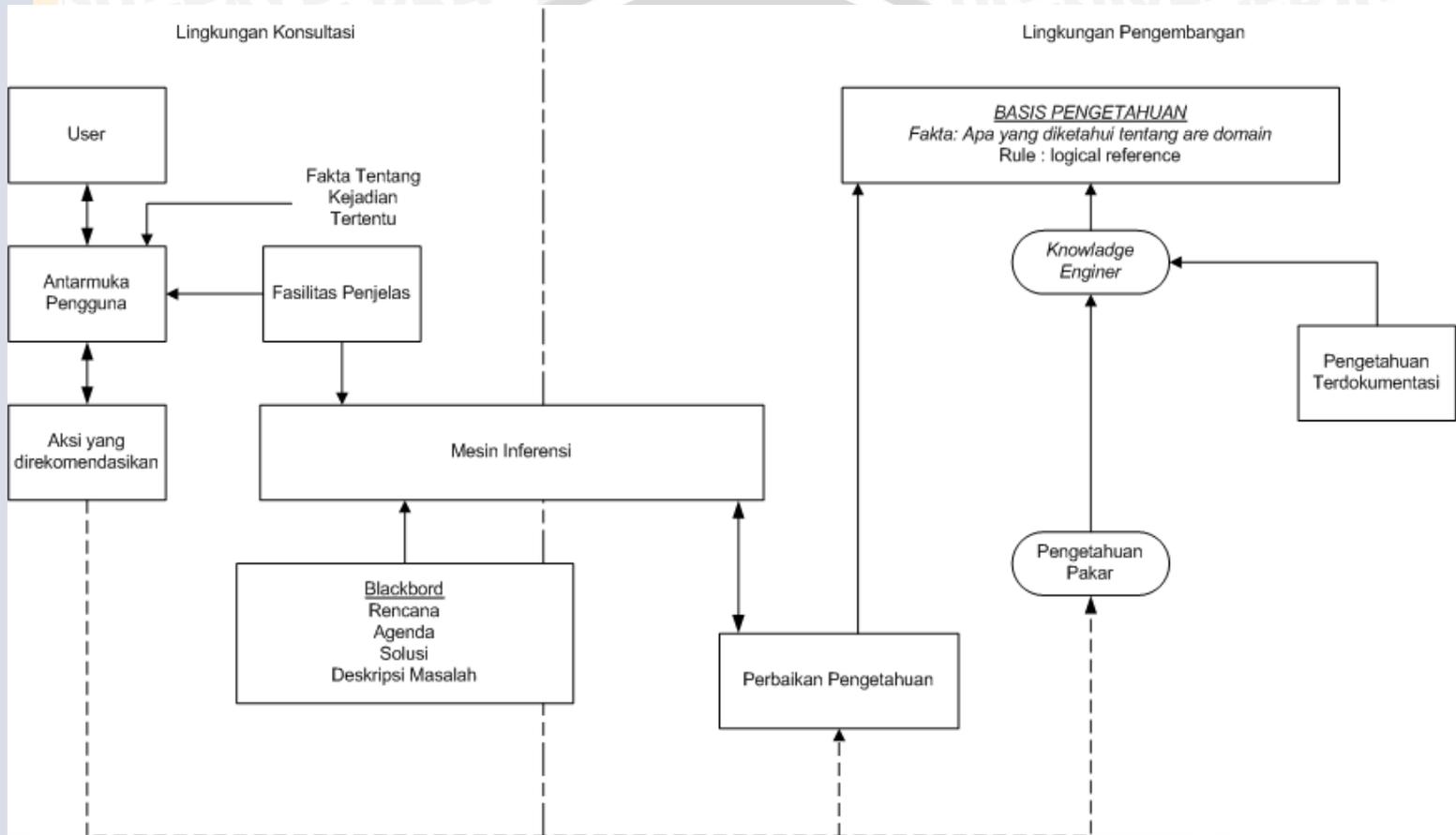
F. Kemampuan Menjelaskan (*Explantion Capability*)

Fasilitas lain dari sistem pakar adalah kemampuannya untuk menjelaskan saran atau rekomendasi yang diberikannya. Penjelasan dilakukan dalam subsistem yang disebut subsistem penjelasan (*explantion*). Bagian dari sitem untuk memeriksa penalaran yang dibuatnya sendiri dan menjelaskan operasi – operasinya.

2.3 Struktur Sistem Pakar

Ada dua bagian penting dari sistem pakar, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultion environment*). Lingkungan pengembangan digunakan oleh pembuat sistem pakar untuk membangun komponen – komponen yang memperkenalkan pengetahuan kedalam *knowledge base* (*basis pengetahuan*). Lingkungan konsultasi digunakan untuk berkonsultasi sehingga pengguna mendapatkan pengetahuan dan nasihat dari sistem pakar layaknya berkonsultasi dengan sorang pakar. Gambar 2.3 merupakan arsitektur sistem pakar.





Gambar 2.2 Struktur sistem pakar

Sumber :[ETB-05:722]

Keterangan gambar :



: Langsung



: Pemisan antara lingkungan dan pengembangan



: Tidak Langsung



: Komunikasi dua arah

Penjelasan struktur sistem pakar adalah sebagai berikut[ETB-05-722]:

1. Basis pengetahuan

Basis pengetahuan berisi pengetahuan yang relevan untuk memahami merumuskan dan memecahkan persoalan. Basis tersebut mencakup dua elemen dasar yaitu fakta dan *heuristic*. Fakta merupakan situasi persoalan dan teori area persoalan, dan *heuristic* adalah aturan khusus yang mengarahkan penggunaan pengetahuan untuk memecahkan persoalan khusus dalam domain tertentu.

2. Mesin inferensi

Mesin inferensi dikenal juga sebagai struktur kontrol atau penerjemah aturan. Komponen ini sebenarnya adalah program komputer yang menyediakan metodologi untuk mempertimbangkan informasi dalam basis pengetahuan dan *blackboard*, dan menarik kesimpulan. Komponen ini menyediakan arahan bagaimana menggunakan pengetahuan sistem, yakni dengan mengembangkan agenda yang mengatur dan mengontrol langkah yang diambil untuk memecahkan permasalahan dan konsultasi berlangsung.

3. Antarmuka pengguna

Sistem pakar berisi prosesor bahasa untuk komunikasi berorientasi persoalan yang mudah antar muka pengguna dan komputer. Komunikasi ini paling baik dilakukan dalam bahasa alami. Dikarenakan alasan teknologi, maka kebanyakan sistem yang ada menggunakan pendekatan pertanyaan dan jawaban untuk berinteraksi dengan pengguna.

4. *Blackboard*

Blackboard adalah area kerja memori yang disimpan sebagai *database* untuk deskripsi persoalan terbaru yang ditetapkan oleh data *input* digunakan juga untuk perekaman hipotesis dan keputusan sementara. Tiga tipe keputusan yang direkam pada *blackboard*: rencana (bagaimana mengatasi persoalan), agenda (tindakan potensial sebelum eksekusi), solusi.

5. Subsistem penjelasan (*justifier*)

Kemampuan untuk melacak tanggung jawab suatu kesimpulan terhadap sumbernya adalah penting untuk transfer keahlian dan dalam pemecahan masalah. Subsistem penjelasan dapat melacak tanggung jawab tersebut dan menjelaskan perilaku inferensi dengan menjawab pertanyaan sebagai interaktif.

6. Sistem perbaikan pengetahuan

Pakar manusia memiliki sistem perbaikan pengetahuan, yakni mereka dapat menganalisis pengetahuannya sendiri dan kegunaannya, belajar darinya, dan meningkatkan untuk konsultasi mendatang. Evaluasi tersebut diperlakukan

dalam pembelajaran komputer sehingga program dapat menganalisis alasan keberhasilan atau kegagalannya. Hal ini dapat mengarah kepada peningkatan sehingga menghasilkan basis pengetahuan yang lebih akurat dan pertimbangan yang lebih efektif.

7. Mesin inferensi

Mesin inferensi adalah modul paling kritis agar sistem pakar dapat berfungsi dengan baik. Pengetahuan harus direpresentasikan dan diatur secara tepat dalam basis pengetahuan. Mesin inferensi kemudian dapat menggunakan pengetahuan tersebut untuk menarik kesimpulan baru dan fakta dari aturan yang ada.

8. User

Pada umumnya penggunaan sistem pakar bukanlah seorang pakar (*non-expert*) yang membutuhkan solusi, saran atau pelatihan (*training*) dari berbagai permasalahan yang ada [SEV_11:169].

2.4 Manfaat Sistem Pakar

Sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya, antara lain [SEV-11:160-161]:

1. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat dari pada manusia.
2. Membuat seseorang yang bekerja seperti layaknya seorang pakar.
3. Meningkatkan kualitas dengan memberi nasihat yang konsisten dan mengurangi kesalahan.
4. Mampu menangkap pengetahuan dan kepakaran seorang.
5. Dapat beroperasi dilingkungan yang berbahaya.
6. Memudahkan akses pengetahuan seorang pakar.

2.5 Tindak Pidana Penganiayaan

2.5.1 Tindak Pidana

Tindak pidana adalah perbuatan yang melanggar larangan yang diatur oleh aturan hukum yang diancam dengan sanksi. Juga dapat diartikan sebagai perbuatan yang dilarang oleh suatu aturan hukum larangan mana disertai ancaman(sanksi) yang berupa pidana tertentu bagi barang siapa yang melanggar larangan tersebut [PWI-03]. Kata perbuatan dalam perbuatan pidana mempunyai arti antara lain :

- a. Adanya kejadian tertentu yang menimbulkan akibat yang dilarang.
- b. Adanya orang yang berbuat sehingga menimbulkan kejadian itu

2.5.2 Tindak Pidana Penganiayaan

Secara umum tindak pidana terhadap tubuh pada KUHP disebut penganiayaan. Sedangkan KUHP sendiri tidak memberikan penjelasan tentang apa yang dimaksud dengan intilah penganiayaan selain hanya menyebut penganiayaan saja.

Apabila suatu penganiayaan mengakibatkan luka berat, maka sesuai Pasal 351 ayat (2) KUHP maksimum hukuman dijadikan 5 (lima) tahun penjara. Sedangkan jika berakibat matinya orang, maka maksimum hukuman meningkat lagi menjadi 7 (tujuh) tahun penjara [KUHP-81].

Dua macam akibat ini harus tidak dituju dan juga harus tidak disengaja, sebab kalau melukai berat ini disengaja, maka ada tindak pidana penganiayaan berat dan Pasal 354 ayat (1) KUHP dengan maksimum hukuman delapan tahun penjara. Hukuman itu menjadi 10 (sepuluh) tahun penjara jika perbuatan ini mengakibatkan matinya orang, sedangkan kalau matinya orang disengaja, tindak pidana menjadi pembunuhan yang diancam dengan maksimum 15 (lima belas) tahun penjara.

Istilah luka berat sesuai Pasal 90 KUHP berarti sebagai berikut [KUHP-81]:

1. Penyakit atau luka yang tidak dapat diharapkan akan sembuh dengan sempurna atau yang menimbulkan bahaya maut;
2. Menjadi senantiasa tidak cakap mengerjakan pekerjaan jabatan atau pencaharian;
3. Kekudung-kudungan;
4. Kelumpuhan;
5. Gangguan daya berpikir selama lebih dan 4 (empat) minggu;
6. Pengguguran kehamilan atau kematian anak yang masih ada dalam kandungan.

Dalam menjatuhkan lama hukuman yang akan dijatuhkan kepada tersangka, para hakim mempunyai penilaian dalam memutuskan suatu persidangan. Akan tetapi berikut merupakan 11 parameter dasar yang digunakan oleh para hakim untuk menentukan berapa lamanya hukuman yang akan dijatuhkan kepada tersangka:

Dalam menjatuhkan lama hukuman yang akan dijatuhkan kepada tersangka, para hakim mempunyai parameter penilaian yang sangat berpengaruh terhadap keputusan persidangan. Diantara parameter penilaian tersebut terdapat beberapa poin yang dapat memberatkan tersangka dan beberapa poin lain yang dapat meringankan tersangka. Berikut 11 parameter penilaian hakim, 4 diantaranya adalah parameter yang memberatkan dan 7 diantaranya adalah parameter yang meringankan.

Meringankan :

1. Terdakwa belum pernah dihukum sebelumnya.
2. Terdakwa masih dibawah umur.
3. Dalam persidangan terdakwa berkata jujur dan memperlancar proses persidangan.
4. Terdakwa mempunyai istri dan beberapa anak kecil.
5. Terdakwa berusia lanjut dan sering sakit-sakitan (60).
6. Terdakwa menyerahkan diri kepada yang berwajib dulu dan mengakui kesalahannya.
7. Terdakwa sedang mengandung \pm 3 bulan

Memberatkan :

8. Pada waktu melakukan kejahatan terdakwa sedang dalam masa percobaan.
9. Terdakwa berbelit-belit dan mempersulit jalannya persidangan.
10. Terdakwa mengancam para saksi.
11. Korban berjumlah lebih dari 1 orang.

2.6 Decision Tree

Decision Tree merupakan salah satu fungsional dari data mining yang menggunakan representasi *tree* untuk menentukan aturan-aturan klasifikasi. *Decision Tree* dapat juga dikatakan sebagai *Flowchart* seperti struktur *tree*, dimana tiap *node* internal menunjukkan sebuah tes pada sebuah atribut, tiap cabang menunjukkan hasil dari tes, dan setiap *node leaf* menunjukkan kelas-kelas atau distribusi kelas. Metode *Decision Tree* sangat terkenal daripada metode klasifikasi yang lainnya, karena metode ini tidak membutuhkan pengetahuan yang lebih atau pengaturan parameter [HAN-01].

Decision Tree adalah sebuah struktur pohon, dimana setiap *node* pohon merepresentasikan atribut yang telah diuji, setiap cabang merupakan suatu pembagian hasil uji, dan *node* daun (*leaf*) merepresentasikan kelompok kelas tertentu. Level *node* teratas dari sebuah *Decision Tree* adalah *node* akar (*root*) yang biasanya berupa atribut yang paling memiliki pengaruh terbesar pada suatu kelas tertentu. Salah satu algoritma *Decision Tree* yaitu ID3 (*Iterative Dichotomiser 3*). ID3 dikembangkan oleh J. Ross Quinlan. Algoritma ID3 dapat diimplementasikan menggunakan fungsi *rekursif*. Algoritma ID3 berusaha membangun *Decision Tree* secara *top-down*, mulai dengan pertanyaan : “atribut mana yang pertama kali harus dicek dan diletakkan pada *root*?” pertanyaan ini dijawab dengan mengevaluasi semua atribut yang ada dengan menggunakan suatu ukuran statistik (*Information Gain*) untuk mengukur efektivitas suatu atribut dalam mengklasifikasikan kumpulan sampel data [WYD-09].

Proses pembangunan *tree* ini dilakukan dengan cara membagi data secara rekursif atau terus menggunakan proses yang sama, hingga tiap bagian terdiri dari beberapa sampel yang berasal dari kelas yang sama. Ketika sampel tidak dalam kelas yang sama maka dilakukan perhitungan *Information Gain*. Suatu atribut yang telah dipilih menjadi tes atribut maka atribut tersebut tidak diikuti lagi dalam perhitungan *Information Gain* [TAN-06].

Pembangunan *tree* dilakukan dengan sebuah algoritma induksi *decision tree*. Rangka algoritma induksi *Decision Tree* disebut *TreeGrowth*. Cara kerja algoritma ini yaitu dengan memilih atribut terbaik dengan cara menghitung *Information Gain* dari atribut – atribut yang ada untuk memisahkan data secara rekursif dan mengembangkan *node leaf* pada *tree* sampai ditemui kriteria untuk berhenti [KOH-99].

Sebuah dataset S terdiri dari n kasus dengan m masukan atribut X dan satu output (tergantung) atribut Y (numerik atau kategori). Tujuan dari algoritma pohon keputusan (*Decision Tree*) adalah untuk menemukan model pohon keputusan itu, berdasarkan nilai-nilai X , mengklasifikasikan kasus ke dalam kelas Y

(model klasifikasi), atau memperkirakan nilai Y (model regresi). Sebuah model pohon keputusan terdiri dari *node*, cabang, dan daun. Daun menentukan prediksi Y, dan aturan yang dapat diperoleh pada pohon keputusan (*Decision Tree*) secara menurun dari akar ke daun. Jumlah daun mendefinisikan jumlah aturan yang dapat diekstraksi dari pohon keputusan. Pohon keputusan (*Decision Tree*) cocok digunakan untuk mendapatkan aturan IF-THEN dalam struktur seperti pohon hirarkis [SUK-11].

2.7 Algoritma Iterative Dichotomiser 3 (ID3)

Iterative Dichotomiser 3 (ID3) adalah salah satu algoritma yang sering digunakan pada pembelajaran dan data mining karena mudah digunakan dan cukup efektif. Algoritma ini dikembangkan oleh J.Rose Quinlan pada tahun 1986. Algoritma ini membangun pohon keputusan dari beberapa data untuk proses klasifikasi dan menentukan klasifikasi data yang baru [LIG-05]. Algoritma ID3 membutuhkan data yang telah memiliki class untuk pembelajaran sebelum melakukan proses klasifikasi. Kurangnya data pembelajaran dapat berakibat kurang optimalnya klasifikasi yang dihasilkan dari algoritma ini.

2.8 Entropy dan Information Gain

Ukuran *Information Gain* digunakan untuk memilih tes atribut pada setiap simpul dalam *tree*. Atribut dengan informasi tertinggi (nilai pengurangan *Entropy* yang terbesar) dipilih sebagai tes atribut untuk simpul tersebut. Atribut ini meminimalkan informasi yang dibutuhkan untuk mengklarifikasikan contoh pada proses pembagian dan mencerminkan ketidakmurnian (*impurity*) [KOH-99].

Persamaan perhitungan adalah persamaan yang dijabarkan oleh Shannon pada tahun 1948 [SUK-11], yakni sebagai berikut :

$$E(S) = - \sum_{j=1}^n f_s(j) \log_2 f_s(j) \tag{2.1}$$

Dimana :

- E(S) adalah *Entropy* dari himpunan S
- N adalah jumlah nilai yang berbeda dari atribut di S
- $f_s(j)$ adalah frekuensi dari nilai j dalam himpunan s

Entropy 0 menandakan satu set telah diklasifikasi dengan sempurna. Perhitungan *Information Gain* kemudian dilakukan setelah nilai *Entropy* didapat dengan menggunakan persamaan :

$$G(S, A) = E(S) - \sum_{i=1}^m f_s(A_i) E(S_{A_i}) \tag{2.2}$$

Dimana :

- G(S,A) adalah nilai gain dari atribut A pada himpunan S
- M adalah jumlah nilai-nilai yang berbeda dari atribut A di S
- $f_s(A_i)$ frekuensi dari item yang memiliki A_i sebagai nilai dari A di S
- A_i adalah i^{th} nilai yang mungkin dari A



- S_{A_i} adalah bagian dari S yang berisi semua item dimana nilai A adalah A_i

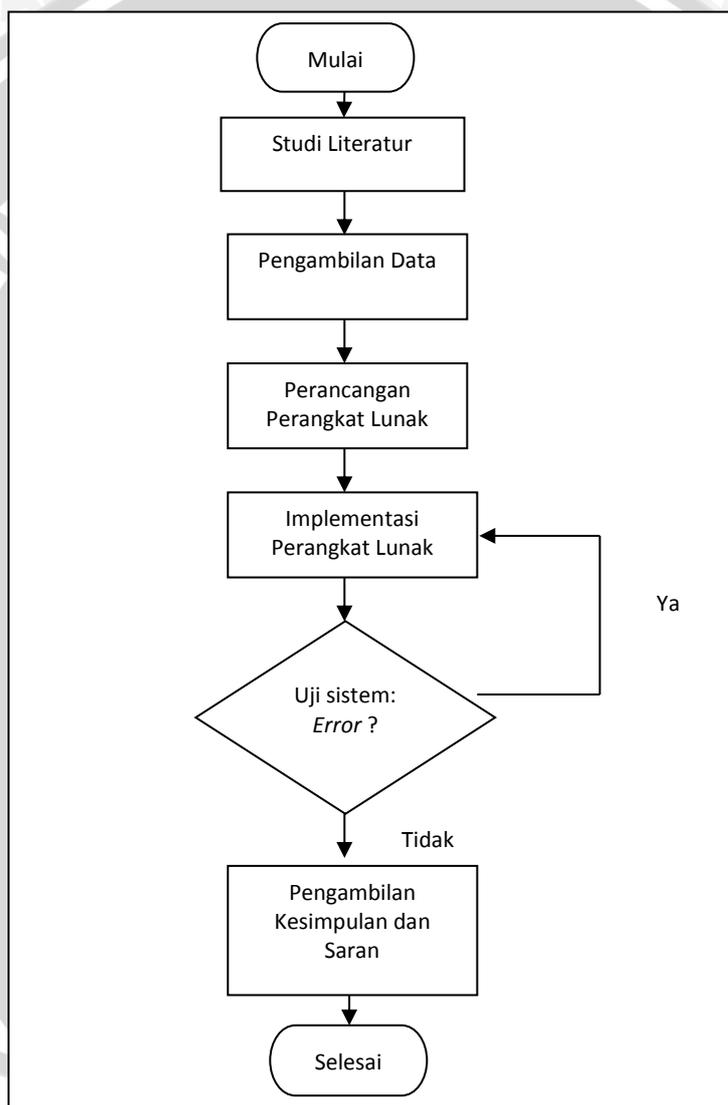
Algoritma ID3 dalam proses pembentukan *treenya* dimulai dengan mencari *best attribute*, yakni atribut yang memiliki nilai terbaik untuk memisahkan data ke dalam *class* nya. Penghitungan *best attribute* itu dimulai dengan menghitung *Entropy* dan dilanjutkan dengan perhitungan *Information Gain*. Atribut dengan *Information Gain* yang terbesar lah yang merupakan *best attribute* yang kemudian dijadikan simpul seperti kutipan di atas.



BAB 3 METODOLOGI DAN PERANCANGAN

3.1 Metodologi Penelitian

Metode penelitian menjelaskan langkah-langkah yang akan ditempuh dalam penyusunan skripsi, yaitu perancangan, implementasi dan pengujian dari aplikasi perangkat lunak yang akan dibuat. Secara umum, langkah-langkah penelitian yang dilakukan untuk membuat sistem pakar (SISPAK) adalah :



Gambar 3.1 Flowchart Perancangan Proses

Sumber : Perancangan

3.1.1 Studi Literatur

Dalam mengerjakan skripsi ini, dilakukan studi literatur tentang sistem pakar. Metode *Forward chaining* dan metode *Iterative Dichotomiser 3* (ID3) untuk menentukan kategori pasal yang dilanggar dan maksimal lama hukuman. Studi literatur dilakukan dengan membaca referensi dari buku dan internet. Literatur yang diambil yaitu aplikasi sistem pakar untuk menentukan tindak pidana yang menyangkut dengan kasus pembunuhan atau menghilangkan nyawa seseorang, penelitian tersebut menggunakan metode *Depth First Search (DFS)* yang didapat dari internet dan juga sistem pakar yang menentukan jenis tindak pidana dan pasal-pasal yang berhubungan dengan kasus yang menyangkut harta kekayaan. Sedangkan literatur yang menggunakan metode *Iterative Dichotomiser 3* (ID3) diambil dari penelitian berjudul *Klasifikasi Keluarga Sejahtera Menggunakan Metode Decision Tree Dengan Algoritma Iterative Dichotomiser 3 (ID3) Pada Kecamatan Batu*.

3.1.2 Pengumpulan Data

Dalam suatu sistem pakar dibutuhkan data yang digunakan sebagai parameter *input* maupun dalam memproses data. Dengan menggunakan metode wawancara yaitu peneliti dengan mendatangi pakar atau hakim pengadilan negeri Pare yaitu bapak Bondan Supodo, S.H. Wawancara dilakukan di rumah beliau sesuai perjanjian yang telah kami sepakati dan juga melalui media telepon. Dengan cara tersebut diperoleh data pengetahuan tentang hakim dan bagaimana hakim mengambil sebuah keputusan, informasi yang bisa didapat dari wawancara tersebut yaitu pasal berapa saja yang termasuk ke dalam penganiayaan, apa saja yang dapat menambah pidana atau memberatkan juga apa saja yang dapat meingankan atau bahkan membatalkan hukuman.

Tabel 3.1 Teknik Pengambilan Data Sampel

No.	Data	Jenis Data	Cara Pengambilan
1	Pasal-Pasal Penganiayaan	Pasal	Wawancara dan referensi KUHP
2	Memberatkan Hukuman	Informasi apa saja yang memberatkan hukuman	Wawancara
3	Menghilangkan dan atau mengurangi hukuman	Informasi apa saja yang meringankan hukuman	Wawancara

Sumber: Rencana Kegiatan

3.1.3 Analisis Kebutuhan

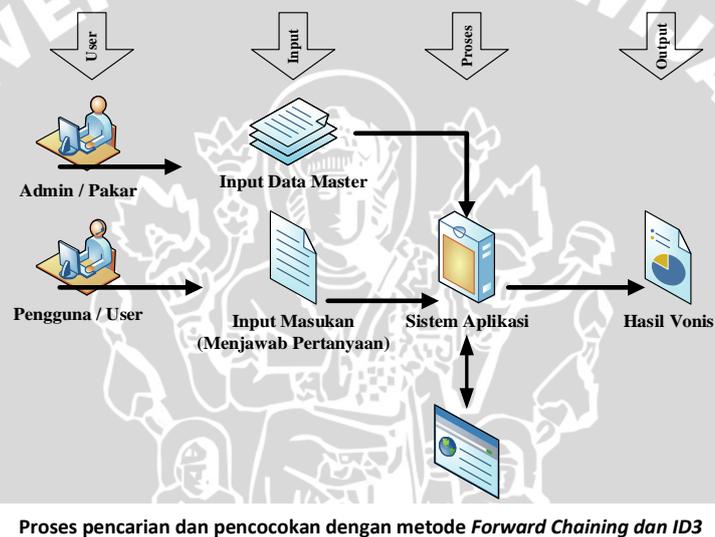
Analisis kebutuhan sistem bertujuan untuk mengidentifikasi apa saja yang dibutuhkan sistem agar tidak menyimpang dari permasalahan dan tujuan penelitian. Analisis kebutuhan yang akan dijabarkan meliputi analisis kebutuhan masukan, analisis kebutuhan proses dan analisis kebutuhan keluaran. Guna

mendefinisikan kebutuhan sistem digunakan bentuk analisa terstruktur, yaitu dengan merancang *Flowchart* sistem, *ERD (Entity Relationship Diagram)*, *Context Diagram* dan *DFD (Data Flow Diagram)*.

3.1.4 Perancangan

Sistem pakar yang akan dibangun digunakan untuk penentuan hukuman dan denda untuk permasalahan tindak pidana penganiayaan. Metode *Forward chaining* dan metode *Iterative Dichotomiser 3 (ID3)* untuk menentukan kategori pasal yang dilanggar dan maksimal lama hukuman dari setiap pertanyaan yang dijawab oleh pengguna. Hasil output sistem terdiri dari : kategori pasal yang dilanggar dan maksimal lama hukuman kurungan.

Perancangan aplikasi sistem dapat dilihat lebih jelas pada perancangan blok diagram Gambar 3.2 di bawah ini :



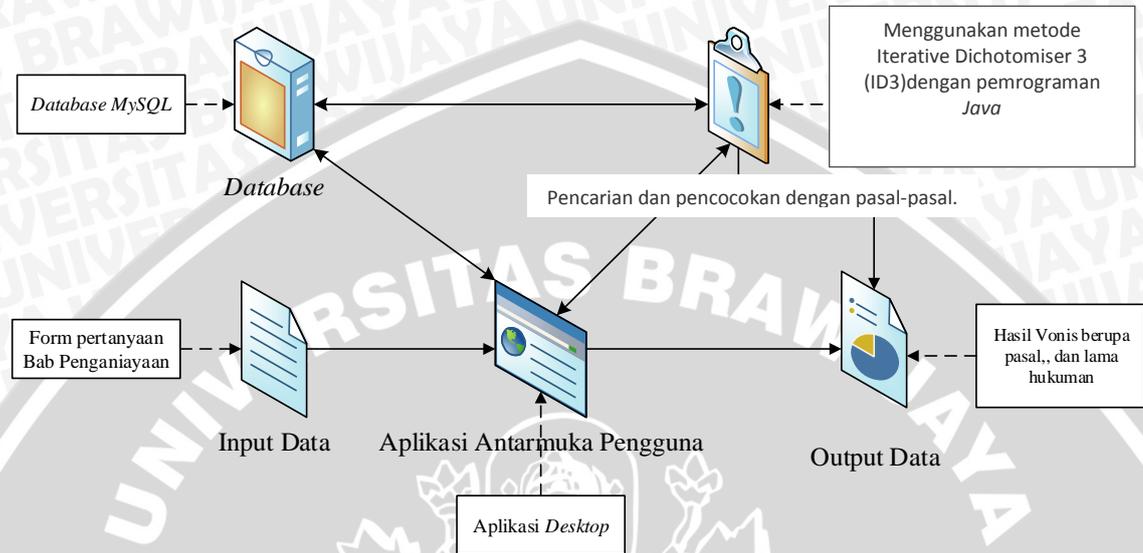
Gambar 3.2 Blok Diagram Perancangan Aplikasi

Sumber : Perancangan

Tahapan yang biasa dilakukan oleh seorang pakar bidang hukum atau seorang hakim dalam melakukan penentuan vonis hukuman adalah dengan melihat kasus-kasus yang dilanggar oleh terdakwa. Semakin spesifik kasus yang dapat diamati, maka semakin besar tingkat keyakinan yang dapat dihasilkan. Sistem menerima masukan berupa keyakinan pengguna terhadap kasus yang dilanggar oleh terdakwa, dengan begitu, semakin besar tingkat keyakinan yang dimasukkan dan semakin spesifik kasus yang dapat diamati maka diharapkan keputusan vonisnya pun dapat mencapai persentase yang semakin tinggi. Hasil akhir berupa keputusan vonis, yaitu pasal apa yang dilanggar dan lama maksimal hukuman.

3.1.5 Implementasi system

Implementasi perangkat lunak dilakukan dengan mengacu kepada perancangan aplikasi. Blok Diagram Implementasi sistem ditunjukkan pada gambar 3.3.

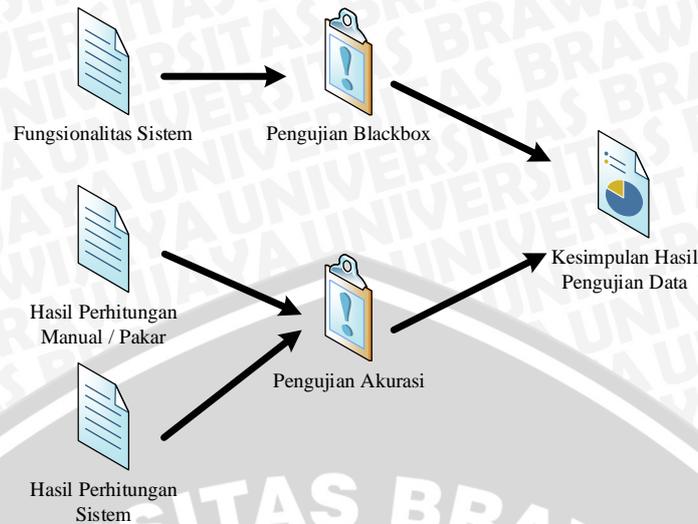


Gambar 3.3 Blok Diagram Implementasi Sistem

Sumber : Perancangan

3.1.6 Pengujian Sistem

Pada tahap ini dilakukan pengujian keberhasilan dan akurasi dari sistem yang telah dibuat pada tahap implementasi sebelumnya. Pengujian dilakukan dengan cara memeriksa apakah sistem sudah berjalan dengan baik dan tidak ada error yang terjadi. Blok diagram pengujian sistem dapat dilihat pada Gambar 3.4.



Gambar 3.4 Blok Diagram Pengujian Sistem

Sumber : Perancangan

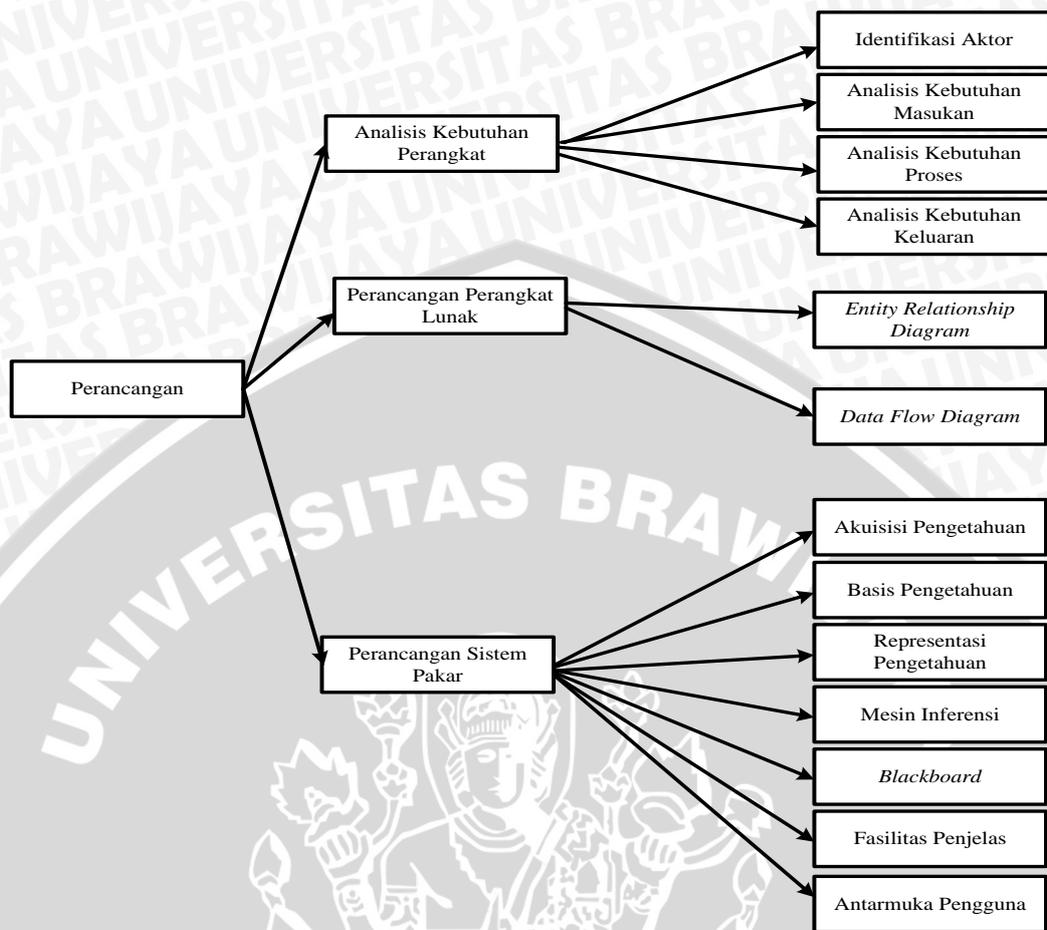
Tahapan pengujian ini dilakukan dengan cara mengambil hasil vonis beberapa kasus penganiayaan yang kemudian diujicobakan dan dibandingkan dengan hasil vonis dari perhitungan pakar. Pengujian *black-box* juga akan diujikan, hal ini sebagai indikator keberhasilan pada setiap fungsionalitas yang ada pada sistem pakar.

3.1.7 Pengambilan Kesimpulan

Kesimpulan dilakukan setelah semua tahapan perancangan, tahapan implementasi dan pengujian metode yang diterapkan telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis metode yang diterapkan. Selanjutnya tahap terakhir dari penulisan adalah saran, dimaksudkan untuk memperbaiki kesalahan yang terjadi serta untuk memberikan pertimbangan atas pengembangan sistem pakar selanjutnya agar lebih baik lagi.

3.2 Perancangan

Perancangan ini dilakukan meliputi tiga tahapan, yaitu proses analisa kebutuhan perangkat, perancangan sistem pakar dan perancangan perangkat lunak. Tahap analisa kebutuhan perangkat terdiri dari identifikasi aktor, analisa kebutuhan masukan, analisa kebutuhan proses dan analisa kebutuhan keluaran. Perancangan sistem pakar terdiri dari perancangan akuisisi pengetahuan, basis pengetahuan, representasi pengetahuan, mesin inferensi, *blackboard*, fasilitas penjelas, dan antarmuka pengguna. Perancangan perangkat lunak terdiri dari membuat *Entity Relationship Diagram*, *Context Diagram*, dan *Data Flow Diagram*. Pohon perancangan sistem pakar dapat dilihat pada **Gambar 3.5** berikut ini :



Gambar 3.5 Pohon Perancangan

Sumber : Perancangan

3.2.1 Analisis Kebutuhan Perangkat

Analisis kebutuhan ini diawali dengan identifikasi aktor-aktor yang terlibat dalam sistem pakar, penjabaran kebutuhan masukan, proses dan keluaran. Analisis kebutuhan ini ditujukan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

Berikut ini adalah kebutuhan yang digunakan dalam pembuatan sistem pakar

:
1. Kebutuhan *Hardware*, meliputi :

- *Processor* : Dengan kecepatan 2.8 Ghz
- Kapasitas *Harddisk* : 40 GB
- *RAM* : 256 MB
- *VGA Card* : 64 MB
- *Monitor*
- *Mouse*

- *Keyboard*
- 2. Kebutuhan *Software*, meliputi :
 - Sistem Operasi Windows 7
 - *NetBeans IDE 8.0*
 - Basisdata MySQL
 - Bahasa Pemrograman java
- 3. Data yang dibutuhkan meliputi :
 - Data pasal KUHP bab penganiayaan
 - Hasil vonis persidangan masalah penganiayaan

3.2.1.1 Identifikasi Aktor

Tahap ini mempunyai tujuan untuk melakukan identifikasi terhadap aktor-aktor yang akan berinteraksi dengan sistem pakar. Pada Tabel 3.1 memperlihatkan dua aktor beserta penjelasannya masing-masing yang merupakan hasil dari proses identifikasi aktor. Gambaran deskripsi aktor dapat dilihat pada Tabel 3.2.

Tabel 3.2 Deskripsi Aktor

Aktor	Deskripsi Aktor
Masyarakat	Aktor yang dapat menggunakan system pakar untuk melihat informasi penentuan hukuman dan denda untuk permasalahan tindak pidana penganiayaan, pengguna tidak perlu melakukan proses login.
Pakar/Hakim	Aktor yang dapat mengakses informasi, login sebagai pakar, dan manipulasi (tambah, ubah, dan hapus) data di dalam sistem.

Sumber : Perancangan

3.2.1.2 Analisa Kebutuhan Masukan

Pakar memberikan masukan berupa :

- Data pasal baru yang belum terdapat dalam sistem. Data pasal meliputi id pasal dan nama pasal.
- Data aturan ditambahkan sesuai dengan kasus dan jenis pasal yang dilanggar.

Dari masukan pakar di atas digunakan sebagai basis pengetahuan dari sistem dalam menentukan hukuman dan denda untuk permasalahan tindak pidana penganiayaan. Selain masukan dari pakar juga terdapat daftar kebutuhan. Daftar kebutuhan ini terdiri dari sebuah kolom yang menguraikan kebutuhan sistem maupun *interface* yang harus disediakan oleh sistem, dan pada kolom yang lain akan menunjukkan nama proses yang akan menunjukkan fungsionalitas masing-masing kebutuhan tersebut. Berikut tabel daftar kebutuhan fungsional keseluruhan sistem. Daftar Kebutuhan Fungsional Sistem dijelaskan pada Tabel 3.3.

Tabel 3.3 Daftar Kebutuhan Fungsional

ID	Requirements	Entitas	Nama Aliran Data
KF_01	Sistem mampu menerima inputan <i>login</i> (pakar)	PK	<i>Login</i> (pakar)
KF_02	Sistem mampu mengelola data pakar	PK	Data Pakar
KF_03	Sistem mampu menerima perubahan data pasal	PK	List Pasal
KF_04	Sistem mampu menerima perubahan data kasus	PK	List Kasus
KF_05	Sistem mampu menerima data kasus yang diinputkan pengguna	P	Data Pengguna
KF_06	Sistem mampu mengelola data pengguna	P	Proses Vonis
KF_07	Sistem mampu menampilkan hasil vonis berdasarkan kasus yang diinputkan pengguna	P	Lihat Hasil Vonis
KF_08	Sistem mampu <i>logout</i>	PK	<i>Logout</i>

Sumber : Perancangan

3.2.1.3 Analisa Kebutuhan Proses

Proses inti dari sistem ini adalah proses penalaran. Sistem akan melakukan penalaran untuk penentuan hukuman dan denda untuk permasalahan tindak pidana penganiayaan berdasarkan kasus yang dimasukkan oleh pengguna. Pada sistem telah disediakan aturan basis pengetahuan untuk penelusuran pasal.

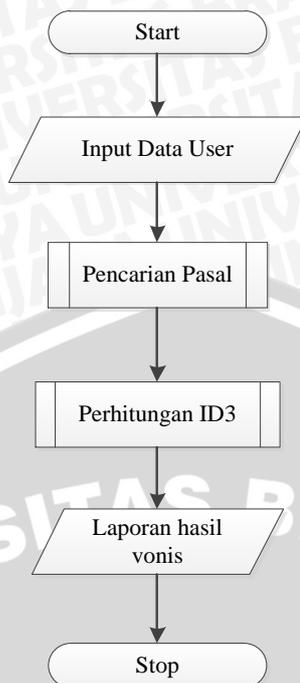
3.2.1.4 Analisa Kebutuhan Keluaran

Data keluaran dari sistem ini adalah hasil vonis menggunakan *Forward chaining* dan metode *Iterative Dichotomiser 3 (ID3)*. Hasil vonis tersebut berdasarkan fakta kasus yang berasal dari masukan pengguna. Hasil *output* sistem terdiri dari : pasal yang dilanggar, lama hukuman dan pidana denda jika ada.

3.2.2 Perancangan Perangkat Lunak

3.2.2.1 Flowchart Aplikasi

Diagram alir atau *Flowchart* merupakan visualisasi dari algoritma yang diterapkan untuk memecahkan persoalan dalam sistem pakar. Berikut merupakan *Flowchart* konsultasi sistem pakar ditunjukkan pada gambar 3.6.

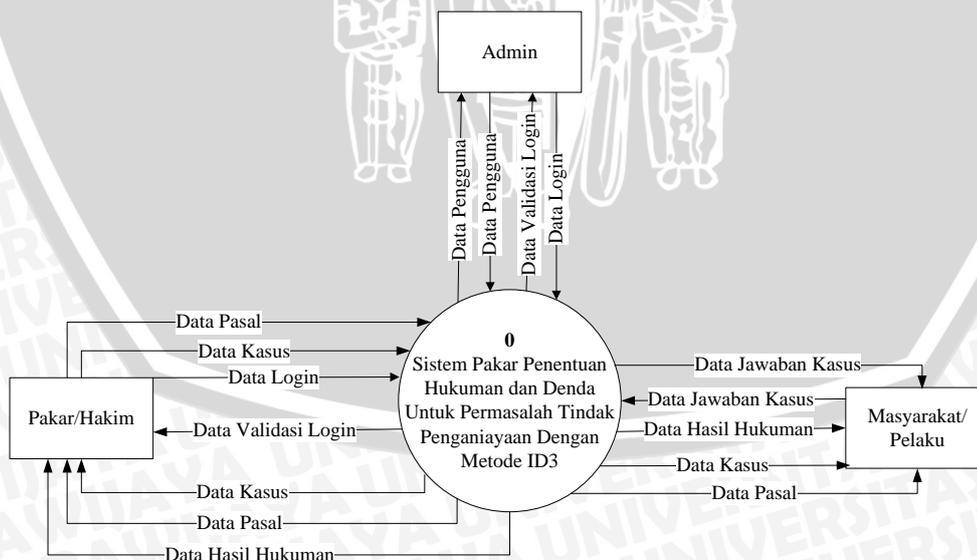


Gambar 3.6 Flowchart Konsultasi Sistem Pakar

Sumber : Perancangan

3.2.2.2 Perancangan Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah diagram alir yang dipresentasikan dalam bentuk lambang-lambang tertentu yang menunjukkan aliran data, proses, tempat penyimpanan data, dan entitas eksternal. DFD untuk sistem pakar penentuan hukuman dan denda untuk permasalahan tindak pidana penganiayaan ini dimulai dari DFD level 0 sampai level 1 dapat dilihat pada gambar 3.7



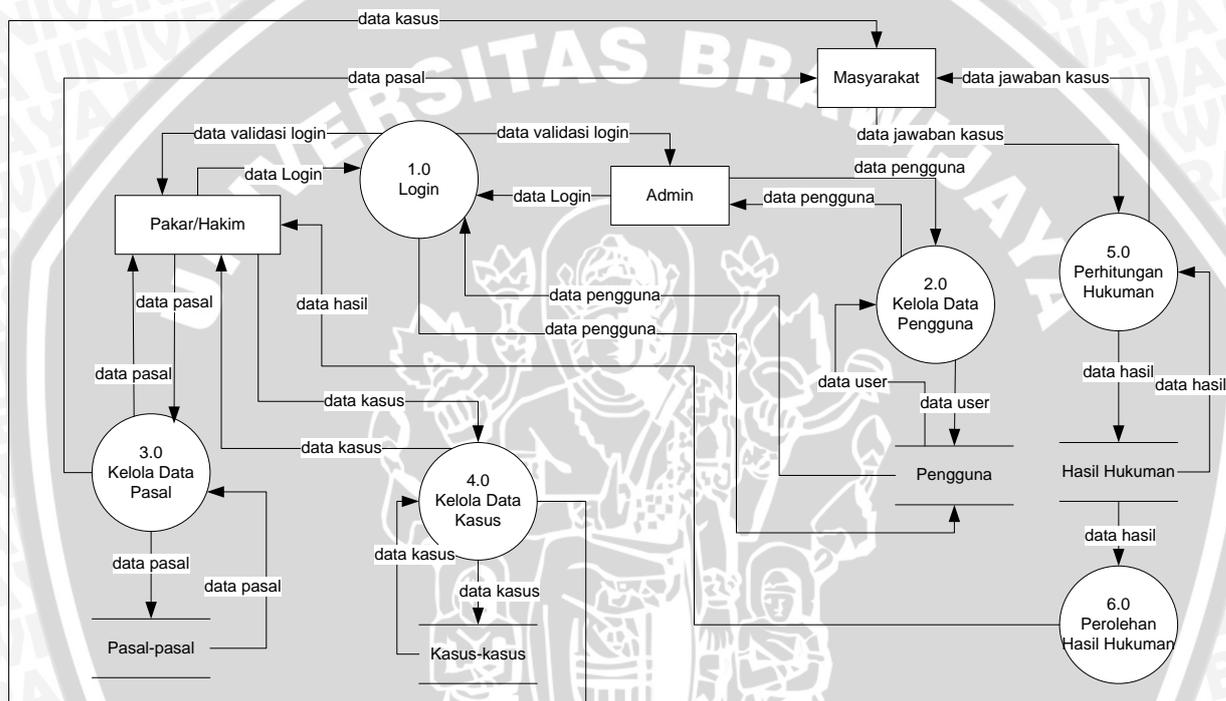
Gambar 3.7 Context Diagram

Sumber : Perancangan

Diagram konteks merupakan gambaran secara umum mengenai sebuah sistem yang dirancang secara global, yaitu suatu diagram yang mempersentasikan atau menggambarkan hubungan antara sistem dengan lingkungan luar sistem yang mempengaruhi operasi sistem. Sistem ditunjukkan dalam satu lingkungan yang menggambarkan keseluruhan proses dalam sistem dan hubungannya dengan entitas. Terdapat dua entitas yang terhubung langsung dengan sistem yaitu pengguna (*user*) dan pakar (*admin*).

- **DFD Level 0**

DFD Level 0 memiliki enam proses utama dengan tiga entitas yaitu Pakar/hakim, admin dan masyarakat (*user*), seperti pada gambar 3.8 berikut.



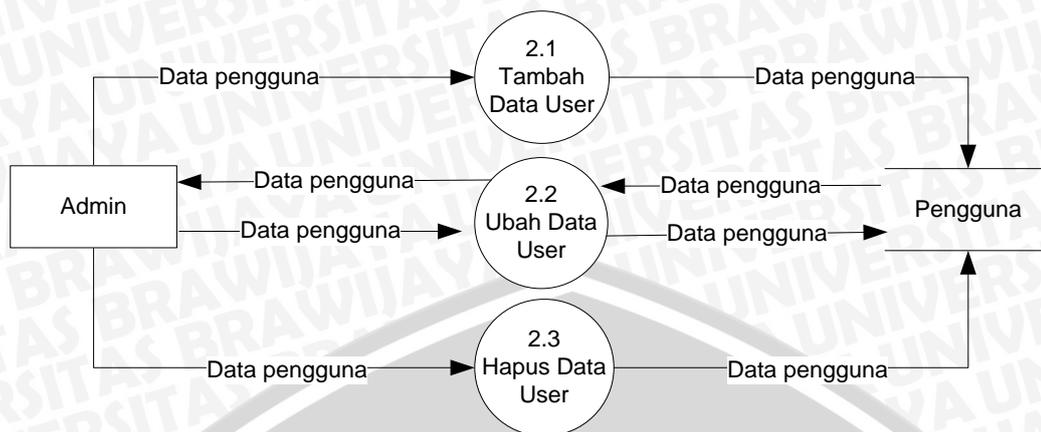
Gambar 3.8 DFD Level 0

Sumber : Perancangan

- **DFD Level 1 proses 2.0**

DFD Level 1 proses 2.0, menggambarkan proses kelola data pengguna meliputi tambah data user, ubah data user dan hapus data user, seperti terlihat pada gambar 3.9 berikut.



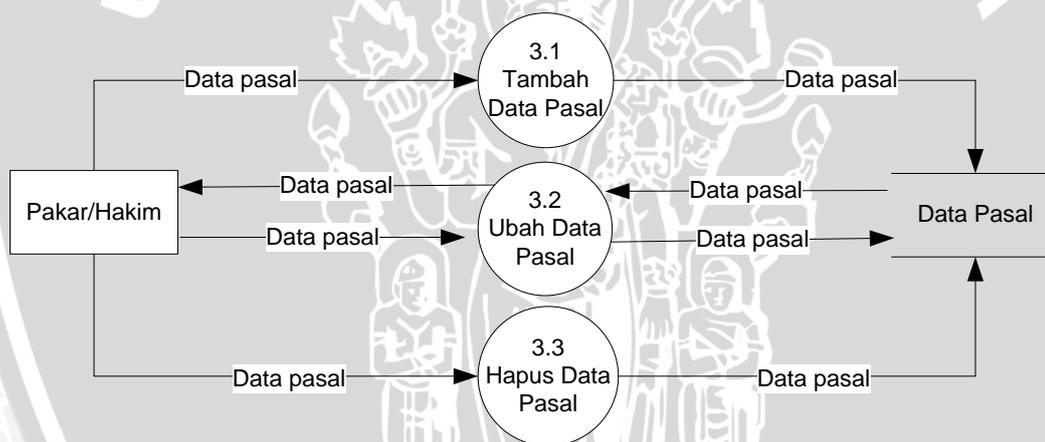


Gambar 3.9 DFD Level 1 Proses 2.0

Sumber : Perancangan

- **DFD Level 1 proses 3.0**

DFD Level 1 proses 3.0, menggambarkan proses kelola data pasal. Dalam mengelola data pasal meliputi, tambah data pasal, ubah data pasal dan hapus data pasal. Untuk lebih jelas, dapat dilihat pada gambar berikut 3.10.



Gambar 3.10 DFD Level 1 Proses 3.0

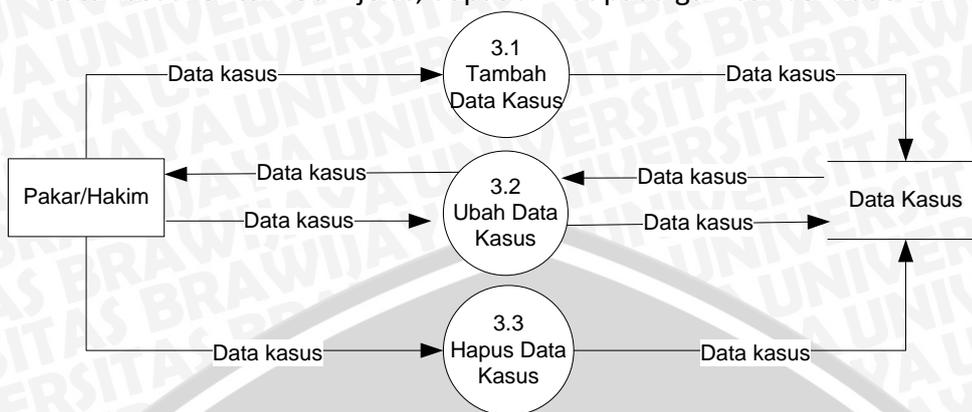
Sumber : Perancangan

- **DFD Level 1 proses 4.0**

DFD Level 1 proses 4.0, menggambarkan proses kelola data kasus. Dalam mengelola data pasal meliputi, tambah data kasus, ubah data kasus dan hapus



data kasus. Untuk lebih jelas, dapat dilihat pada gambar berikut 3.11.

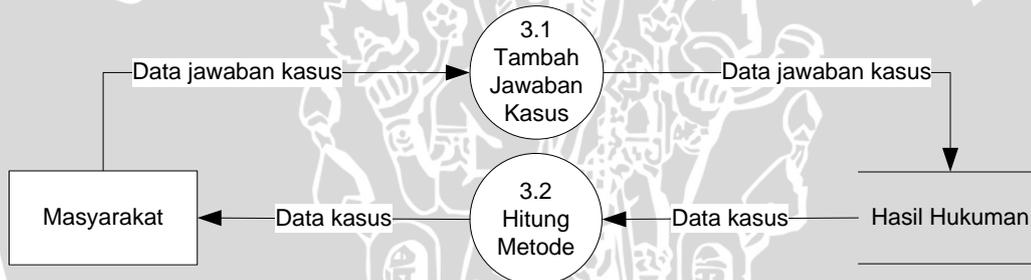


Gambar 3.11 DFD Level 1 Proses 4.0

Sumber : Perancangan

• **DFD Level 1 proses 5.0**

DFD Level 1 proses 5.0, menggambarkan proses perhitungan hukuman. Dalam perhitungan hukuman meliputi, tambah jawaban kasus dan hitung metode. Untuk lebih jelas, dapat dilihat pada gambar berikut 3.12.



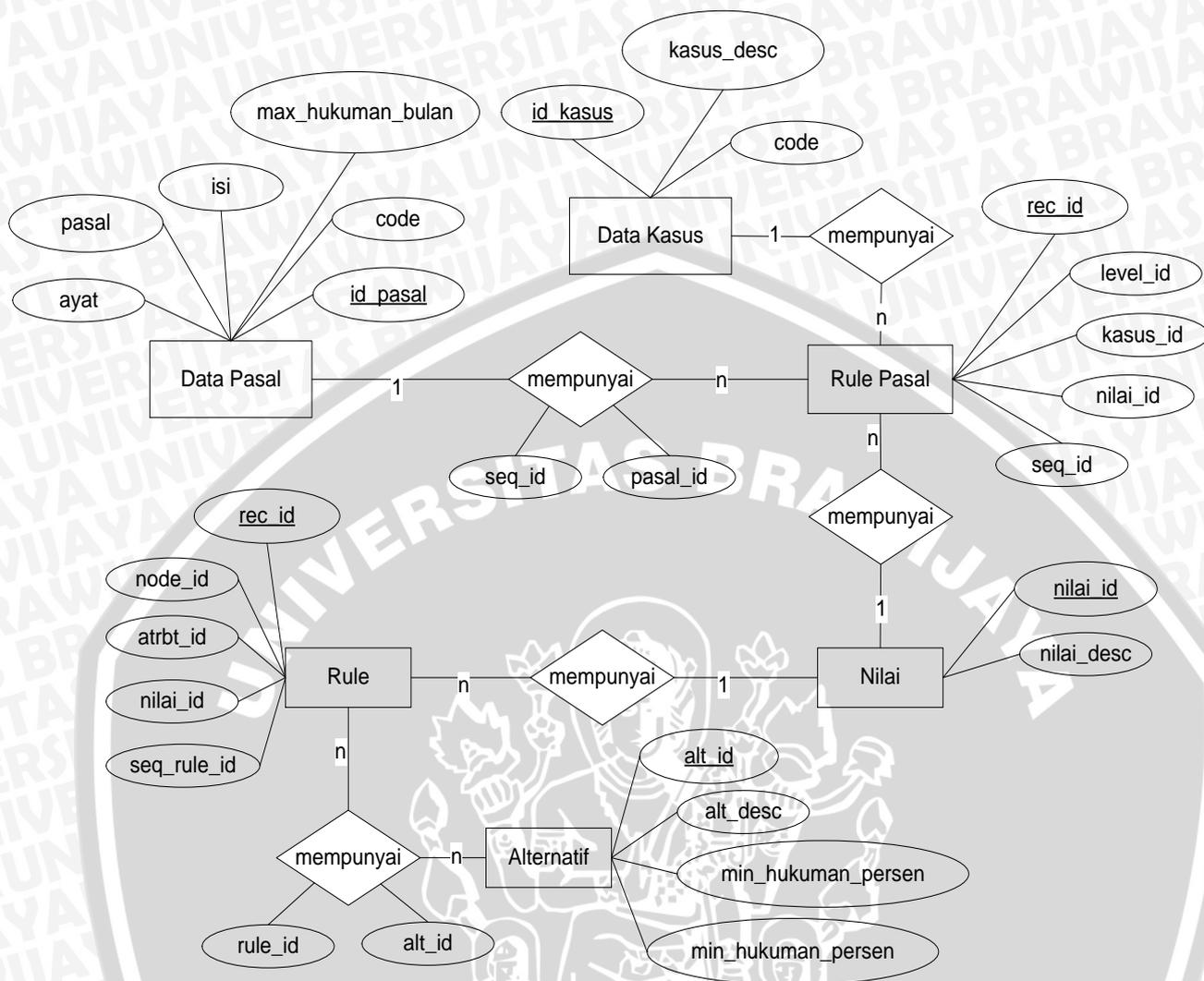
Gambar 3.12 DFD Level 1 Proses 5.0

Sumber : Perancangan

3.2.2.3 Perancangan Entity Relational Diagram (ERD)

Memperhatikan data serta informasi yang akan digunakan dalam proses pembangunan aplikasi ini, maka dibangun sebuah desain basis data dengan menggunakan *tools Entity Relational Diagram (ERD)* pada gambar 3.13 berikut:





Gambar 3.13 Entity Relational Diagram (ERD)

Sumber : Perancangan

Tabel Data Pasal

No	Nama Field	Type	Lebar	Keterangan
1	id_pasal	INT	11	ID untuk penomoran Pasal (Primary Key)
2	code	VARCHAR	10	Pengkodean Pasal
3	isi	VARCHAR	500	Isi dari Pasal
4	pasal	INT	11	Nomor Pasal sesuai dengan undang-undang

5	ayat	VARCHAR	10	Nomor ayat sesuai dengan undang-undang
6	max_hukuman_bulan	FLOAT		Maksimal hukuman yang akan dikenakan

Tabel Data Kasus

No	Nama Field	Tipe	Lebar	Keterangan
1	id_kasus	INT	11	ID untuk penomoran Kasus (<i>Primary Key</i>)
2	code	VARCHAR	10	Pengkodean Kasus
3	kasus_desc	VARCHAR	255	Deskripsi dari masing-masing Kasus

Tabel Rule Pasal

No	Nama Field	Tipe	Lebar	Keterangan
1	rec_id	INT	11	ID untuk penomoran row pada tabel (<i>Primary Key</i>)
2	level_id	INT	11	Penomoran level pada tree
3	kasus_id	INT	11	Deskripsi dari masing-masing Kasus (<i>Foreign Key</i>)
4	nilai_id	INT	11	Id dari nilai (<i>Foreign Key</i>)
5	seq_id	INT	11	Nomor untuk pengelompokan Rule

Tabel Nilai

No	Nama Field	Tipe	Lebar	Keterangan
1	nilai_id	INT	11	ID untuk master nilai (<i>Primary Key</i>)
2	nilai_desc	VARCHAR	50	Penjelasan nilai

Tabel Rule

No	Nama Field	Tipe	Lebar	Keterangan
1	rec_id	INT	11	ID untuk penomoran row pada tabel (<i>Primary Key</i>)
2	node_id	INT	11	Penomoran level pada tree
3	atrbt_id	INT	11	Id untuk atribut (<i>Foreign Key</i>)
4	nilai_id	INT	11	Id dari nilai (<i>Foreign Key</i>)
5	seq_Rule_id	INT	11	Nomor untuk pengelompokan Rule

Tabel Alternatif

No	Nama Field	Tipe	Lebar	Keterangan
1	alt_id	INT	11	ID untuk alternatif (<i>Primary Key</i>)
2	alt_desc	VARCHAR	70	Penjelasan untuk alternatif
3	min_hukuman_persen	INT	11	Minimal lamanya hukuman dalam bentuk persentase
4	max_hukuman_persen	INT	11	Maksimal lamanya hukuman dalam bentuk persentase

3.2.3 Perancangan Sistem Pakar

Sistem pakar ini digunakan untuk penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan. Metode *Forward chaining* di sini digunakan untuk proses pencarian dan penentuan pasal, sedangkan metode *Iterative Dichotomiser 3* (ID3) digunakan untuk menentukan lama hukuman.

Tahapan yang dilakukan yaitu sistem menerima masukan berupa pernyataan pengguna terhadap kasus yang ditanyakan oleh sistem, semakin spesifik kasus yang dimasukkan oleh pengguna maka diharapkan keputusan vonis pun dapat mencapai persentase yang semakin tinggi. Hasil akhir berupa pasal yang dilanggar, lama maksimal hukuman.

3.2.3.1 Akuisisi Pengetahuan

Akuisisi pengetahuan merupakan suatu proses untuk mengumpulkan data-data pengetahuan akan suatu masalah dari pakar. Bahan pengetahuan dapat ditempuh dengan beberapa cara, misalnya mendapatkan dari buku. Sumber pengetahuan tersebut harus diperoleh dengan kemampuan untuk mengolah data-data yang tersedia menjadi solusi yang efisien, komunikasi yang baik dan kerjasama tim yang solid. Karena semua kemampuan menjadi nilai mutlak yang diperlukan bagi pengembang sistem. Pada penelitian ini metode yang dipakai peneliti yaitu metode wawancara dengan pakar.

Metode yang digunakan dalam akuisisi pengetahuan, yaitu :

1. Wawancara

Tujuan dari wawancara ini adalah memperoleh wawasan dari pakar untuk domain masalah tertentu. Pada wawancara ini mengumpulkan semua informasi tentang hakim dan bagaimana hakim mengambil sebuah keputusan, informasi yang bisa didapat dari wawancara tersebut yaitu pasal berapa saja yang termasuk ke dalam penganiayaan, apa saja yang dapat menambah pidana atau memberatkan juga apa saja yang dapat meingankan atau bahkan membatalkan hukuman.

3.2.3.2 Basis Pengetahuan

Dalam metode *Forward chaining* digunakan untuk pengambilan data sebagai pengetahuan yang dibutuhkan terutama pada kasus-kasus dalam menentukan jenis pasal yang dilanggar, sedangkan pada 11 parameter dasar yang diberikan pakar akan dijadikan sebagai bahan perhitungan metode metode *Iterative Dichotomiser 3 (ID3)* untuk menentukan lama hukuman.

3.2.3.3 Representasi Pengetahuan

Pengetahuan yang telah diuraikan, akan direpresentasikan kedalam aturan yang menghasilkan solusi atau jenis pasal dari tiap kasus yang dilanggar. Untuk memprediksi jenis pasal yang dilanggar oleh tersangka maka setiap kasus yang ada butuh dianalisa dan setelah diketahui kasus apa saja yang mempengaruhi pasal tertentu kemudian dibuatlah aturan (*Rule*)-nya. Tabel *Rule* dapat dilihat pada tabel 3.4.

Tabel 3.4 Rule

No	Kasus							Pasal
1	K003=Ya							352A2 atau 351A5
2	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Ya	K007=Ya/K008=Ya/K009=Ya	K011=Ya	355A1 jo 356
3	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Ya	K007=Ya/K008=Ya/K009=Ya	K012=Ya	355A2 jo 356
4	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Ya	K007=Tidak/K008=Tidak/K009=Tidak	K011=Ya	355A1
5	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Ya	K007=Tidak/K008=Tidak/K009=Tidak	K012=Ya	355A2
6	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K010=Ya	353A1 jo 356
7	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K011=Ya	353A2 jo 356
8	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K012=Ya	353A3 jo 356
9	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K010=Ya	353A1
10	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K011=Ya	353A2
11	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K012=Ya	353A3
12	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Ya	K007=Ya/K008=Ya/K009=Ya	K011=Ya	354A1 jo 356
13	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Ya	K007=Ya/K008=Ya/K009=Ya	K012=Ya	354A2 jo 356
14	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Ya	K007=Tidak/K008=Tidak/K009=Tidak	K011=Ya	354A1
15	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Ya	K007=Tidak/K008=Tidak/K009=Tidak	K012=Ya	354A2
16	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K010=Ya	351A1 jo 356
17	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K011=Ya	351A2 jo 356
18	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K012=Ya	351A3 jo 356
19	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K013=Ya	351A4 jo 356
20	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K014=Ya	352A1 jo 356
21	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K010=Ya	351A1
22	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K011=Ya	351A2
23	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K012=Ya	351A3

(lanjutan)

No	Kasus	Pasal	No	Kasus	Pasal	No	Kasus	Pasal
24	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K013=Ya	351A4
25	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K014=Ya	352A1
26	K001=Ya	K002=Ya	K004=Tidak			K007=Ya/K008=Ya/K009=Ya	K011=Ya	358-1 jo 356
27	K001=Ya	K002=Ya	K004=Tidak			K007=Ya/K008=Ya/K009=Ya	K012=Ya	358-2 jo 356
28	K001=Ya	K002=Ya	K004=Tidak			K007=Tidak/K008=Tidak/K009=Tidak	K011=Ya	358-1
29	K001=Ya	K002=Ya	K004=Tidak			K007=Tidak/K008=Tidak/K009=Tidak	K012=Ya	358-2

keterangan:

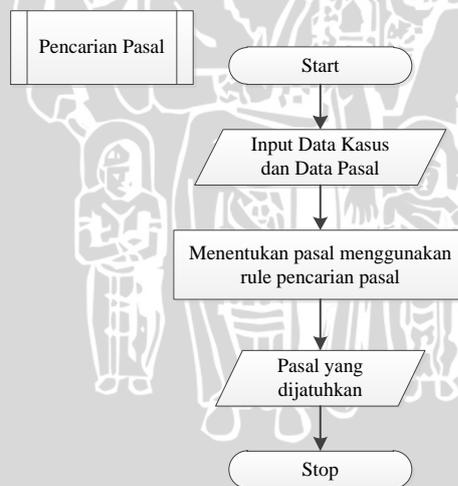
No	Kasus	Keterangan
1	K001	Apakah anda melakukan penganiayaan?
2	K002	Apakah anda melakukan penganiayaan di Indonesia?
3	K003	Apakah anda hanya sekedar melakukan percobaan penganiayaan?
4	K004	Apakah penganiayaan tersebut anda lakukan secara individu?
5	K005	Apakah penganiayaan tersebut anda lakukan secara berencana?
6	K006	Apakah ada unsur kesengajaan dalam penganiayaan yang anda lakukan?
7	K007	Apakah anda memiliki hubungan kekeluargaan dengan korban?
8	K008	Apakah korban merupakan anggota pejabat?
9	K009	Apakah anda menggunakan racun dalam penganiayaan tersebut?
10	K010	Apakah penganiayaan tersebut mengakibatkan luka ringan?
11	K011	Apakah penganiayaan tersebut mengakibatkan luka berat?
12	K012	Apakah penganiayaan tersebut mengakibatkan kematian?
13	K013	Apakah penganiayaan tersebut merusak kesehatan?
14	K014	Apakah penganiayaan tersebut mengakibatkan penyakit?

Sumber: Perancangan

Dari uraian metode pemecahan diatas dianggap belum bisa memecahkan ketidakpastian vonis pasal. Dalam kasus penentuan pasal, metode *Rule Base* dalam penentuan pasal hanya beracuan dari *Rule* yang telah dibuat, jadi apabila ada satu kondisi yang terdapat dalam sebuah *Rule* tidak terpenuhi, maka *Rule* tersebut tidak akan dilakukan proses atau tidak diikuti dalam pengambilan keputusan hasil vonis. Jika semua kondisi dalam tersebut semua memenuhi dari data yang di cocokkan maka akan diikuti proses pengambilan keputusan. Sedangkan untuk penentuan lama hukuman dalam kasus penentuan pasal dan lama hukuman yaitu parameter yang diberikan oleh pakar akan dihitung menggunakan metode *forward chaining* dan *Iterative Dichotomiser 3 (ID3)* yang telah diinputkan data latih berdasarkan kasus-kasus penganiayaan sehingga dapat menghasilkan keputusan berupa berapa lama hukuman apakah ringan, sedang ataupun lama berdasarkan kasus yang dilanggar dan pasal yang telah diputuskan.

3.2.3.4 Analisa Proses

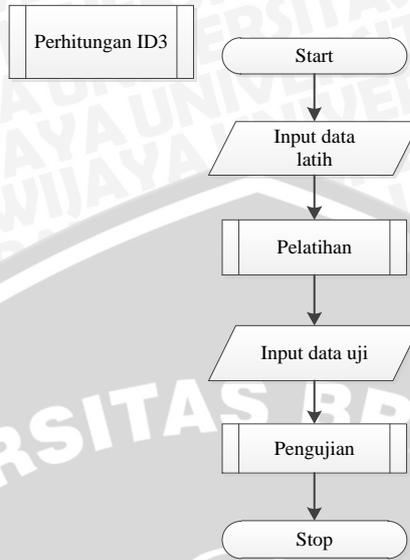
Pada bagian ini dijelaskan rancangan proses serta gambaran umum bagaimana proses kerja sistem melakukan pencarian pasal, kemudian membentuk suatu *Decision Tree* dan pengklasifikasian menggunakan algoritma ID3 guna menentukan lamanya hukuman. Inputan sistem dapat menggunakan 2 macam jenis data, yakni text dan excel, sedangkan untuk hasil dari sistem ini menggunakan database untuk penyimpanan datanya. Proses pencarian pasal dapat dijelaskan menggunakan *Flowchart* seperti pada gambar 3.14 berikut.



Gambar 3.14 Flowchart Sistem Pencarian Pasal

Setelah pasal ditentukan, selanjutnya menentukan lamanya hukuman menggunakan metode ID3. Proses pertama pada perhitungan ID3 nantinya ialah pembacaan data training yang bisa diinputkan menggunakan text atau excel, setelah data training masuk maka dilanjutkan dengan memasukkan data uji. Hasil dari pemrosesan data uji akan disimpan didatabase. *Decision Tree* yang terbentuk dari proses pengolahan data kemudian akan diekstrak *Rulanya*. *Rule* yang didapatkan dari *Decision Tree* yang terbentuk kemudian diterapkan ke data untuk mengklasifikasikan status TKS secara otomatis. Penjelasan gambaran umum

proses yang dimulai dari pembacaan data hingga proses akhir pada perhitungan ID3 akan digambarkan pada gambar 3.2 .

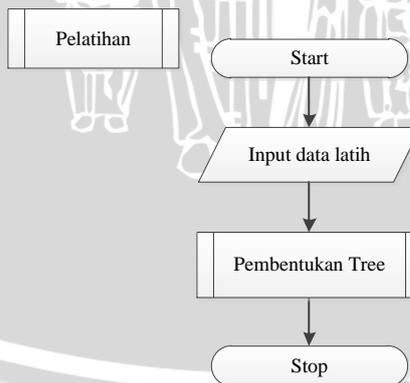


Gambar 3.15 Flowchart Sistem Pada Perhitungan ID3

Gambaran umum proses-proses utama yang akan dilakukan oleh sistem dalam penelitian ini dijelaskan pada Gambar 3.15. Detail dari masing-masing proses pada Gambar 3.15 akan dijelaskan lebih lanjut pada sub bab ini mulai dari proses perhitungan *Entropy* hingga proses mendapatkan *Rule*.

3.2.3.5 Pelatihan

Proses pelatihan ini adalah proses dimana *tree* nantinya dibentuk dari inputan data latih yang dimasukkan user. *Tree* nantinya digunakan untuk proses klasifikasi pada tahap pengujian. *Flowchart* proses pelatihan ditunjukkan pada gambar 3.16.



Gambar 3.16 Flowchart Pelatihan

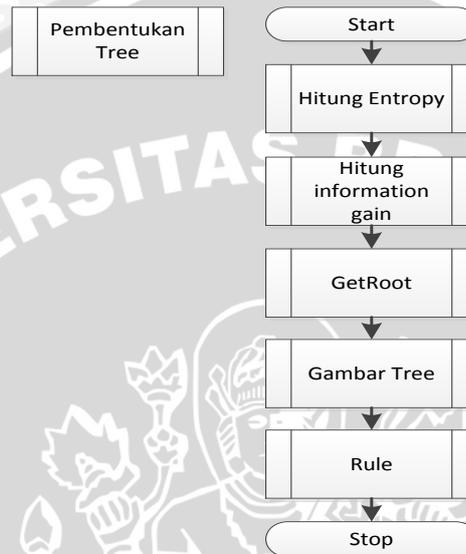
3.2.3.6 Pembentukan *Tree*

Tahap pembentukan *tree* dibagi menjadi 4 proses utama, yakni menghitung *Entropy*, menghitung nilai *Information Gain*, lalu *GetRoot* yakni proses pemilihan suatu atribut menjadi *node*, dan yang terakhir Gambar *Tree* dimana



node yang terpilih dari *GetRoot* dibentuk menjadi *tree*. Setelah *tree* terbentuk kemudian mencari *Rule* untuk digunakan pada data latih.

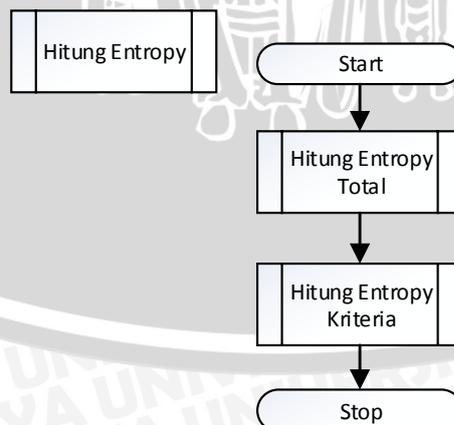
Penanganan missing value didasarkan dari penelitian yang telah dilakukan [BHA-09]. Penerapan yang dilakukan pada penelitian ini kelas ya (v) diganti dengan nilai 1 dan kelas tidak (x) diganti dengan nilai 0. Penginisialisasian nilai dengan angka 1 dan 0 pada tiap kelas ini dilakukan hanya agar memudahkan dalam penerapan proses perulangan pada aplikasi. *Flowchart* proses pembentukan *tree* ditunjukkan pada gambar 3.17



Gambar 3.17 *Flowchart* Pembentukan *Tree*

3.2.3.7 Hitung Entropy

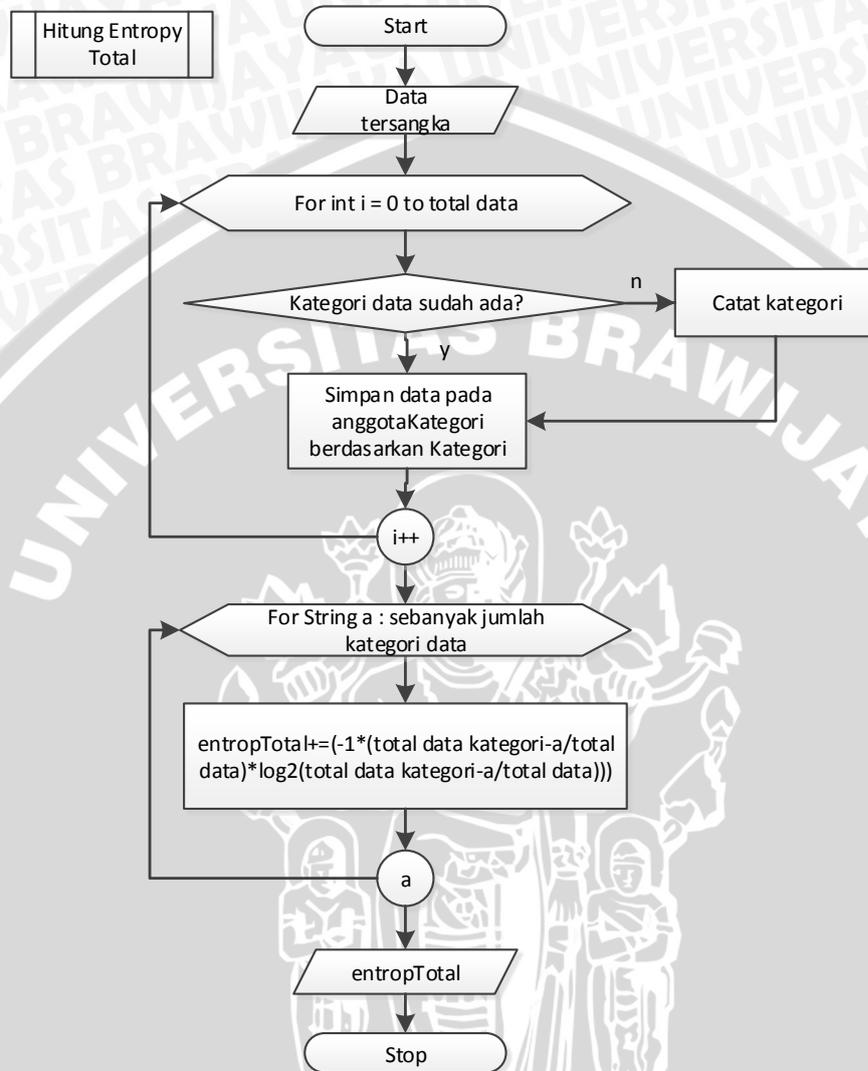
Proses Hitung *Entropy* terdiri dari 2 macam perhitungan, yakni Hitung *Entropy* Total dan Hitung *Entropy* Kriteria seperti yang ditunjukkan pada gambar 3.18.



Gambar 3.18 *Flowchart* Hitung *Entropy*

Flowchart perhitungan *Entropy* Total dan *Entropy* Kriteria ditunjukkan pada gambar 3.19 dan 3.20. Proses perhitungan ini dilakukan untuk mendapatkan

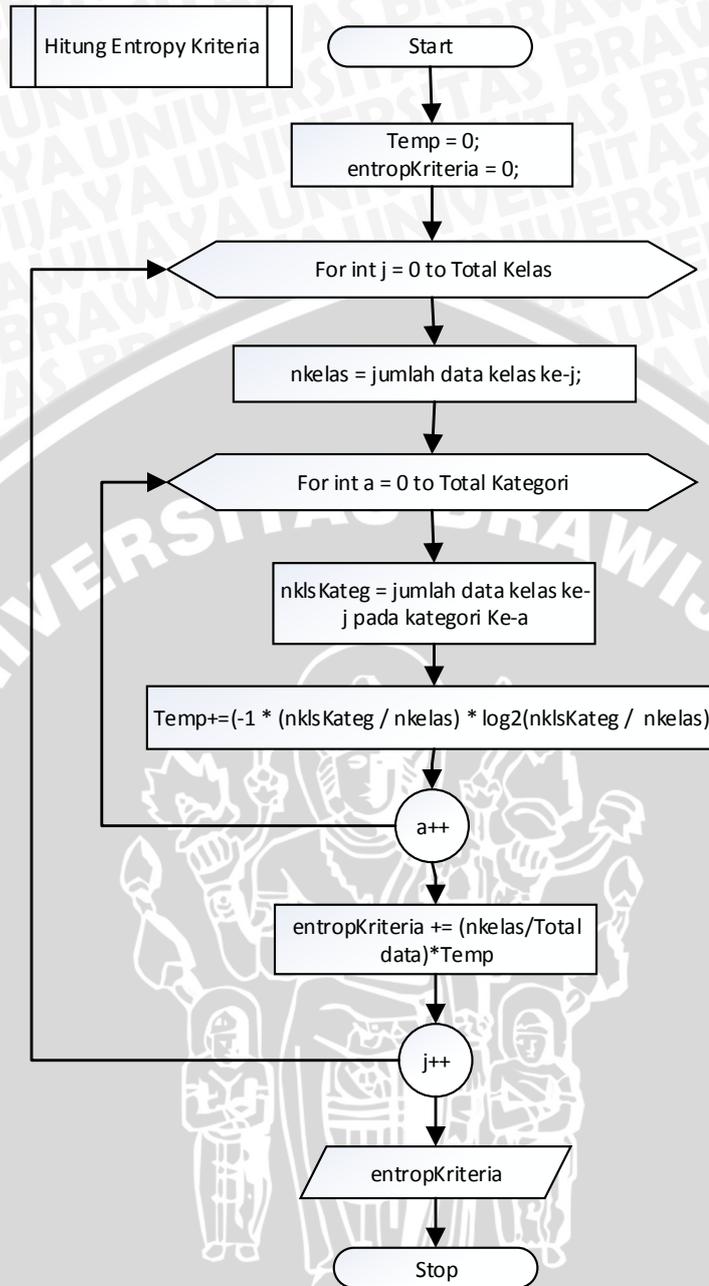
nilai *Entropy* yang nantinya digunakan untuk mencari nilai *Information Gain* pada masing-masing kriteria. Terdapat 2 langkah perhitungan *Entropy* yaitu perhitungan *Entropy* total yang melibatkan semua data dan perhitungan *Entropy* pada kriteria tertentu.



Gambar 3.19 Flowchart Perhitungan *Entropy* total

Pada proses perhitungan *Entropy* total ini diawali dengan mencari jumlah data pada tiap-tiap kategori. Caranya yaitu pertama melakukan pemetaan kategori terhadap semua data. Kategori yang belum ada sebelumnya akan disimpan sebagai kategori yang baru. Kemudian dilakukan pencarian banyaknya data pada masing-masing kategori. Langkah terkakhir dilakukan perhitungan menggunakan rumus *Entropy* terhadap masing-masing kategori. Nilai *Entropy* tiap kategori data tersebut kemudian dijumlahkan dan hasilnya disimpan dalam variabel *entropTotal*.

Proses kedua yaitu menghitung *Entropy* dalam kelas ke-*i* hingga *n*, *n* adalah banyaknya kelas. Nilai kelas yang dimaksud yaitu +1 dan 0. Caranya yaitu mengalikan frekuensi data dengan nilai *Entropy* pada kelas +1 dan 0.



Gambar 3.20 Flowchart Perhitungan Entropy Nilai dalam Atribut

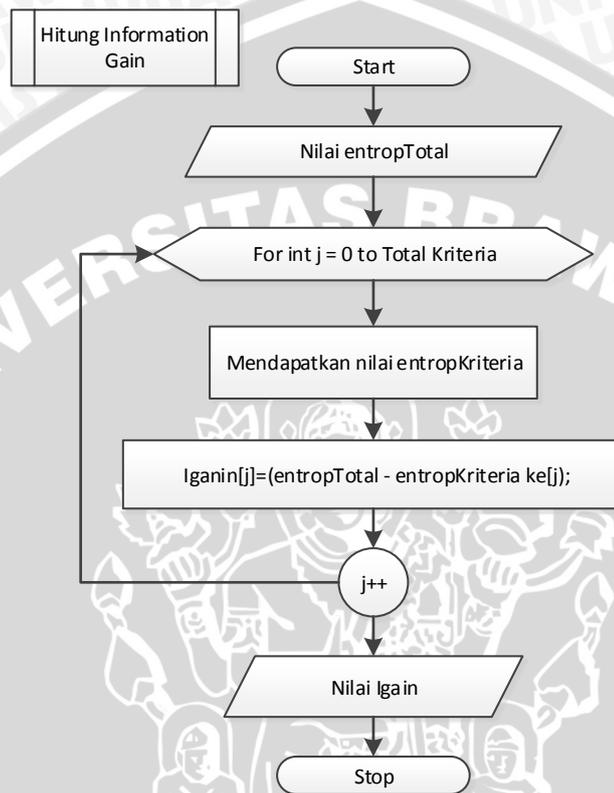
Langkah pertama yaitu mencari jumlah data pada tiap kelas semua kategori yang disimpan dalam variabel *nkelas*. Kemudian mencari jumlah data tiap kelas pada kategori tertentu yang disimpan dalam variabel *nklsKateg*. Nilai pada *nklsKateg* dibagi dengan nilai *nkelas* untuk mencari frekuensi atau probabilitas munculnya data dengan kelas ke-*j* terhadap kategori ke-*a*. Didapatkan nilai *Entropy* tiap kategori menggunakan rumus *Entropy* (2.1). Nilai *Entropy* dari tiap kategori pada kelas ke-*j* kemudian dijumlahkan, hasilnya dikalikan dengan frekuensi atau probabilitas munculnya data kelas ke-*j* terhadap data sampel. Hasil perkalian tersebut kemudian dijumlahkan sehingga didapat *Entropy* kriteria yang

repository.ub.ac.id

disimpan dalam variable *entropKriteria*. Proses tersebut diulang sebanyak jumlah kriteria sehingga didapat *Entropy* masing-masing kriteria.

3.2.3.8 Perhitungan *Information Gain*

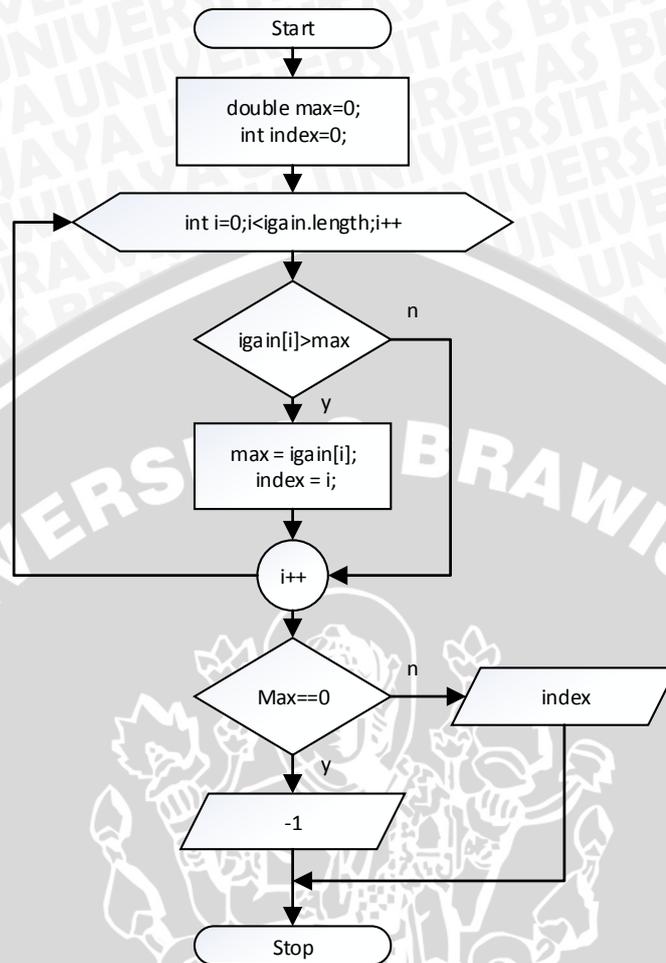
Proses selanjutnya yaitu menghitung nilai *Information Gain* terhadap kriteria yang digunakan. Nilai *Information Gain* ini nantinya akan digunakan untuk menentukan *root* atau *node* dari *tree* yang akan dibuat.



Gambar 3.21 Flowchart Perhitungan *Entropy* Atribut

Pertama, nilai *Entropy* total yang disimpan dalam variabel *entropTotal* yang telah didapat pada proses sebelumnya dijadikan sebagai inputan untuk menghitung nilai *Information Gain*. Kemudian mendapatkan nilai *Entropy* kriteria ke-*j* yang telah diperoleh dari perhitungan sebelumnya. Nilai *Information Gain* kriteria ke-*j* diperoleh dari *Entropy* total dikurangi nilai *Entropy* kriteria ke-*j*.

3.2.3.9 GetRoot



Gambar 3.22 Flowchart GetRoot

Method GetRoot ini digunakan untuk mengambil nilai *Information Gain* tertinggi. Di inialisasikan variabel max dan index dengan nilai awal nol, kemudian dilakukan pengecekan apakah igain pada index i lebih dari max, jika ya maka nilai max diganti dengan nilai igain dan index di isi dengan nilai index dari igain tersebut, pengecekan dilakukan sebanyak jumlah igain. Jika nilai max adalah nol maka mengembalikan nilai -1 yang artinya tidak ada index dari igain yang dipilih sehingga nilai max tetap nol, sedangkan jika nilai max adalah selain nol maka nilai yang dikembalikan dari method ini adalah index ke-i yang artinya igain ke-i memiliki nilai paling besar. Sehingga kriteria yang memiliki igain paling besar tersebut yang terpilih sebagai *node*.

3.2.3.10 Gambar Tree

Gambar Tree adalah proses dimana tree mulai dibentuk. Mengambil nilai *Information Gain* yang telah diproses sebelumnya dan menggunakan method GetRoot didapat kriteria yang terpilih sebagai *node*. *Node* tersebut disimpan didalam Stack. *Node* yang ada dalam stack dikeluarkan, dan dicek apakah *node* tersebut adalah *leaf*, jika tidak maka berlanjut ke proses selanjutnya. Membuat

variable databagi, databagi ini adalah data yang akan diproses, untuk membentuk *tree*. Databagi di cek apakah jumlahnya lebih dari nol, jika ya maka databagi di cek lagi menggunakan method apakahSamaKelas, jika nilai dari apakahSamaKelas kurang dari sama dengan 1 maka *Node* tersebut diset sebagai *leaf*. Jika nilai dari apakahSamaKelas lebih dari 1 maka menghitung nilai *Information Gain* tertinggi lagi dari databagi dan mencari kriteria yang terpilih dengan *Information Gain* tertinggi menggunakan *GetRoot* dan di simpan dalam variabel a. Jika a lebih dari 0 maka *node* tersebut dimasukkan ke dalam stack jalanCrita. Jika a sama dengan 0 berarti dalam perhitungan tersebut tidak ada nilai *Information Gain* yang lebih dari 0 namun masih terdapat lebih dari satu class. Jika nilai i adalah 1 maka *node* tersebut ditaruh pada anak kanan, atau anak yang bernilai ya. Jika i adalah 0 maka *node* tersebut ditaruh pada anak kiri, atau anak yang bernilai tidak. *Flowchart* proses.

3.2.3.11 Rule

Rule didapat dari *node tree* yang berstatus sebagai *leaf*, dari *leaf – leaf* tersebut diambil aturan dengan menelusuri dari *node* paling atas (*root*) hingga ke *leaf*. Aturan yang didapat tersebut kemudian diterapkan ke data uji untuk mendapatkan class prediksi berdasarkan pohon yang terbentuk.

3.3 Perhitungan Manual

Perhitungan ID3 dilakukan menggunakan 12 data sample dari kasus tindak pidana penganiayaan yang dapat ditunjukkan pada tabel 3.5 berikut.

Tabel 3.5 Deskripsi Aktor

Inisial Nama	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	Hukuman
AS	1	1	1	0	0	0	0	1	0	1	0	Ringan
DR	0	1	1	0	0	0	0	0	0	0	0	Ringan
FG	1	0	1	1	1	0	0	0	0	0	1	Ringan
MLB	1	1	1	0	0	1	0	1	0	1	0	Ringan
KDH	1	1	0	0	0	1	0	0	0	1	0	Ringan
ARN	0	1	1	0	0	0	0	0	0	0	0	Ringan
NB	1	0	0	0	0	1	1	1	0	1	1	Sedang
MAL	0	0	0	0	1	0	0	0	1	0	1	Sedang
LHM	0	0	1	0	0	1	1	0	0	0	0	Sedang
DNS	1	0	0	1	0	0	0	0	1	1	0	Sedang
BN	0	0	0	0	1	0	0	0	1	1	0	Sedang
HOU	1	0	0	1	0	0	0	1	1	0	1	Berat

(lanjutan)

KHS	0	0	0	0	1	0	0	0	1	1	1	Berat
MBS	1	0	0	1	0	0	0	1	0	1	0	Berat
NH	1	0	0	0	1	1	0	0	1	0	1	Berat
AB	1	0	0	0	0	0	0	1	1	0	1	Berat
VN	0	0	0	0	0	0	0	1	1	1	0	Berat

Sumber : pakar

Kode	Deskripsi Parameter
P1	Terdakwa belum pernah dihukum sebelumnya
P2	Terdakwa masih dibawah umur
P3	Dalam persidangan terdakwa berkata jujur dan memperlancar proses persidangan
P4	Terdakwa mempunyai istri dan beberapa anak kecil
P5	Terdakwa berusia lanjut dan sering sakit-sakitan(60)
P6	Terdakwa menyerahkan diri kepada yang berwajib dan mengakui kesalahannya
P7	Terdakwa sedang mengandung kurang lebih 3 bulan
P8	Pada waktu melakukan kejahatan terdakwa sedang dalam masa percobaan
P9	Terdakwa berbelit-belit dan mempersulit jalannya persidangan
P10	Terdakwa mengancam para saksi
P11	Korban berjumlah lebih dari 1 orang

Kategori Hukuman	Lama Hukuman
Ringan	0 - 30% dari maksimal hukuman yang berada dipasal
Sedang	30%-60% dari maksimal hukuman yang berada dipasal
Berat	60% - maksimal

Sumber : pakar

Data yang diperoleh dari wawancara dengan pakar tersebut akan dijadikan sebagai patokan dalam penentuan lamanya hukuman seseorang yang akan diterima pada kasus tindak pidana penganiayaan. Langkah pertama pada perhitungan ID3 ini yaitu mencari *Entropy S (Entropy total)*.

$$\begin{aligned}
\text{Entropy (S)}[R, S, B] &= - \sum_{j=1}^n f_s(j) \log_2 f_s(j) \\
&= - f_s(\text{Ringan}) \log_2 f_s(\text{Ringan}) - \\
&\quad f_s(\text{Sedang}) \log_2 f_s(\text{Sedang}) - f_s(\text{Berat}) \log_2 f_s(\text{Berat}) \\
&= - (6/17) * \text{Log}_2(6/17) - (5/17) * \text{Log}_2(5/17) - (6/17) * \text{Log}_2(6/17) \\
&= - (0,35294) * (\text{Log} (0,35294)/\text{Log}2) - (0,29412) * (\text{Log} \\
&\quad (0,29412)/\text{Log}2) - (0,35294) * (\text{Log} (0,35294)/\text{Log}2) \\
&= - (0,35294) * (-0,4523/0,3010) - (0,29412) * (-0,53148/ \\
&\quad 0,3010) - (0,35294) * (-0,4523/0,3010) \\
&= - (0,35294) * (-1,50266) - (0,29412) * (-1,76571) - \\
&\quad (0,35294) * (-1,50266) \\
&= 0,53035 + 0,51933 + 0,53035 \\
&= 1,57986
\end{aligned}$$

Berikutnya dilakukan perhitungan *Information Gain* untuk masing-masing atribut atau kriteria. Atribut yang pertama yang akan dihitung adalah P1. Value(P1) = {Ya (1), Tidak (0)}.

Perhitungan *Entropy* untuk atribut P1 (Value(P1) = {Ya (1), Tidak (0)}):

$$\begin{aligned}
\text{Entropy}(S_{\text{Ya}})[R, S, B] &= - (4/10) * \text{Log}_2(4/10) - (2/10) * \text{Log}_2(2/10) \\
&\quad - (4/10) * \text{Log}_2(4/10) \\
&= - (0,4) * (\text{Log}(0,4)/\text{Log}2) - (0,2) * (\text{Log}(0,2)/\text{Log}2) \\
&\quad - (0,4) * (\text{Log}(0,4)/\text{Log}2) \\
&= 0,52877 + 0,46439 + 0,52877 \\
&= 1,52193
\end{aligned}$$

$$\begin{aligned}
\text{Entropy}(S_{\text{Tidak}})[R, S, B] &= - (2/7) * \text{Log}_2(2/7) - (3/7) * \text{Log}_2(3/7) - (2/7) * \text{Log}_2(2/7) \\
&= - (0,28571) * (\text{Log}(0,28571)/\text{Log}2) \\
&\quad - (0,42857) * (\text{Log}(0,42857)/\text{Log}2) \\
&\quad - (0,28571) * (\text{Log}(0,28571)/\text{Log}2) \\
&= 0,51639 + 0,52388 + 0,51639 \\
&= 1,55666
\end{aligned}$$

Begitu seterusnya hingga *Entropy* pada atribut P11. Sehingga didapatkan hasil perhitungan *Entropy* semua atribut pada kelas Ya (1) dan Tidak (0) yang ditunjukkan tabel berikut.

Tabel 3.6 Nilai *Entropy* atribut perhitungan manual tahap I

Atribut	<i>Entropy</i>	
	Ya	Tidak
P1	1,52193	1,55666
P2	0,00000	1,32501
P3	1,35165	0,65564
P4	1,50000	0,516228
P5	0,59316	1,554585
P6	1,52193	1,554585
P7	0,00000	1,521928
P8	1,37878	1,521928
P9	1,55459	1,370951
P10	1,58497	1,561278
P11	1,37878	1,485475

Perhitungan *Information Gain* untuk atribut P1:

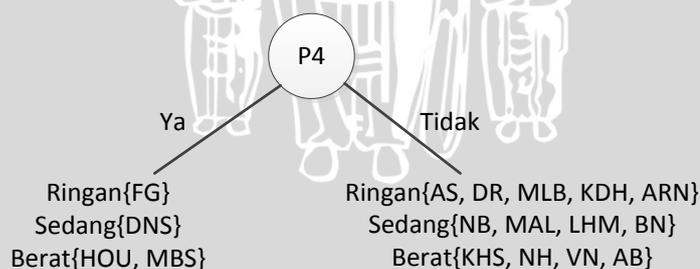
$$\begin{aligned}
 \text{Gain (S, P1)} &= \text{Entropy (S)} - ((10/17) * \text{Entropy}(S_{\text{Ya}})[R, S, B] + \\
 &\quad (7/17) * \text{Entropy}(S_{\text{Tidak}})[R, S, B]) \\
 &= 1,57986 - ((10/17) * (1,52193) + (7/17) * (1,55666)) \\
 &= 1,57986 - (0,98525 + 0,64098) \\
 &= 0,04364
 \end{aligned}$$

Menggunakan cara yang sama seperti perhitungan *Information Gain* pada atribut P1, begitupun seterusnya untuk perhitungan *Information Gain* pada atribut P2 hingga P11. Sehingga hasil *Information Gain* masing-masing atribut dapat dilihat pada tabel berikut.

Tabel 3.7 Nilai *Information Gain* perhitungan manual tahap I

Atribut	IG(S)
P1	0,04364
P2	0,64456
P3	0,55575
P4	0,83216
P5	0,30805
P6	0,03488
P7	0,23699
P8	0,11688
P9	0,07929
P10	0,00605
P11	0,13832

Pemilihan atribut sebagai *node* atau *root* dipilih berdasarkan nilai *Information Gain* yang dihasilkan. Atribut yang memiliki nilai *Information Gain* terbesar akan dijadikan *root* atau *node*. Berdasarkan tabel hasil perhitungan *Information Gain* untuk masing-masing atribut di atas dapat dilihat bahwa atribut yang memiliki *Information Gain* terbesar adalah P4 dengan nilai 0,9570914. Hal ini berarti kriteria P4 menjadi prioritas utama dalam menentukan hasil keputusan tindak pidana penganiayaan. Sehingga atribut P4 dijadikan *root* untuk awal pembentukan *tree*.



Gambar 3.23 *Tree* hasil perhitungan manual tahap I.

Oleh karena P4 sebagai *root* atau *node* yang terpilih maka data dikelompokkan berdasarkan kelas pada atribut P4 yaitu Ya (1) dan Tidak (0). Pada kelas Ya (1) terdapat 4 anggota dengan 3 kategori yaitu FG di kategori Ringan, DNS di kategori Sedang, HOU dan MBS di kategori Berat. Pada kelas Tidak (0) terdapat 13 anggota dengan 3 kategori yaitu AS, DR, MLB, KDH, dan ARN di kategori Ringan, NB, MAL, LHM, dan BN di kategori Sedang, serta KHS, NH, VN, AB di kategori Berat.



Perhitungan ID3 akan dilanjutkan hingga anggota dari cabang *root* atau *node* hanya menyisakan satu kategori.

Perhitungan ID3 pada kelas Ya (1) node P4

Tabel berikut adalah data tindak pidana penganiayaan yang ada pada kelas Ya (1) atribut P4. Untuk atribut P4 tidak diikuti dalam perhitungan lagi karena atribut P4 telah menjadi *root* atau *node* pada perhitungan sebelumnya.

Tabel 3.8 Data Tindak Pidana kelas Ya atribut P4

Inisial Nama	P1	P2	P3	P5	P6	P7	P8	P9	P10	P11	Hukuman
FG	1	0	1	1	0	0	0	1	0	1	Ringan
DNS	1	0	0	0	0	0	0	1	1	0	Sedang
HOU	1	0	0	0	0	0	1	1	0	1	Berat
MBS	1	0	1	0	0	0	1	0	1	0	Berat

Seperti pada perhitungan ID3 sebelumnya, langkah pertama yaitu mencari nilai *Entropy* total. Perhitungan *Entropy* ini dilakukan terhadap masing-masing atribut yang ada di kelas Ya (1) atribut P4 menggunakan data pada tabel di atas. Sehingga *Entropy* total yang digunakan adalah nilai *Entropy* dari kelas Ya(1) atribut P4. $Entropy(s) = 1,50000$.

Perhitungan *Entropy* untuk atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Ya (1) atribut P4:

$$\begin{aligned}
 Entropy(S_{Ya})[R, S, B] &= -(1/4)*Log_2(1/4) - (1/4)*Log_2(1/4) - (2/4)*Log_2(2/4) \\
 &= -(0,25)*(Log(0,25)/Log2) - (0,25)*(Log(0,25)/Log2) \\
 &\quad - (0,5)*(Log(0,5)/Log2) \\
 &= 0,5 + 0,5 + 0,5 \\
 &= 1,5
 \end{aligned}$$

$$Entropy(S_{Tidak})[R, S, B] = 0$$

$Entropy(S_{Tidak})[R, S, B]$ bernilai nol (0) karena tidak ada anggota yang memiliki kelas Tidak (0) pada atribut P1. Begitu seterusnya hingga perhitungan *Entropy* pada atribut P11. Sehingga didapatkan nilai *Entropy* semua atribut yang ada di cabang kelas Ya (1) atribut P4 yang ditunjukkan pada tabel berikut.



Tabel 3.9 Nilai *Entropy* atribut perhitungan manual tahap II

Kriteria	<i>Entropy</i>	
	Ya	Tidak
P1	1,50000	0,00000
P2	0,00000	1,50000
P3	1,00000	1,00000
P5	0,00000	0,91830
P6	0,00000	1,50000
P7	0,00000	1,50000
P8	0,00000	1,00000
P9	1,58496	0,00000
P10	1,00000	1,00000
P11	1,00000	1,00000

Perhitungan *Information Gain* atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Ya (1) atribut P4:

$$\begin{aligned}
 \text{Gain}(S, P1) &= \text{Entropy}(S) - ((4/4) * \text{Entropy}(S_{Ya})[R, S, B] + (0/4) * \text{Entropy}(S_{Tidak})[R, S, B]) \\
 &= 1,50000 - ((4/4) * (1,50000) + (0/4) * (0)) \\
 &= 1,50000 - 1,50000 - 0 \\
 &= 0
 \end{aligned}$$

Begitupun seterusnya dilakukan perhitungan *Information Gain* pada atribut P2 hingga P11. Sehingga hasil *Information Gain* semua atribut yang ada di cabang kelas Ya (1) atribut P4 dapat dilihat pada tabel berikut.

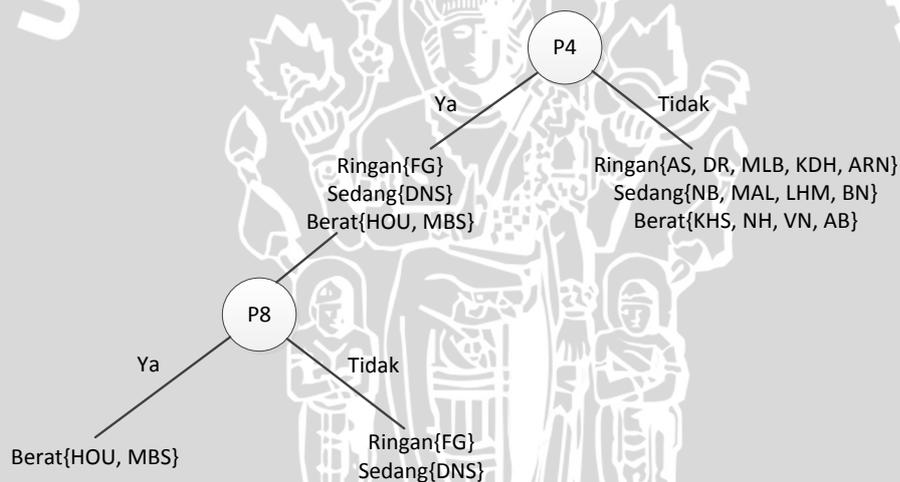
Tabel 3.10 Nilai *Information Gain* perhitungan manual tahap II

Atribut	IG(S)
P1	0,00000
P2	0,00000
P3	0,50000
P5	0,81128

Table 3.10 Nilai *information Gain* perhitungan manual tahap II (lanj)

Atribut	IG(S)
P6	0,00000
P7	0,00000
P8	1,00000
P9	0,31128
P10	0,50000
P11	0,50000

Berdasarkan tabel di atas dapat dilihat bahwa nilai *Information Gain* tertinggi adalah 1,00000 yaitu pada atribut P8. Kelas Ya (1) berisi kategori Sedang dan kelas Tidak (0) berisi kategori Berat. Kemudian data pada kelas Ya (1) atribut P4 dikelompokkan berdasarkan kelas pada atribut P8 yaitu kelas Ya (1) dan Tidak (0). Sehingga gambar *tree* yang baru dapat ditunjukkan pada gambar berikut.

**Gambar 3.24** *Tree* hasil perhitungan manual tahap II.

Pada kelas Ya (1) berisi kategori Berat yang beranggotakan HOU dan MBS, sedangkan kelas Tidak (0) terdapat 2 anggota dengan 2 kategori yaitu FG di kategori Ringan dan DNS di kategori Sedang. Oleh karena pada cabang Tidak (0) *node* P8 masih terdapat lebih dari 1 kategori maka perhitungan ID3 akan dilanjutkan hingga anggota dari cabang *root* atau *node* hanya menyisakan satu kategori.

Perhitungan ID3 pada kelas Tidak (0) *node* P8

Tabel berikut adalah data tindak pidana penganiayaan yang ada pada kelas Tidak (0) atribut P8. Untuk atribut P8 tidak diikuti dalam perhitungan lagi karena atribut P8 telah menjadi *root* atau *node* pada perhitungan sebelumnya.

Tabel 3.11 Data Tindak Pidana kelas Ya atribut P8

Nama	P1	P2	P3	P5	P6	P7	P9	P10	P11	Hukuman
FG	1	0	1	1	0	0	1	0	1	Ringan
DNS	1	0	0	0	0	0	1	1	0	Sedang

Seperti pada perhitungan ID3 sebelumnya, langkah pertama yaitu mencari nilai *Entropy* total. Perhitungan *Entropy* ini dilakukan terhadap masing-masing atribut yang ada di kelas Tidak (0) atribut P8 menggunakan data pada tabel di atas. Sehingga *Entropy* total yang digunakan adalah nilai *Entropy* dari kelas Tidak(0) atribut P8. $Entropy(s) = 1,00000$.

Perhitungan *Entropy* untuk atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) atribut P8:

$$\begin{aligned}
 Entropy(S_{Ya})[R, S, B] &= -(1/2) * \text{Log}_2(1/2) - (1/2) * \text{Log}_2(1/2) \\
 &= -(0,5) * (\text{Log}(0,5) / \text{Log}2) - (0,5) * (\text{Log}(0,5) / \text{Log}2) \\
 &= 0,5 + 0,5 \\
 &= 1,00000
 \end{aligned}$$

$$Entropy(S_{Tidak})[R, S, B] = 0$$

$Entropy(S_{Tidak})[R, S, B]$ bernilai nol (0) karena tidak ada anggota yang memiliki kelas Tidak (0) pada atribut P1. Begitu seterusnya hingga perhitungan *Entropy* pada atribut P11. Sehingga didapatkan nilai *Entropy* semua atribut yang ada di cabang kelas Tidak (0) atribut P8 yang ditunjukkan pada tabel berikut.

Tabel 3.12 Nilai *Entropy* atribut perhitungan manual tahap III

Kriteria	<i>Entropy</i>	
	Ya	Tidak
P1	1,00000	0,00000
P2	0,00000	1,00000
P3	0,00000	0,00000
P5	0,00000	0,00000
P6	0,00000	1,00000
P7	0,00000	1,00000
P9	1,00000	0,00000
P10	0,50000	0,50000



Tabel 3.12 Nilai entropy atribut perhitungan manual tahap III (lanj)

Kriteria	Entropy	
	Ya	Tidak
P11	0,00000	0,00000

Perhitungan *Information Gain* atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) atribut P8:

$$\begin{aligned}
 \text{Gain (S, P1)} &= \text{Entropy (S)} - ((2/2) * \text{Entropy}(\text{Sya})[\text{R, S, B}] + (0/2) * \\
 &\quad \text{Entropy}(\text{Stidak})[\text{R, S, B}]) \\
 &= 1,00000 - ((2/2) * (1,00000) + (0/2) * (0)) \\
 &= 1,00000 - 1,00000 - 0 \\
 &= 0
 \end{aligned}$$

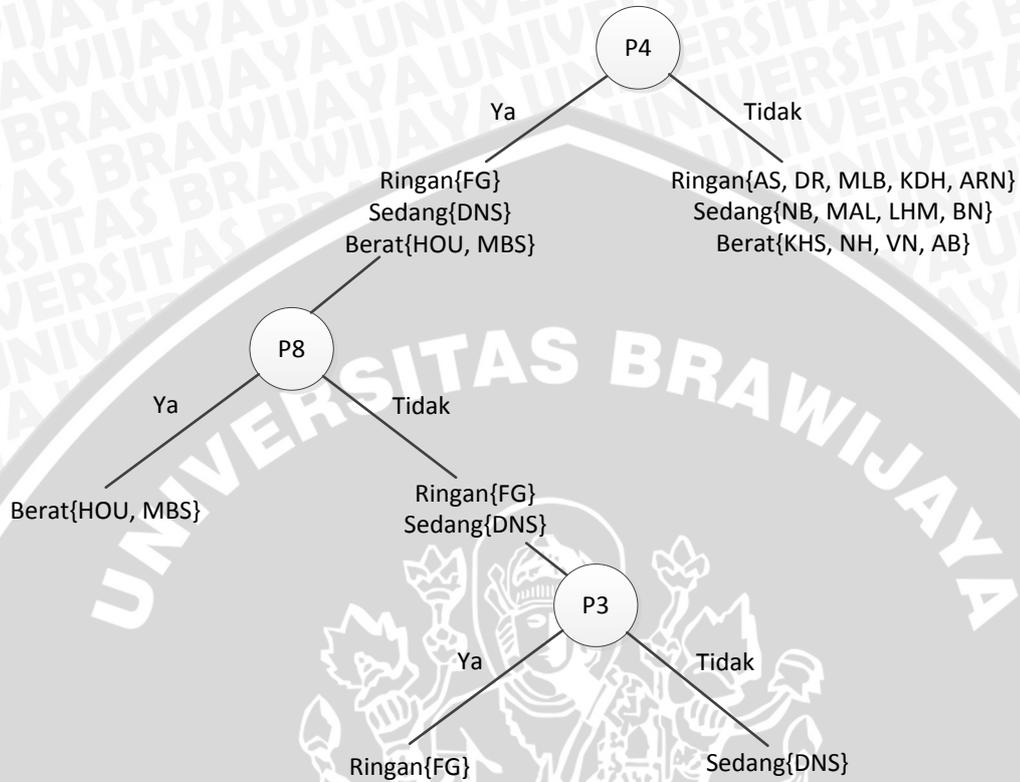
Begitupun seterusnya dilakukan perhitungan *Information Gain* pada atribut P2 hingga P11. Sehingga hasil *Information Gain* semua atribut yang ada di cabang kelas Tidak (0) atribut P8 dapat dilihat pada tabel berikut.

Tabel 3.13 Nilai *Information Gain* perhitungan manual tahap III

Atribut	IG(S)
P1	0,00000
P2	0,00000
P3	1,00000
P5	1,00000
P6	0,00000
P7	0,00000
P9	0,00000
P10	0,50000
P11	1,00000

Berdasarkan tabel di atas dapat dilihat bahwa nilai *Information Gain* tertinggi adalah 1,00000 yaitu pada atribut P3, P5, dan P11. Karena ketiga atribut memiliki nilai *Information Gain* yang sama dan dapat memisahkan kategori secara maksimal, sehingga pemilihan atribut dengan kode terkecil yang akan dipilih sebagai *node* selanjutnya yaitu P3. Pada atribut P3, Kelas Ya (1) berisi kategori Ringan dan kelas Tidak (0) berisi kategori Sedang. Kemudian data pada kelas Tidak

(1) atribut P8 dikelompokkan berdasarkan kelas yang ada pada atribut P3 yaitu kelas Ya (1) dan Tidak (0). Sehingga gambar *tree* yang baru dapat ditunjukkan pada gambar berikut.



Gambar 3.25 Tree hasil perhitungan manual tahap III

Perhitungan ID3 pada kelas Tidak (0) node P4

Tabel berikut adalah data tindak pidana penganiayaan yang ada pada kelas Tidak (0) *node* P4. Untuk atribut P4 tidak diikuti dalam perhitungan lagi karena atribut P4 telah menjadi *root* atau *node* pada perhitungan sebelumnya. Sedangkan atribut P8 tetap diikuti dalam perhitungan karena penetapan atribut P8 sebagai *node* sebelumnya dilakukan di kelas yang berbeda pada *root* atau *node* P4.

Tabel 3.14 Data Tindak Pidana kelas Tidak atribut P4

Nama	P1	P2	P3	P5	P6	P7	P8	P9	P10	P11	Hukuman
AS	1	1	1	0	0	0	1	1	1	0	Ringan
DR	0	1	1	0	0	0	0	0	0	0	Ringan
MLB	1	1	1	0	1	0	1	1	1	0	Ringan
KDH	1	1	0	0	1	0	0	0	1	0	Ringan
ARN	0	1	1	0	0	0	0	0	0	0	Ringan

Tabel 3.14 Data Tindak Pidana kelas Tidak atribut P4 (lanj)

Nama	P1	P2	P3	P5	P6	P7	P8	P9	P10	P11	Hukuman
NB	1	0	0	0	1	1	1	0	1	1	Sedang
MAL	0	0	0	1	0	0	0	1	0	1	Sedang
LHM	0	0	1	0	1	1	0	1	0	0	Sedang
BN	0	0	0	1	0	0	0	1	1	0	Sedang
KHS	0	0	1	1	0	0	0	1	1	1	Berat
NH	1	0	0	1	1	0	0	1	0	1	Berat
VN	0	0	1	0	0	0	1	1	1	0	Berat
AB	1	0	0	0	0	0	1	1	0	1	Berat

Seperti pada perhitungan ID3 sebelumnya, langkah pertama yaitu mencari nilai *Entropy* total. Perhitungan *Entropy* ini dilakukan terhadap masing-masing atribut yang ada di kelas Tidak (0) atribut P4 menggunakan data pada tabel di atas. Sehingga *Entropy* total yang digunakan adalah nilai *Entropy* dari kelas Tidak (0) atribut P4. $Entropy(s) = 0,516228$.

Perhitungan *Entropy* untuk atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) atribut P4:

$$\begin{aligned}
 Entropy(Sya)[R, S, B] &= -(3/6) * \text{Log}_2(3/6) - (1/6) * \text{Log}_2(1/6) - (2/6) * \text{Log}_2(2/6) \\
 &= -(0,5) * (\text{Log}(0,5) / \text{Log}2) - (0,1667) * (\text{Log}(0,1667) / \text{Log}2) \\
 &\quad - (0,333) * (\text{Log}(0,333) / \text{Log}2) \\
 &= -(0,5) * (-1) - (0,1667) * (-2,58496) - (0,333) * (-1,58496) \\
 &= 0,5 - 0,43083 - 0,52832 \\
 &= 1,45915
 \end{aligned}$$

$$\begin{aligned}
 Entropy(Stidak)[R, S, B] &= -(2/7) * \text{Log}_2(2/7) - (3/7) * \text{Log}_2(3/7) - (2/7) * \text{Log}_2(2/7) \\
 &= -(0,28571) * (\text{Log}(0,28571) / \text{Log}2) \\
 &\quad - (0,42857) * (\text{Log}(0,42857) / \text{Log}2) \\
 &\quad - (0,28571) * (\text{Log}(0,28571) / \text{Log}2) \\
 &= -(0,28571) * (-0,80735) - (0,42857) * (-1,22239) \\
 &\quad - (0,28571) * (-0,80735) \\
 &= 0,50000 + 0,43083 + 0,50000
 \end{aligned}$$

$$= 1,55666$$

Begitu seterusnya hingga *Entropy* pada atribut P11. Sehingga didapatkan hasil perhitungan *Entropy* semua atribut pada kelas Tidak (1) dan Tidak (0) yang ditunjukkan tabel berikut.

Tabel 3.15 Nilai *Entropy* atribut perhitungan manual tahap IV

Kriteria	<i>Entropy</i>	
	Ya	Tidak
P1	1,45915	1,55666
P2	0,00000	1,00000
P3	1,37878	1,45915
P6	1,52192	1,56128
P7	0,00000	1,49491
P8	1,52193	1,56128
P9	1,53049	0,81128
P10	1,55666	1,58496
P11	0,97095	1,29879

Perhitungan *Information Gain* atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) atribut P4:

$$\begin{aligned}
 \text{Gain}(S, P1) &= \text{Entropy}(S) - ((5/9) * \text{Entropy}(S_{ya})[R, S, B] + (4/9) * \text{Entropy}(S_{tidak})[R, S, B]) \\
 &= 0,516228 - ((5/9) * (1,45915) + (4/9) * (1,55666)) \\
 &= 0,516228 - 0,67345 - 0,83820 \\
 &= -0,99543
 \end{aligned}$$

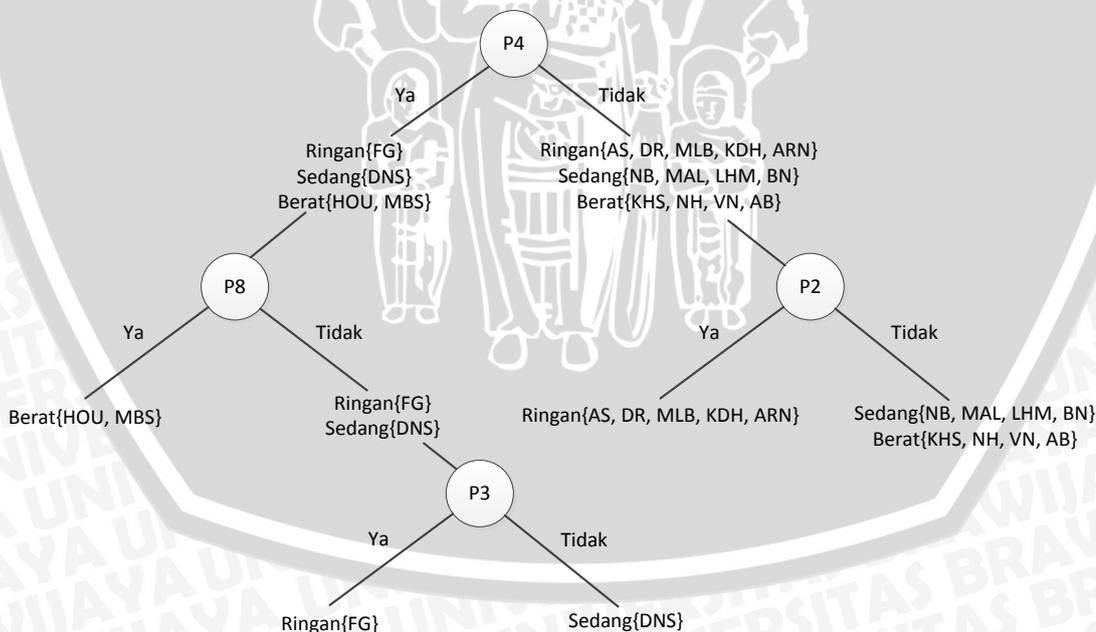
Begitupun seterusnya dilakukan perhitungan *Information Gain* hingga P11. Sehingga hasil *Information Gain* semua atribut yang ada di cabang kelas Tidak (0) atribut P4 dapat dilihat pada tabel berikut.



Tabel 3.16 Nilai *Information Gain* perhitungan manual tahap IV

Atribut	IG(S)
P1	-0,99543
P2	-0,09914
P3	-0,89965
P6	-1,02689
P7	-0,74870
P8	-1,02992
P9	-0,79297
P10	-1,05349
P11	-0,65647

Berdasarkan tabel di atas dapat dilihat bahwa nilai *Information Gain* tertinggi yaitu pada atribut P2 sebesar -0,09914. Sehingga atribut P2 dipilih sebagai *node* berikutnya pada cabang kelas Tidak (0) atribut P4. Kemudian data pada kelas Tidak (0) atribut P4 dikelompokkan berdasarkan kelas yang ada pada P2 yaitu Ya (1) dan Tidak (0). Sehingga gambar *tree* yang baru dapat ditunjukkan pada gambar berikut.



Gambar 3.26 *Tree* hasil perhitungan manual tahap IV

Pada *node* dengan atribut P2 kelas Ya (1) berisi kategori Ringan yang beranggotakan AS, DR, MLB, KDH, dan ARN sedangkan untuk kelas Tidak (0) berisi



2 kategori yaitu Sedang yang beranggotakan NB, MAL, LHM, dan BN, kategori Berat beranggotakan KHS, NH, VN, dan AB. Untuk kelas Ya (1) menyisakan satu kategori sehingga perhitungan ID3 untuk cabang Ya (1) pada *node* P2 berhenti. Sedangkan kelas Tidak (0) memiliki dua kategori sehingga perhitungan ID3 untuk cabang Tidak (0) pada *node* P2 dilanjutkan.

Perhitungan ID3 pada kelas Tidak (0) *node* P2

Tabel berikut adalah data tindak pidana penganiayaan yang masuk pada kelas Tidak (0) *node* P2. Untuk atribut P2 tidak diikuti dalam perhitungan selanjutnya karena atribut P2 telah terpilih sebagai *node* pada perhitungan sebelumnya.

Tabel 3.17 Data Tindak Pidana kelas Tidak atribut P2

Nama	P1	P3	P5	P6	P7	P8	P9	P10	P11	Hukuman
NB	1	0	0	1	1	1	0	1	1	Sedang
MAL	0	0	1	0	0	0	1	0	1	Sedang
LHM	0	1	0	1	1	0	1	0	0	Sedang
BN	0	0	1	0	0	0	1	1	0	Sedang
KHS	0	1	1	0	0	0	1	1	1	Berat
NH	1	0	1	1	0	0	1	0	1	Berat
VN	0	1	0	0	0	1	1	1	0	Berat
AB	1	0	0	0	0	1	1	0	1	Berat

Seperti pada perhitungan ID3 sebelumnya, langkah pertama yaitu mencari nilai *Entropy* total. Perhitungan *Entropy* ini dilakukan terhadap masing-masing atribut yang ada di kelas Tidak (0) atribut P2 menggunakan data pada tabel di atas. Sehingga *Entropy* total yang digunakan adalah nilai *Entropy* dari kelas Tidak (0) *node* P2. $Entropy(s) = 1,00000$.

Perhitungan *Entropy* untuk atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) *node* P2:

$$\begin{aligned}
 Entropy(S_{ya})[S, B] &= - (1/3) * \text{Log}_2(1/3) - (2/3) * \text{Log}_2(2/3) \\
 &= - (0,333) * (\text{Log}(0,333)/\text{Log}2) \\
 &\quad - (0,667) * (\text{Log}(0,667)/\text{Log}2) \\
 &= - (0,333) * (-0,58496) - (0,667) * (-0,58496) \\
 &= 0,91830
 \end{aligned}$$

$$Entropy(S_{tidak})[S, B] = - (3/5) * \text{Log}_2(3/5) - (2/5) * \text{Log}_2(2/5)$$



$$\begin{aligned}
 &= - (0,6) * (\text{Log}(0,6)/\text{Log}2) \\
 &\quad - (0,4) * (\text{Log} (0,4) / \text{Log}2) \\
 &= - (0,6)*(- 0,73697) - (0,4)*(- 1,32193) \\
 &= 0,44218 + 0,52877 \\
 &= 0,97095
 \end{aligned}$$

Begitu seterusnya hingga *Entropy* pada atribut P11. Sehingga didapatkan hasil perhitungan *Entropy* semua atribut pada kelas Tidak (1) dan Tidak (0) yang ditunjukkan tabel berikut.

Tabel 3.18 Nilai *Entropy* atribut perhitungan manual tahap V

Atribut	<i>Entropy</i>	
	Ya	Tidak
P1	0,91830	0,97095
P3	0,91830	0,97095
P5	1,00000	1,00000
P6	0,91830	0,97095
P7	0,00000	0,91830
P8	0,91830	0,97095
P9	0,98523	0,00000
P10	1,00000	1,00000
P11	0,97095	0,91830

Perhitungan *Information Gain* atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) node P2:

$$\begin{aligned}
 \text{Gain (S, P1)} &= \text{Entropy (S)} - ((3/8)*\text{Entropy}(\text{Sya})[\text{S, B}] + (5/8) * \\
 &\quad \text{Entropy}(\text{Stidak})[\text{S, B}]) \\
 &= 1,00000 - ((3/8)*(0,91830) + (5/8)*(0,97095)) \\
 &= 1,00000 - 0,34436 - 0,60684 \\
 &= 0,04879
 \end{aligned}$$

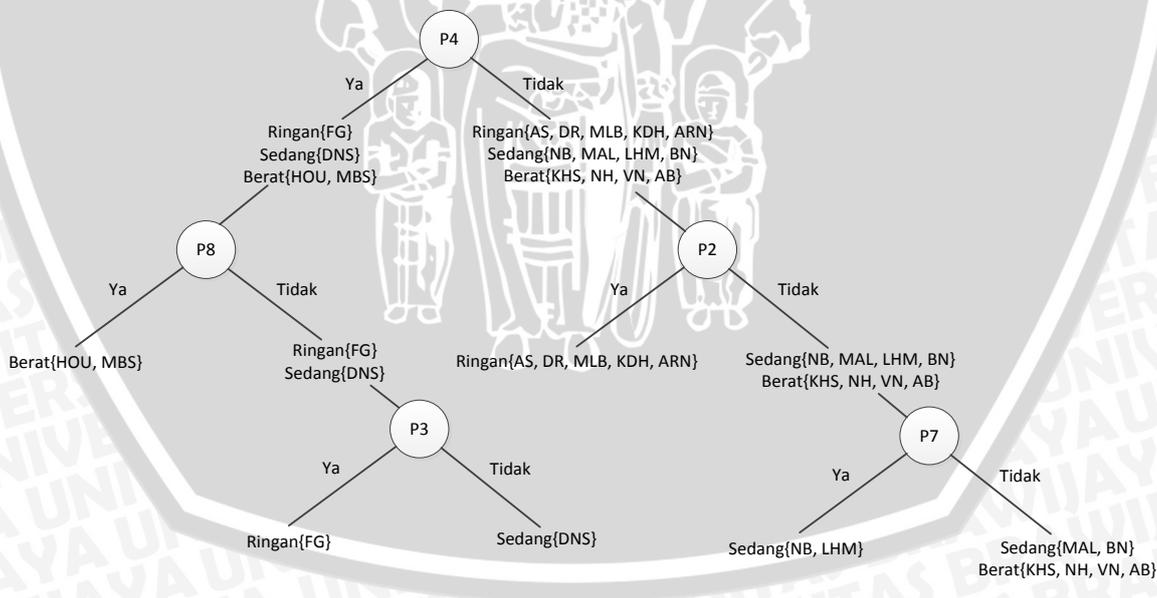
Begitupun seterusnya dilakukan perhitungan *Information Gain* hingga P11. Sehingga hasil *Information Gain* semua atribut yang ada di cabang kelas Tidak (0) ndoe P2 dapat dilihat pada tabel berikut.



Tabel 3.19 Nilai *Information Gain* perhitungan manual tahap V

Atribut	IG(X,S)
P1	0,04879
P3	0,04879
P5	0,00000
P6	0,04879
P7	0,31128
P8	0,04879
P9	0,13793
P10	0,00000
P11	0,04879

Berdasarkan tabel di atas dapat dilihat bahwa nilai *Information Gain* tertinggi yaitu pada atribut P7 sebesar 0,31128. Sehingga atribut P7 dipilih sebagai *node* berikutnya pada cabang kelas Tidak (0) *node* P2. Kemudian data pada kelas Tidak (0) *node* P2 dikelompokkan berdasarkan kelas yang ada pada P7 yaitu Ya (1) dan Tidak (0). Sehingga gambar *tree* yang baru dapat ditunjukkan pada gambar berikut.



Gambar 3.27 *Tree* hasil perhitungan manual tahap V

Pada *node* dengan atribut P7 kelas Ya (1) berisi kategori Sedang yang beranggotakan NB dan LHM, sedangkan untuk kelas Tidak (0) berisi 2 kategori yaitu Sedang yang beranggotakan MAL dan BN, kategori Berat beranggotakan KHS,



NH, VN, dan AB. Untuk kelas Ya (1) menyisakan satu kategori sehingga perhitungan ID3 untuk cabang Ya (1) pada atribut P7 berhenti. Sedangkan kelas Tidak (0) memiliki dua kategori sehingga perhitungan ID3 untuk cabang Tidak (0) pada atribut P7 dilanjutkan.

Perhitungan ID3 pada kelas Tidak (0) node P7

Tabel berikut adalah data tindak pidana penganiayaan yang masuk pada kelas Tidak (0) node P7. Untuk atribut P7 tidak diikuti dalam perhitungan selanjutnya karena atribut P7 telah terpilih sebagai node pada perhitungan sebelumnya.

Tabel 3.20 Data Tindak Pidana kelas Tidak atribut P7

Nama	P1	P3	P5	P6	P8	P9	P10	P11	Hukuman
MAL	0	0	1	0	0	1	0	1	Sedang
BN	0	0	1	0	0	1	1	0	Sedang
KHS	0	1	1	0	0	1	1	1	Berat
NH	1	0	1	1	0	1	0	1	Berat
VN	0	1	0	0	1	1	1	0	Berat
AB	1	0	0	0	1	1	0	1	Berat

Seperti pada perhitungan ID3 sebelumnya, langkah pertama yaitu mencari nilai *Entropy* total. Perhitungan *Entropy* ini dilakukan terhadap masing-masing atribut yang ada di kelas Tidak (0) atribut P7 menggunakan data pada tabel di atas. Sehingga *Entropy* total yang digunakan adalah nilai *Entropy* dari kelas Tidak (0) node P7. $Entropy(s) = 0,91830$.

Perhitungan *Entropy* untuk atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) node P7:

$$\begin{aligned}
 Entropy(S_{ya})[S, B] &= 0 - (2/2) * \text{Log}_2(2/2) \\
 &= 0 - (1) * (\text{Log}(1) / \text{Log}2) \\
 &= -0 - (1) * (0) \\
 &= 0 + 0 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 Entropy(S_{tidak})[S, B] &= - (2/4) * \text{Log}_2(2/4) - (2/4) * \text{Log}_2(2/4) \\
 &= - (0,5) * (\text{Log}(0,5) / \text{Log}2) \\
 &\quad - (0,5) * (\text{Log}(0,5) / \text{Log}2)
 \end{aligned}$$



$$\begin{aligned}
 &= - (0,5)*(-1) - (0,5)*(-1) \\
 &= 0,5 + 0,5 \\
 &= 1
 \end{aligned}$$

Begitu seterusnya hingga *Entropy* pada atribut P11. Sehingga didapatkan hasil perhitungan *Entropy* semua atribut pada kelas Tidak (1) dan Tidak (0) yang ditunjukkan tabel berikut.

Tabel 3.21 Nilai *Entropy* atribut perhitungan manual tahap VI

Kriteria	<i>Entropy</i>	
	Ya	Tidak
P1	0,00000	1,00000
P3	0,00000	1,00000
P5	1,00000	0,00000
P6	0,00000	0,97095
P7	0,00000	1,00000
P9	0,00000	0,00000
P10	0,91830	0,91830
P11	0,81130	1,00000

Perhitungan *Information Gain* atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) node P7:

$$\begin{aligned}
 \text{Gain (S, P1)} &= \text{Entropy (S)} - ((2/6) * \text{Entropy(Sya)[S, B]} + (4/6) * \text{Entropy(Stidak)[S, B]}) \\
 &= 0,91830 - ((0,333)*(0,0000) + (0,667)*(1,00000)) \\
 &= 0,91830 - 0,00000 - 0,66700 \\
 &= - 0,25163
 \end{aligned}$$

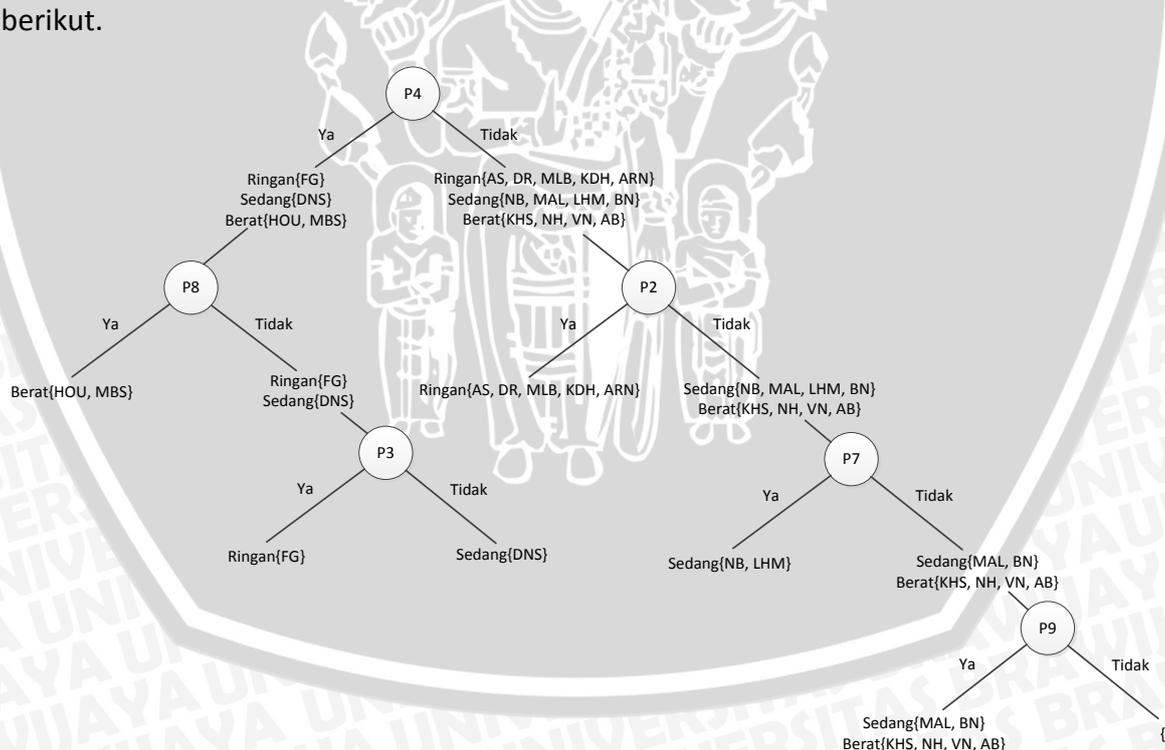
Begitupun seterusnya dilakukan perhitungan *Information Gain* hingga P11. Sehingga hasil *Information Gain* semua atribut yang ada di cabang kelas Tidak (0) node P7 dapat dilihat pada tabel berikut.



Tabel 3.22 Nilai *Information Gain* perhitungan manual tahap VI

Kriteria	IG(X,S)
P1	0,25163
P3	0,25163
P5	0,25163
P6	0,33572
P7	0,25163
P9	0,91830
P10	0,00000
P11	0,04411

Berdasarkan tabel di atas dapat dilihat bahwa nilai *Information Gain* tertinggi yaitu pada *node* P8 sebesar 0,91830. Sehingga atribut P9 dipilih sebagai *node* berikutnya pada cabang kelas Tidak (0) *node* P7. Kemudian data pada kelas Tidak (0) *node* P7 dikelompokkan berdasarkan kelas yang ada pada P9 yaitu Ya (1) dan Tidak (0). Sehingga gambar *tree* yang baru dapat ditunjukkan pada gambar berikut.



Gambar 3.28 *Tree* hasil perhitungan manual tahap VI

Pada *node* dengan atribut P9 kelas Ya (1) berisi 2 kategori yaitu Sedang yang beranggotakan MAL, dan BN, kategori Berat beranggotakan KHS, NH, VN, dan



AB, sedangkan kelas Tidak (0) tidak memiliki anggota. Untuk kelas Tidak (0) tidak memiliki satu pun kategori sehingga perhitungan ID3 untuk cabang Tidak (0) pada atribut P9 berhenti. Sedangkan kelas Ya (1) memiliki dua kategori sehingga perhitungan ID3 untuk cabang Ya (1) pada atribut P9 dilanjutkan.

Perhitungan ID3 pada kelas Ya (1) node P9

Tabel berikut adalah data tindak pidana penganiayaan yang masuk pada kelas Ya (1) node P9. Untuk atribut P9 tidak diikuti dalam perhitungan selanjutnya karena atribut P9 telah terpilih sebagai *node* pada perhitungan sebelumnya.

Tabel 3.23 Data Tindak Pidana kelas Ya node P9

Nama	P1	P3	P5	P6	P8	P10	P11	Hukuman
MAL	0	0	1	0	0	0	1	Sedang
BN	0	0	1	0	0	1	0	Sedang
KHS	0	1	1	0	0	1	1	Berat
NH	1	0	1	1	0	0	1	Berat
VN	0	1	0	0	1	1	0	Berat
AB	1	0	0	0	1	0	1	Berat

Seperti pada perhitungan ID3 sebelumnya, langkah pertama yaitu mencari nilai *Entropy* total. Perhitungan *Entropy* ini dilakukan terhadap masing-masing atribut yang ada di kelas Ya (1) atribut P9 menggunakan data pada tabel di atas. Sehingga *Entropy* total yang digunakan adalah nilai *Entropy* dari kelas Ya (1) node P9. $Entropy(s) = 0,0000$.

Perhitungan *Entropy* untuk atribut P1 ($Value(P1) = \{Ya(1), Tidak(0)\}$) pada kelas Tidak (0) node P9:

$$\begin{aligned}
 Entropy(Sya)[S, B] &= 0 - (2/2) * \log_2(2/2) \\
 &= 0 - (1) * (\log(2)/\log(2)) \\
 &= -0 - (1) * (0) \\
 &= 0 + 0 \\
 &= 0
 \end{aligned}$$



$$\begin{aligned}
 Entropy(Stidak)[S, B] &= -0 - (2/4)*\text{Log}_2(2/4) - (2/4)*\text{Log}_2(2/4) \\
 &= - (0,5) * (\text{Log}(0,5)/\text{Log}2) \\
 &\quad - (0,5) * (\text{Log}(0,5)/\text{Log}2) \\
 &= - (0,5)*(-1) - (0,5)*(-1) \\
 &= 0,5 + 0,5 \\
 &= 1
 \end{aligned}$$

Begitu seterusnya hingga *Entropy* pada atribut P11. Sehingga didapatkan hasil perhitungan *Entropy* semua atribut pada kelas Ya (1) dan Tidak (0) yang ditunjukkan tabel berikut.

Tabel 3.24 Nilai *Entropy* atribut perhitungan manual tahap VII

Kriteria	<i>Entropy</i>	
	Ya	Tidak
P1	0,00000	1,00000
P3	0,00000	1,00000
P5	1,00000	0,00000
P6	0,00000	0.97095
P8	0,00000	1,00000
P10	0.91830	0.91830
P11	0.81128	1,00000

Perhitungan *Information Gain* atribut P1 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) node P9:

$$\begin{aligned}
 Gain(S, P1) &= Entropy(S) - ((2/6)*Entropy(Sya)[S, B] \\
 &\quad + (4/6)*Entropy(Stidak)[S, B]) \\
 &= 0,0000 - ((0,3333)*(0,0000) + (0,6667)*(0.1000)) \\
 &= 0,0000 - 0,0000 - 0,6667 \\
 &= - 0,6667
 \end{aligned}$$

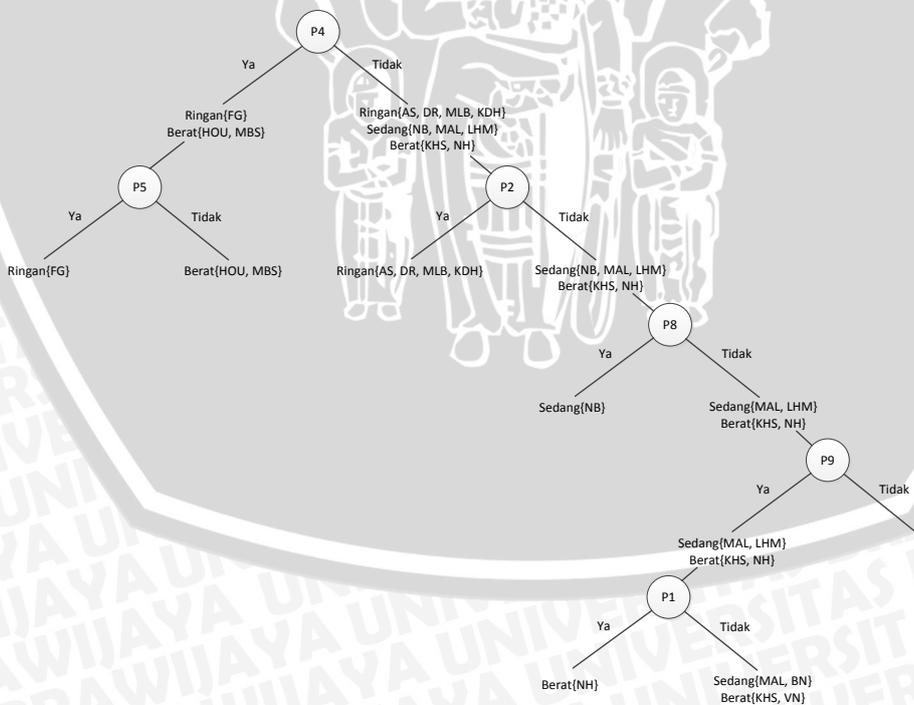
Begitupun seterusnya dilakukan perhitungan *Information Gain* hingga P11. Sehingga hasil *Information Gain* semua atribut yang ada di cabang kelas Tidak (0) node P9 dapat dilihat pada tabel berikut.



Tabel 3.25 Nilai *Information Gain* perhitungan manual tahap VII

Kriteria	IG(X,S)
P1	-0.66667
P3	-0.66667
P5	-0.66667
P6	-0.80912
P8	-0.66667
P10	-0.91830
P11	-0.87420

Berdasarkan tabel di atas dapat dilihat bahwa nilai *Information Gain* tertinggi yaitu -0.66667. Terdapat beberapa atribut yang memiliki *Information Gain* tertinggi, maka sebagai penentuan *node* berikutnya akan dipilih atribut yang mengelompokkan kategori paling maksimal. Atribut P1 dipilih sebagai *node* pada cabang kelas Ya (1) *node* P9 karena memiliki nilai *Information Gain* tertinggi dan atribut pertama yang mengelompokkan kategori paling maksimal. Kemudian data pada kelas Ya (1) *node* P9 dikelompokkan berdasarkan kelas yang ada pada P1 yaitu Ya (1) dan Tidak (0). Sehingga gambar *tree* yang baru dapat ditunjukkan pada gambar berikut.



Gambar 3.29 *Tree* hasil perhitungan manual tahap VII



Pada *node* dengan atribut P1 kelas Ya (1) berisi 1 kategori yaitu Berat yang beranggotakan NH dan AB, sedangkan kelas Tidak (0) berisi 2 kategori yaitu kategori Sedang beranggotakan MAL dan BN, kategori Berat beranggotakan KHS dan VN. Untuk kelas Ya (1) memiliki satu kategori sehingga perhitungan ID3 untuk cabang Ya (1) pada atribut P1 berhenti karena sudah tidak dapat dipisah lagi. Sedangkan kelas Tidak (0) memiliki dua kategori sehingga perhitungan ID3 untuk cabang Tidak (0) pada atribut P1 dilanjutkan.

Perhitungan ID3 pada kelas Tidak (0) *node* P1

Tabel berikut adalah data tindak pidana penganiayaan yang masuk pada kelas Tidak (0) *node* P1. Untuk atribut P1 tidak diikuti dalam perhitungan selanjutnya karena atribut P1 telah terpilih sebagai *node* pada perhitungan sebelumnya.

Tabel 3.26 Data Tindak Pidana kelas Tidak *Node* P1

Nama	P3	P5	P6	P8	P10	P11	Hukuman
MAL	0	1	0	0	0	1	Sedang
BN	0	1	0	0	1	0	Sedang
KHS	1	1	0	0	1	1	Berat
VN	1	0	0	1	1	0	Berat

Seperti pada perhitungan ID3 sebelumnya, langkah pertama yaitu mencari nilai *Entropy* total. Perhitungan *Entropy* ini dilakukan terhadap masing-masing atribut yang ada di kelas Tidak (0) *node* P1 menggunakan data pada tabel di atas. Sehingga *Entropy* total yang digunakan adalah nilai *Entropy* dari kelas Tidak (0) *node* P1. $Entropy(s) = 1,0000$.

Perhitungan *Entropy* untuk atribut P3 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) *node* P1:

$$\begin{aligned}
 Entropy(Sya)[S, B] &= - ((2/2)*Log_2((2/2),2)) - 0 \\
 &= - (1)*(Log(2)/Log2) - 0 \\
 &= - (1)*(0) - 0 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 Entropy(Stidak)[S, B] &= - ((2/2)*Log_2((2/2),2)) - 0 \\
 &= - (1)*(Log(2)/Log2) - 0 \\
 &= - (1)*(0) - 0 \\
 &= 0
 \end{aligned}$$



Begitu seterusnya hingga *Entropy* pada atribut P11. Sehingga didapatkan hasil perhitungan *Entropy* semua atribut pada kelas Tidak (1) dan Tidak (0) yang ditunjukkan tabel berikut.

Tabel 3.27 Nilai *Entropy* atribut perhitungan manual tahap VIII

Kriteria	<i>Entropy</i>	
	Ya	Tidak
P3	0,00000	0,00000
P5	0,91830	0,00000
P6	0,00000	1,00000
P8	0,91830	0,00000
P10	0,91830	0,00000

Perhitungan *Information Gain* atribut P3 (Value(P1) = {Ya (1), Tidak (0)}) pada kelas Tidak (0) node P1:

$$\begin{aligned}
 \text{Gain (S, P3)} &= \text{Entropy (S)} - ((2/4) * \text{Entropy}(\text{Sya})[\text{S, B}] \\
 &\quad + (2/4) * \text{Entropy}(\text{Stidak})[\text{S, B}]) \\
 &= 1,0000 - ((0,5)*(0,0000) + (0,5)*(0,0000)) \\
 &= 1,0000 - 0,0000 - 0,0000 \\
 &= 1,0000
 \end{aligned}$$

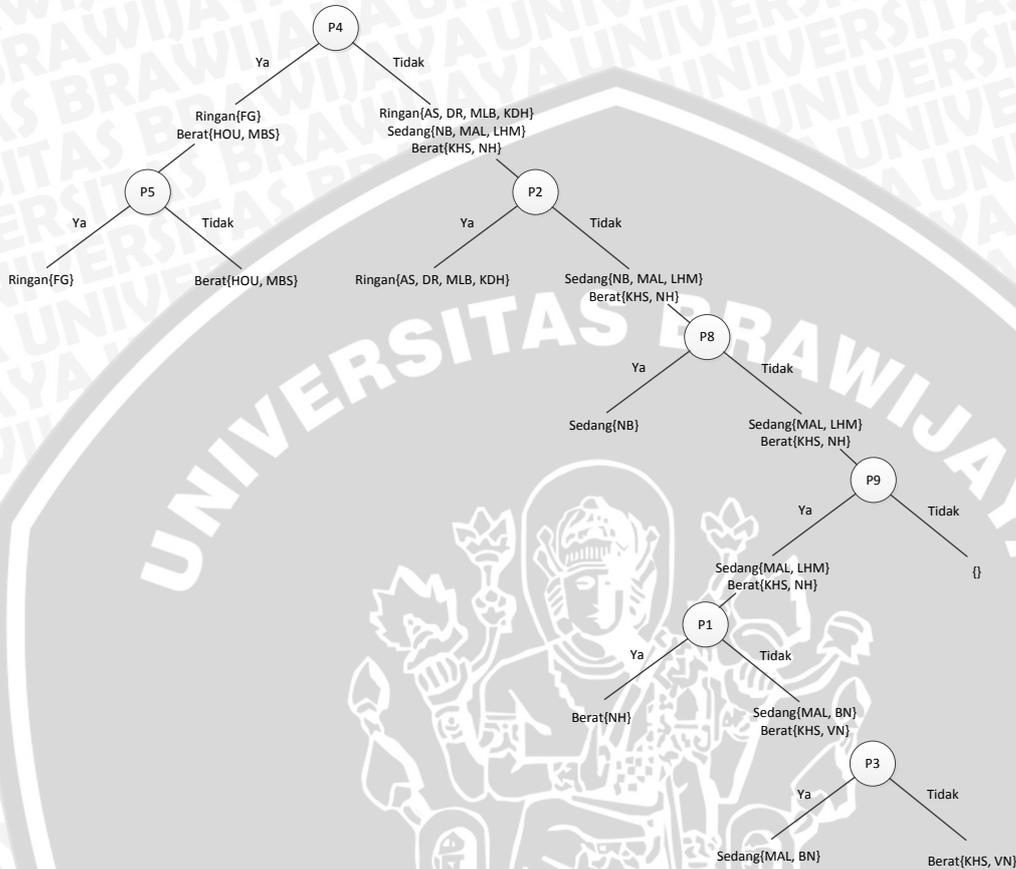
Begitupun seterusnya dilakukan perhitungan *Information Gain* hingga P11. Sehingga hasil *Information Gain* semua atribut yang ada di cabang kelas Tidak (0) node P1 dapat dilihat pada tabel berikut.

Tabel 3.28 Nilai *Information Gain* perhitungan manual tahap VIII

Kriteria	IG(X,S)
P3	1,00000
P5	0,31128
P6	0,66667
P8	0,38780
P10	0,31130
P11	0,00000

Berdasarkan tabel di atas dapat dilihat bahwa nilai *Information Gain* tertinggi yaitu 1,0000. Atribut P3 dipilih sebagai *node* pada cabang kelas Tidak (0)

node P1 karena memiliki nilai *Information Gain* tertinggi. Kemudian data pada kelas Tidak (0) *node* P1 dikelompokkan berdasarkan kelas yang ada pada P3 yaitu Ya (1) dan Tidak (0). Sehingga gambar *tree* yang baru dapat ditunjukkan pada gambar berikut.



Gambar 3.30 *Tree* hasil perhitungan manual tahap VIII

Pada *node* dengan atribut P3 kelas Ya (1) berisi kategori Sedang yang beranggotakan MAL dan BN, sedangkan kelas Tidak (0) berisi kategori Berat yang beranggotakan KHS dan VN. Untuk kedua kelas masing-masing hanya memiliki 1 kategori sehingga perhitungan ID3 telah selesai karena kedua kelas sudah tidak bisa dikelompokkan lagi.

Proses selanjutnya setelah *tree* berhasil memisahkan tiap data ke classnya masing – masing ialah peng-ekstrak-an *Rule*/aturan untuk proses klasifikasi data. *Rule* yang dihasilkan dari *tree* diatas yakni :

1. IF P4 = “Ya” and P5 = “Ya” Then Category = “Ringan”
2. IF P4 = “Ya” and P5 = “Tidak” Then Category = “Berat”
3. IF P4 = “Tidak” and P2 = “Ya” Then Category = “Ringan”
4. IF P4 = “Tidak” and P2 = “Tidak” and P8 = “Ya” Then Category = “Sedang”
5. IF P4 = “Tidak” and P2 = “Tidak” and P8 = “Tidak” and P9 = “Ya” and P1 = “Ya” Then Category = “Berat”
6. IF P4 = “Tidak” and P2 = “Tidak” and P8 = “Tidak” and P9 = “Ya” and P1 =

“Tidak” and P3 = “Ya” Then Category = “Sedang”

- 7. IF P4 = “Tidak” and P2 = “Tidak” and P8 = “Tidak” and P9 = “Ya” and P1 = “Tidak” and P3 = “Tidak” Then Category = “Berat”

Selanjutnya *Rule* tersebut digunakan pada perhitungan data uji yang bertujuan untuk mengetahui keakuratan dari perhitungan yang telah dilakukan pada data training.

3.3.1.1 Blackboard (Daerah Kerja)

Blackboard merupakan area memori yang berfungsi sebagai basis data untuk merekam hasil sementara. *Blackboard* berisi rencana solusi yang berupa data yang digunakan sebagai bahan pertimbangan dalam memberikan kesimpulan akhir. Pada aplikasi Sistem Pakar Penentuan Pasal dan Lama Hukuman untuk permasalahan tindak pidana penganiayaan.

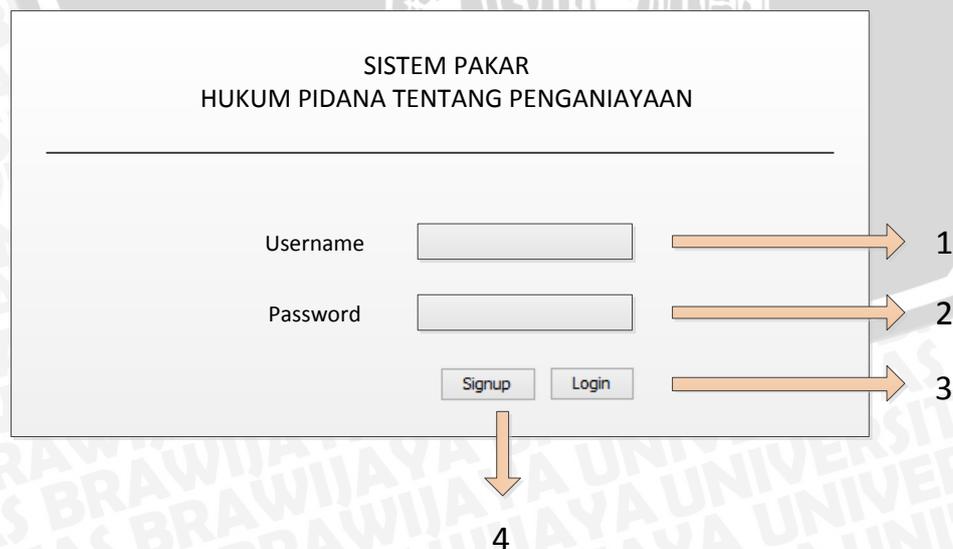
3.3.1.2 Fasilitas Penjelasan

Pada umumnya, fasilitas penjelasan berisi tuntunan penggunaan aplikasi diagnosa penyakit kulit pada anak. Fasilitas penjas yang akan diberikan dalam aplikasi anak ini yaitu penjelasan tentang informasi kegunaan aplikasi dan penjelasan informasi cara penggunaan aplikasi Fasilitas penjelasan ini penting untuk memberikan informasi kepada para pengguna mengenai manfaat dari aplikasi ini dan bagaimana cara penggunaannya.

3.3.1.3 Antarmuka Pengguna

Perancangan subsistem ini bertujuan agar *user* (pengguna) dapat berkomunikasi dan memberikan perintah pada sistem. Perancangan antarmuka dalam sistem ini akan dijelaskan melalui alur *sitemap* dan desain antarmuka tiap halaman. Perancangan ini dibuat untuk mewakili keadaan yang sebenarnya ketika aplikasi ini dibuat.

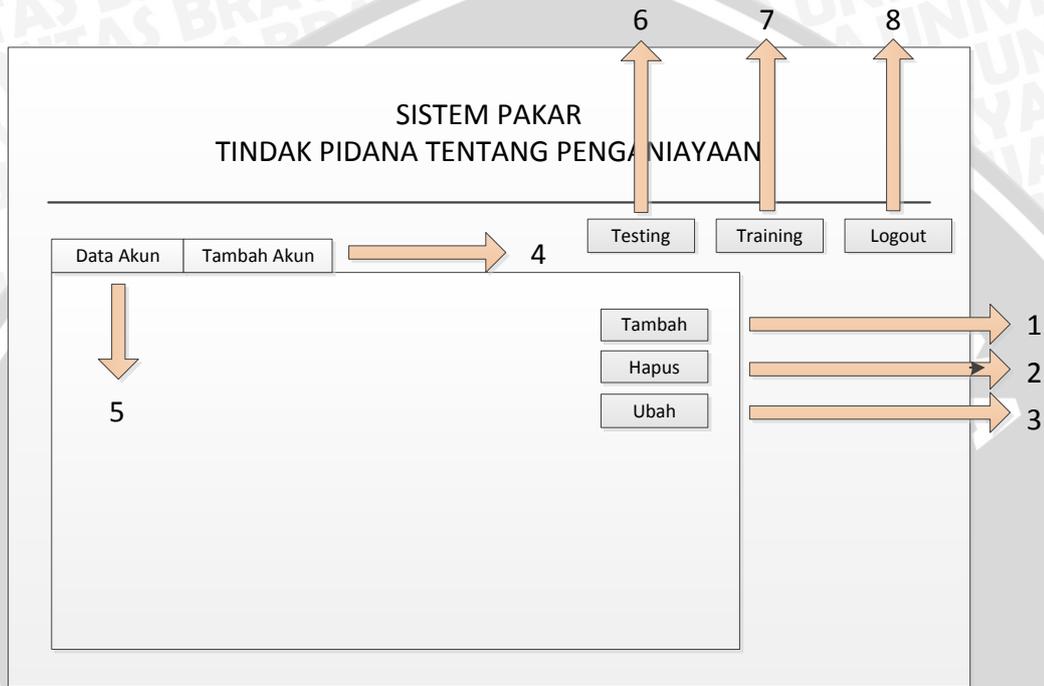
Halaman Login



Keterangan:

1. Text field untuk Username
2. Text field untuk Password
3. Button untuk Login
4. Button untuk Sign up

Halaman Admin

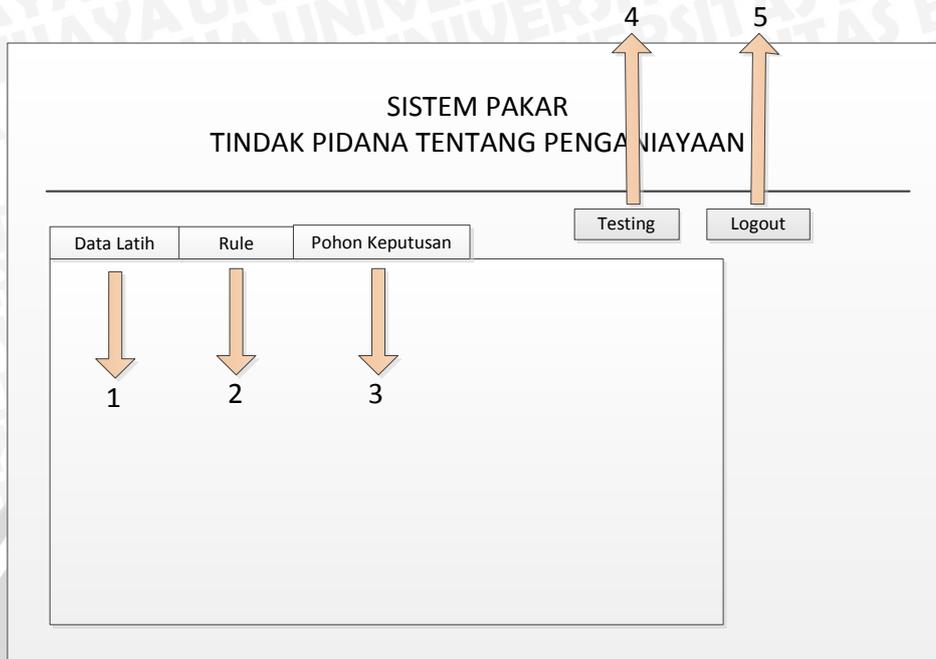


Keterangan:

1. Tombol untuk menambah pengguna
2. Tombol untuk mengubah data pengguna.
3. Tombol untuk menghapus pengguna.
4. Tab untuk tambah pengguna
5. Tab untuk menampilkan data pengguna
6. Tombol menuju ke halaman testing untuk pengujian sistem
7. Tombol menuju ke halaman training untuk pelatihan data

Halaman Pakar/Hakim

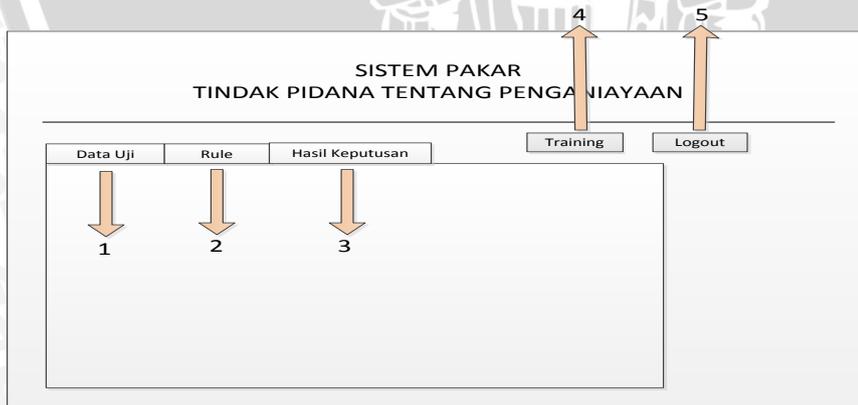
Halaman data training



Keterangan:

1. Tab untuk menampilkan data latih
2. Tab untuk menampilkan *Rule* berdasarkan hasil perhitungan data latih
3. Tab untuk menampilkan pohon keputusan
4. Tombol untuk beralih ke halaman data testing
5. Tombol untuk keluar dari akun

Halaman data testing

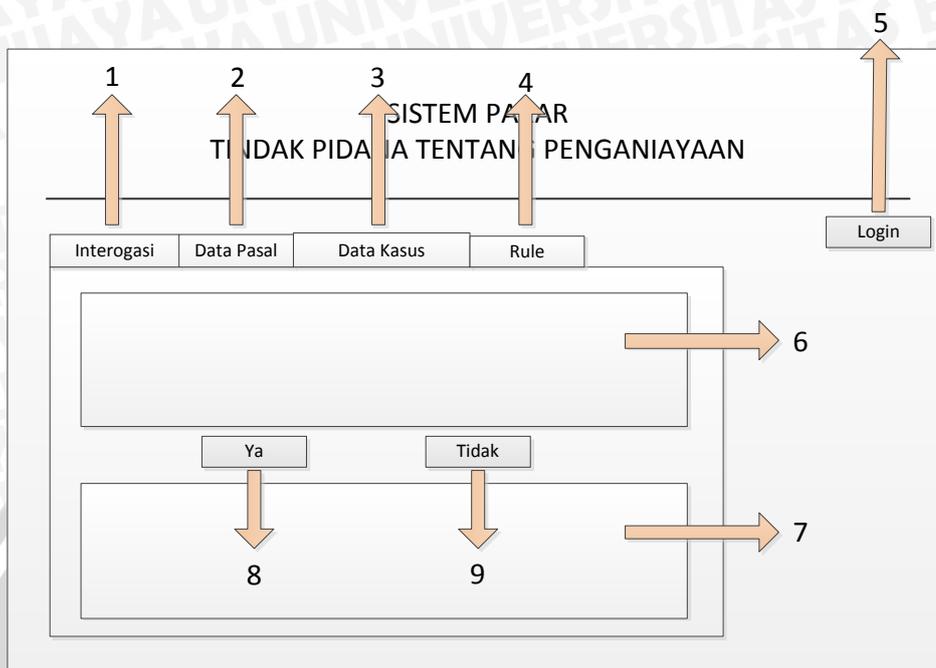


Keterangan:

1. Tab untuk menampilkan data uji
2. Tab untuk menampilkan *Rule* berdasarkan hasil perhitungan data latih
3. Tab untuk menampilkan hasil keputusan
4. Tombol untuk beralih ke halaman data training

5. Tombol untuk keluar dari akun

Halaman Masyarakat atau Pelaku



Keterangan:

1. Tab untuk menampilkan halaman interogasi berisi pertanyaan tentang kasus tindak pidana yang akan dijawab dengan pertanyaan ya atau tidak
2. Tab untuk menampilkan data pasal
3. Tab untuk menampilkan data kasus
4. Tab untuk menampilkan *Rule* hasil dari perhitungan data training
5. Tombol untuk beralih ke login
6. Text box berisi pertanyaan interogasi
7. Text box hasil dari keputusan
8. Tombol jawaban ya
9. Tombol jawaban tidak



BAB 4 IMPLEMENTASI

Pada bab empat menjelaskan tentang implementasi program sistem pakar penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan. Pembahasan implementasi tersebut terdiri dari spesifikasi perangkat lunak dan perangkat keras, batasan sistem, implementasi algoritma dan implementasi antarmuka pengguna.

4.1 Spesifikasi Perangkat Lunak dan Perangkat Keras

Pada sub bab ini membahas tentang perangkat keras dan perangkat lunak yang digunakan untuk membuat aplikasi Sistem Pakar Penentuan Pasal dan Lama Hukuman Untuk Permasalahan Tindak Pidana Penganiayaan. Perangkat lunak yang digunakan adalah sebagai berikut:

- Sistem operasi Windows 7 Home Premium (64 Bit)
- netBeans IDE 8.0
- PhpMyadmin dengan XAMPP 3.2.1

Sedangkan perangkat keras yang digunakan *notebook* VAIO VPCEA16FG dengan spesifikasi sebagai berikut:

- CPU *processor Intel*® Core™ i5-520M @ 2.40GHz
- RAM *memory* 4 GB
- Memory Storage* 500 GB

4.2 Implementasi Program

Berdasarkan pada bab 3 yaitu analisa dan perancangan sistem, maka pada subbab ini berisi mengenai implementasi program yang dirancang menggunakan bahasa pemrograman Java. Jalan awal program dimulai dengan input data kasus kemudian dilanjutkan dengan perhitungan ID3, dengan cara pembacaan inputan data yang berupa *file .xls*, kemudian perhitungan nilai *Entropy*, perhitungan nilai *Information Gain*, pembentukan *tree*, dan menghasilkan *Rule* yang digunakan untuk menentukan lama hukuman yang didapatkan. Penjelasan lebih lanjut akan dijelaskan dalam subbab dibawah ini.

4.2.1 Pembacaan Data

Dalam penelitian ini sistem dalam menerima inputan berupa data dengan format *.xls*. pembacaan data dilakukan pada tiap-tiap kolom pada baris pertama hingga terakhir, hasil dari pembacaan tersebut kemudian disimpan kedalam variabel *array*. Data dalam *array* tersebut diantaranya adalah No urut, Nama Inisial, nilai kriteria dan kategori.

```
ArrayList<Data> data;  
    Boolean isColumnName = jCBColName.isSelected();
```

```

        if(jTFPathFile.getText().endsWith(".txt"))
            data=
Core.FileProcessing.bacaFile(jTFPathFile.getText(),
isColumnName);
        else if(jTFPathFile.getText().endsWith(".xls")){
            data=
Core.FileProcessing.bacaFiles(jTFPathFile.getText(),
isColumnName);
        }
        else if(jTFPathFile.getText().isEmpty()){
            JOptionPane.showMessageDialog(this, "File belum
dipilih", "Error", JOptionPane.ERROR_MESSAGE);
            return ;
        }
        else{
            JOptionPane.showMessageDialog(this, "Format tidak
mendukung", "Error", JOptionPane.ERROR_MESSAGE);
            return ;
        }
        Object isiTable[][] = new Object[data.size()][];
        int j = 0;
        for (Data a : data) {
            isiTable[j] = a.getIsiTable();
            j++;
        }
        jTable1.setModel(new
javax.swing.table.DefaultTableModel(
            isiTable,
            new String[]{"No", "Nama Inisial", "P1", "P2",
"P3", "P4", "P5", "P6", "P7", "P8", "P9", "P10", "P11",
"Alternatif"}
        ));
        frame.setTree(new ID3(kriteria));
        frame.getTree().prosesID3(data);
        double gapBetweenLevels = 50;
        double gapBetweenNodes = 200;
        DefaultConfiguration<TextInBox> configuration = new
DefaultConfiguration<TextInBox>(
            gapBetweenLevels, gapBetweenNodes,
Configuration.Location.Top);
        TextInBoxNodeExtentProvider nodeExtentProvider = new
TextInBoxNodeExtentProvider();

```

```

        TreeLayout<TextInBox>          treeLayout          =          new
TreeLayout<TextInBox>(getSampleTree(frame.getTree().getRoot()),
                                nodeExtentProvider,
configuration,true);

        TextInBoxTreePane              panel              =          new
TextInBoxTreePane(treeLayout);

        setGraph();

        setRule(frame.getTree().getRoot());

```

Source Code 4.1 Fungsi Pembacaan File

4.2.2 Perhitungan *Entropy*

Berikut adalah implementasi perhitungan *Entropy* yang digunakan pada implementasi algoritma *ID3*. Proses awal dari pembentukan *tree* ini dimulai dari perhitungan *Entropy*. Pada perhitungan ini diawali dengan perhitungan *Entropy* total, kemudian mencari *Entropy* nilai dalam atribut dan terakhir yaitu menghitung *Entropy* dari masing-masing atribut. Pada perhitungan *Entropy* total dimulai dengan menghitung banyak data dan mengelompokkan berdasarkan kelasnya data tersebut lalu disimpan dalam variabel *anggotaKelas*. Dan dihitung menggunakan persamaan 2.1. proses yang sama digunakan untuk mendapatkan *Entropy* dari tiap tiap atribut. Proses perhitungan *Entropy* total, *Entropy* nilai dalam atribut dan *Entropy* tiap atribut berturut-turut ditunjukkan pada *source code* 4.2, 4.3.

```

public double getEntropTotal(ArrayList<Data> data) {
    double Entrop = 0;
    HashMap<String,Integer> anggotaKelas = new
HashMap<String,Integer>();
    for (int i = 0; i < data.size(); i++) {

if(!anggotaKelas.containsKey(data.get(i).kelas)){//menggunakan
hashmap, containskey mksdnya mengecek data itu class apa, kalau
uda ada dimasukin ke array, klo blm ada buat class baru
        anggotaKelas.put(data.get(i).kelas, 1);
    }else{
        int a = anggotaKelas.get(data.get(i).kelas);
        a++;
        anggotaKelas.put(data.get(i).kelas, a);
    }

    System.out.println("Data get: " + data.get(i).kelas +
" - Key: " + anggotaKelas.get(data.get(i).kelas));
    }
    System.out.println("Data Size: " + data.size());
    for (String a : anggotaKelas.keySet()) {

```

```

        if (anggotaKelas.get(a) != 0) {
            Entrop += (-
1 * (((double) anggotaKelas.get(a) / (double) data.size()) * log2((double)
anggotaKelas.get(a) / (double) data.size())));
            System.out.println("Entrop      +=      -
1*" + anggotaKelas.get(a) + "/" + data.size() + "*log2" + anggotaKelas.get
(a) + "/" + data.size());
        }
    }
    System.out.println("Entropi total: " + Entrop);
    return Entrop;
}

```

Source code 4.2: perhitungan *Entropy* total

```

public double[] getEntropKriteria(ArrayList<Data> data, int
kriteria) {
    HashMap<String, Integer> anggotaKelas = new
HashMap<String, Integer>();
    for (int i = 0; i < data.size(); i++) {
        if (!anggotaKelas.containsKey(data.get(i).kelas)) {
            anggotaKelas.put(data.get(i).kelas, 1);
        } else {
            int a = anggotaKelas.get(data.get(i).kelas);
            a++;
            anggotaKelas.put(data.get(i).kelas, a);
        }
    }
    double EntropiKriteri [] = new double[] {
        getE(data, kriteria, 1, anggotaKelas.keySet()),
        getE(data, kriteria, 0, anggotaKelas.keySet())
    };
    return EntropiKriteri;
}

```

Source Code 4.3 : perhitungan *Entropy* Atribut

4.2.3 Perhitungan *Information Gain*

Nilai *Entropy* yang didapat pada proses perhitungan sebelumnya dijadikan inputan pada perhitungan *Information Gain* menggunakan persamaan 2.2. Hasil perhitungan *information gain* ini nantinya digunakan untuk menentukan atribut sebagai *node*. Perhitungan ini dilakukan sebanyak atribut sehingga menggunakan perulangan seperti yang ditunjukkan pada *source code* 4.4.

```

public double [] getIgain(ArrayList<Data> data) {
    double []Iganin=new double[banyak_kriteria];
    double entropiTotal = getEntropiTotal(data);
    for (int i = 0; i <Iganin.length; i++) {
        double [] entropiKriteria =
getEntropiKriteria(data,i);
        System.out.println("entropi          kriteria:
"+entropiKriteria[0]+"-"+entropiKriteria[1]);
        Iganin[i]=(entropiTotal-(entropiKriteria[0])-
(entropiKriteria[1]));
    }
    return Iganin;
}

```

Source code 4.4: Perhitungan Information Gain

4.2.4 GetRoot

Pada proses ini dipilih satu atribut untuk dijadikan *root*. Atribut yang memiliki *Information Gain* yang tertinggi merupakan atribut yang terpilih sebagai *root*. Implementasi proses pemilihan atribut sebagai *root* ditunjukkan pada *source code 4.5*.

```

public int getRoot(double [] igain,ArrayList<Data> data,int []
cek){
    double max=0;
    int index=0;
    boolean al = false;
    for(int i=0;i<igain.length;i++){
        if(cek[i]!=0)
            continue;
        if(igain[i]>max){
            max = igain[i];
            index = i;
            al = true;
        }
    }
    if(!al){
        return -1;
    }
    return index;
}

```

Source code 4.5: GetRoot

4.2.5 Gambar Tree

Pada proses ini method perhitungan *Information Gain* dan *method GetRoot* digunakan untuk membentuk *Decision Tree* berdasarkan data inputan. Kedua proses tersebut dilakukan hingga percabangan *tree* sampai pada *leaf*, ketika *tree* sudah terbentuk maka hasil akhir yang didapatkan adalah *Rule*. *Rule* tersebut disimpan dalam *database* yang nantinya digunakan untuk menampilkan alternatif makanan dan pengujian data. *Source code* 4.6 menjelaskan proses pembentukan *tree*.

```
public void prosesID3(ArrayList<Data> data) {
    Stack<Node>jalanCrita = new Stack<Node>();
    int [] newCek = new int [banyak_kriteria];
    double [] IgainRoot = getIgain(data);
    int idKriteria = getRoot(IgainRoot,data,newCek);
    newCek[idKriteria]=1;
    Node root = new Node(idKriteria, data,IgainRoot);
    root.setCeks(newCek);
    this.root = root;
    jalanCrita.push(root);
    while(!jalanCrita.isEmpty()){
        Node temp = jalanCrita.pop();
        if(temp.isLeaf )continue;
        for(int i=0;i<2;i++){
            ArrayList<Data> dataBagi = new ArrayList<Data>();
            for(Data d : temp.data){
                if(d.kriteria.get(temp.id)==i){
                    dataBagi.add(d.clone());
                }
            }
            if(dataBagi.size()!=0){
                Node anakTemp;
                Set<String> sekKelas =
                apakahSamaKelas(dataBagi);
                if(sekKelas.size()<=1){
                    anakTemp = new Node(dataBagi);
                    for(String f : sekKelas)
                        anakTemp.setLeaf (f);
                }else{
                    IgainRoot = getIgain(dataBagi);
```



```

        objKasus.iKasusYaTidak = new
        ArrayList<Integer[]>();
        nextKasus = sqlLogic.getNextKasus(objKasus);
        jTAKasus.setText(nextKasus.get(1).toString());
        jBtnMulai.setEnabled(false);
        jBtnYa.setEnabled(true);
        jBtnTidak.setEnabled(true);
        firstTime = true;
        bLamaHukuman = false;
        jTAHasil.setText("");
    }
}

Private void
jBtnYaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    sqlLogic = new ToSqlLogic();
    if(!bLamaHukuman){
        if(!dataKasus.isEmpty()){
            if(firstTime){
                firstTime = false;
            }
            objKasus.iKasusYaTidak.add(new
            Integer[]{Integer.parseInt(nextKasus.get(0).toString()), 1});
            nextKasus = sqlLogic.getNextKasus(objKasus);

            if(!nextKasus.get(0).toString().equals("notfound")){
                objKasus.seq_id =
                Integer.parseInt(nextKasus.get(nextKasus.size()-1).toString());
                System.out.println("seq_id: " +
                objKasus.seq_id);
                iKasus++;
            }
            if(!nextKasus.isEmpty()){
                if(nextKasus.get(0).toString().equals("found")){
                    setStatementPasal();
                    int iOption =
                    JOptionPane.showConfirmDialog(null, "Apakah anda ingin
                    melanjutkan proses menentukan lamanya hukuman?", "Menentukan lama
                    hukuman", JOptionPane.YES_NO_OPTION);
                    atributTemp = new ArrayList<Object[]>();
                    if(iOption == 0){

```

```

        atributTemp = sqlLogic.getAtributAll();
        if(!atributTemp.isEmpty()){
            jTAHasil.setText("");
            bLamaHukuman = true;
            jTAKasus.setText(atributTemp.get(indxKasusHukuman)[2].toString());
            jBtnMulai.setEnabled(false);
            jBtnYa.setEnabled(true);
            jBtnTidak.setEnabled(true);
        }
    }
    else
    if(!nextKasus.get(0).toString().equals("notfound")){
        jTAKasus.setText(nextKasus.get(1).toString());
        }
        else{
            JOptionPane.showMessageDialog(null,
            "Tidak ada pasal yang cocok untuk menjerat kasus anda!");
            jTAKasus.setText("");
            jBtnMulai.setEnabled(true);
            jBtnYa.setEnabled(false);
            jBtnTidak.setEnabled(false);
        }
    }
}
}
else{
    System.out.println("index: " + indxKasusHukuman);
    if(atributTemp.size() - indxKasusHukuman >= 1){
        Object atributID = atributTemp.get(indxKasusHukuman)[0];
        Object atributCode = atributTemp.get(indxKasusHukuman)[1];
        Object atributDesc = atributTemp.get(indxKasusHukuman)[2];
        atributTemp.set(indxKasusHukuman, new
        Object[]{atributID, atributCode, atributDesc, 1});
        indxKasusHukuman++;
    }
}
}
}

```

```

1) {
    if(atributTemp.size() - indxKasusHukuman >=
        jTAKasus.setText("Apakah " +
atributTemp.get(indxKasusHukuman)[2].toString() + "?");
    }
    else{
        System.out.println("atributTemp: " +
atributTemp.size());
        tampilAlternatif(atributTemp);
        jTAKasus.setText("");
        jBtnMulai.setEnabled(true);
        jBtnYa.setEnabled(false);
        jBtnTidak.setEnabled(false);
    }
}
else{
    tampilAlternatif(atributTemp);
    jTAKasus.setText("");
    jBtnMulai.setEnabled(true);
    jBtnYa.setEnabled(false);
    jBtnTidak.setEnabled(false);
}
}
}
private void
jBtnTidakActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    sqlLogic = new ToSqlLogic();
    if(!lLamaHukuman){
        if(!dataKasus.isEmpty()){
            if(firstTime){
                firstTime = false;
            }
            objKasus.iKasusYaTidak.add(new
Integer[]{Integer.parseInt(nextKasus.get(0).toString()), 2});
            nextKasus = sqlLogic.getNextKasus(objKasus);
        }
        if(!nextKasus.get(0).toString().equals("notfound")){
            objKasus.seq_id =
Integer.parseInt(nextKasus.get(nextKasus.size()-1).toString());

```

```

objKasus.seq_id);
        System.out.println("seq_id: " +
        iKasus++;
    }
    if(!nextKasus.isEmpty()){
if(nextKasus.get(0).toString().equals("found")){
        setStatementPasal();
        int iOption =
JOptionPane.showConfirmDialog(null, "Apakah anda ingin
melanjutkan proses menentukan lamanya hukuman?", "Menentukan lama
hukuman", JOptionPane.YES_NO_OPTION);
        atributTemp = new ArrayList<Object[]>();
        if(iOption == 0){
            atributTemp =
sqlLogic.getAtributAll();
            if(!atributTemp.isEmpty()){
hasil //jTAHasil.setText(""); tampil
                bLamaHukuman = true;
jTAKasus.setText(atributTemp.get(indxKasusHukuman)[2].toString()
);
                jBtnMulai.setEnabled(false);
                jBtnYa.setEnabled(true);
                jBtnTidak.setEnabled(true);
            }
        }
        else
if(!nextKasus.get(0).toString().equals("notfound")){
jTAKasus.setText(nextKasus.get(1).toString());
        }
        else{
            JOptionPane.showMessageDialog(null,
"Tidak ada pasal yang cocok untuk menjerat kasus anda!");
            jTAKasus.setText("");
            jBtnMulai.setEnabled(true);
            jBtnYa.setEnabled(false);
            jBtnTidak.setEnabled(false);
        }
    }
}

```


4.2.7 Pencarian Lama Hukuman

Pada proses ini menjelaskan bagaimana mendapatkan lama hukuman hasil dari data pakar menggunakan proses ID3. User diminta untuk memasukkan masukan sesuai dengan data yang diperoleh dari pakar. Nilai-nilai tersebut dicocokkan dengan *Rule* yang telah disimpan di *database*. Kemudian tampil salah satu *Rule* yang sesuai dengan nilai masing-masing atribut. Proses menampilkan lama hukuman ditunjukkan pada source code 4.8.

```
public void tampilAlternatif(ArrayList<Object[]> dataAtribut){
    System.out.println("dataAtribut: " +
dataAtribut.get(0)[3]);
    ArrayList<Integer> kriteria = new ArrayList<Integer>();
    ArrayList<Object[]> dataRule = new ArrayList<Object[]>();
    int nilai = 0;
    sqlLogic = new ToSqlLogic();
    int RuleCount = sqlLogic.getRuleCount();
    int countNode = 1;
    String Rule = "";
    String atribut = "";
    String nilaiAtrbt = "";
    String alternatif = "";
    String altID = "";
    int iNilaiAtrbt = 0;
    int atrbtID = 0;
    for(int i=0; i<RuleCount; i++){
        dataRule = sqlLogic.getRule(i+1);
        countNode = 0;
        Rule = "";
        for(int j=0; j<dataRule.size(); j++){
            atribut = dataRule.get(j)[3].toString();
            nilaiAtrbt = dataRule.get(j)[5].toString();
            iNilaiAtrbt
Integer.parseInt(dataRule.get(j)[4].toString());
            atrbtID
Integer.parseInt(dataRule.get(j)[2].toString());
            //System.out.println("-----"+atribut+"-
"+nilaiAtrbt+"-
"+kriteria.get(0)+kriteria.get(1)+kriteria.get(2)+kriteria.get(3
));
            loopFor2:{
                for(int k=0; k<dataAtribut.size(); k++){
                    System.out.println("sinisini");
```

```

if(Integer.parseInt(dataAtribut.get(k)[0].toString()) == atrbtID
&& Integer.parseInt(dataAtribut.get(k)[3].toString()) ==
iNilaiAtrbt){
    String nilaiAtribut = "";
    if(iNilaiAtrbt == 1)
        nilaiAtribut = "Ya";
    else
        nilaiAtribut = "Tidak";
    Rule = Rule + (countNode+1) + ". Jika "
+ dataAtribut.get(k)[2].toString() + " = '"+nilaiAtribut+"' \n";
    countNode++;
    break loopFor2;
}
}
}
}
if(countNode == dataRule.size()){
    Boolean b356 = false;
    ArrayList<Integer> nHukuman = new
ArrayList<Integer>();
    double totalHukuman = 0;
    double nDHukuman = 0.0;
    for(int p=0; p<hukuman.size(); p++){
        if(hukuman.get(p)[0].toString().equals("356")){
            b356 = true;
            nDHukuman =
Double.parseDouble(hukuman.get(p)[1].toString());
        }
        else{
            nHukuman.add(Integer.parseInt(hukuman.get(p)[1].toString()));
            totalHukuman = totalHukuman +
Double.parseDouble(hukuman.get(p)[1].toString());
        }
    }
    if(b356){
        totalHukuman = totalHukuman + (totalHukuman *
nDHukuman);
    }
}
}
}
}

```

```

        double minHukuman = Math.round(totalHukuman *
Double.parseDouble(dataRule.get(0)[9].toString()) / 100);

        double maxHukuman = Math.round(totalHukuman *
Double.parseDouble(dataRule.get(0)[10].toString()) / 100);

        totalHukuman = Math.round(totalHukuman);
        double bulanTotal = totalHukuman % 12;
        double bulanMin = minHukuman % 12;
        double bulanMax = maxHukuman % 12;
        totalHukuman = (totalHukuman - bulanTotal)/12;
        minHukuman = (minHukuman - bulanMin)/12;
        maxHukuman = (maxHukuman - bulanMax)/12;

        String sBulanTotal = (bulanTotal == 0.0 ? "" :
Math.round(bulanTotal) + " bulan");
        String sBulanMin = (bulanMin == 0.0 ? "" :
Math.round(bulanMin) + " bulan");
        String sBulanMax = (bulanMax == 0.0 ? "" :
Math.round(bulanMax) + " bulan");

        Rule = Rule + "\nHukuman maksimal sesuai pasal:
"+ Math.round(totalHukuman) + " tahun "+ sBulanTotal
        + "\nKategori hukuman: " +
dataRule.get(0)[7] + "\nAncaman hukuman penjara: minimal "+
Math.round(minHukuman) + " tahun "+ sBulanMin
        + " hingga maksimal " +
Math.round(maxHukuman) + " tahun "+ sBulanMax + "\n\n";

        alternatif = dataRule.get(0)[7].toString();
        altID = dataRule.get(0)[8].toString();
        i = RuleCount;
    }
}

System.out.println("rrrrrruuuuuuuuuuuuuuulllllllllll-
"+Rule);

Rule = "\nDengan pertimbangan sebagai berikut:\n" + Rule;
jTAHasil.setText(setPasal + "\n" + Rule);
}

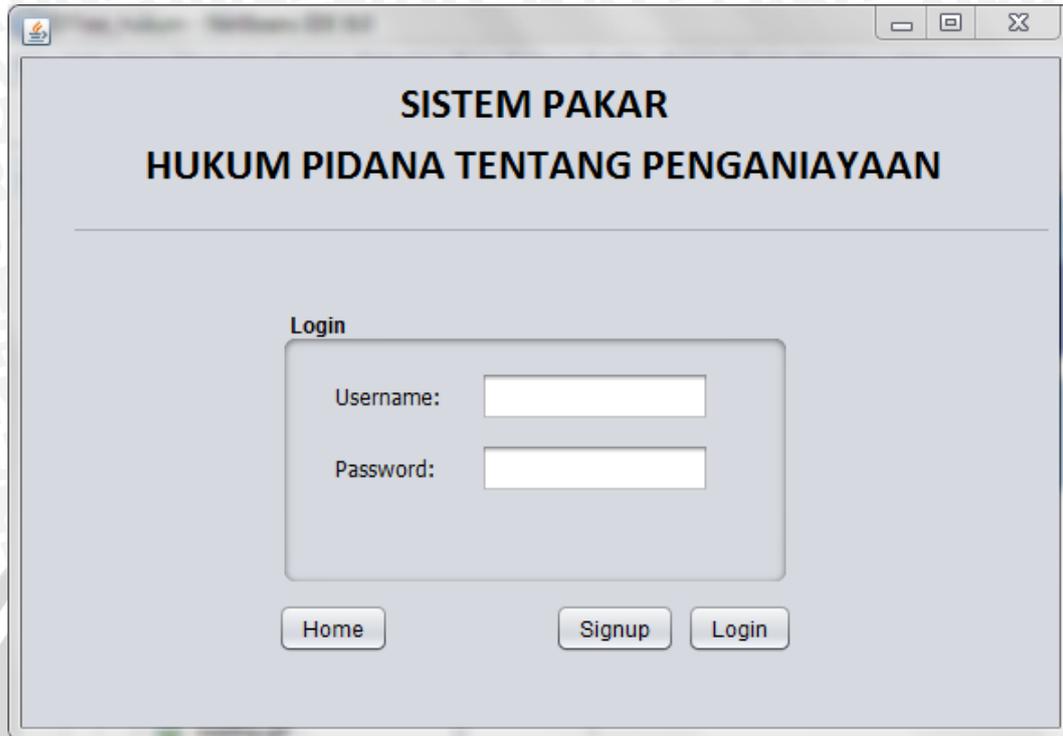
```

Source code 4.8 : Pencarian Lama Hukuman

4.3 Implementasi Antarmuka

Pada bab ini merupakan implementasi antarmuka sistem pakar penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan. Berikut merupakan antarmuka sistem sistem pakar penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan.

4.3.1 Login



SISTEM PAKAR
HUKUM PIDANA TENTANG PENGANIAYAAN

Login

Username:

Password:

Gambar 4.1 Antarmuka Login

Implementasi antarmuka login berisi tombol *login* dan *signup*. Tombol *login* digunakan untuk akun admin, pakar atau hakim. Tombol *signup* untuk mendaftar bila belum memiliki akun.

4.3.2 Antarmuka Pakar



Aplikasi Saya

> >

Baris pertama adalah nama kolom

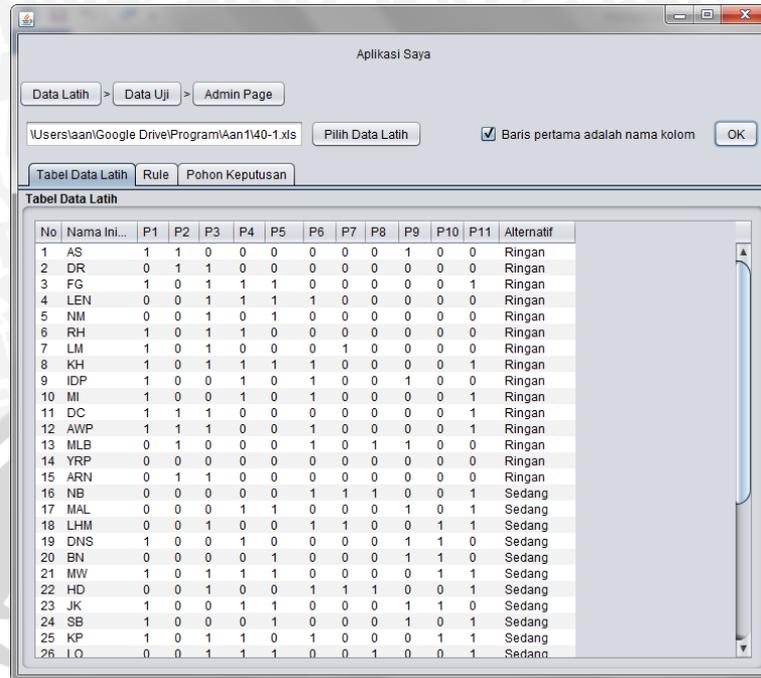
Tabel Data Latih

No	Nama	IND-1	IND-2	IND-3	IND-4	IND-5	IND-6	IND-7

Gambar 4.2 Antarmuka Pakar

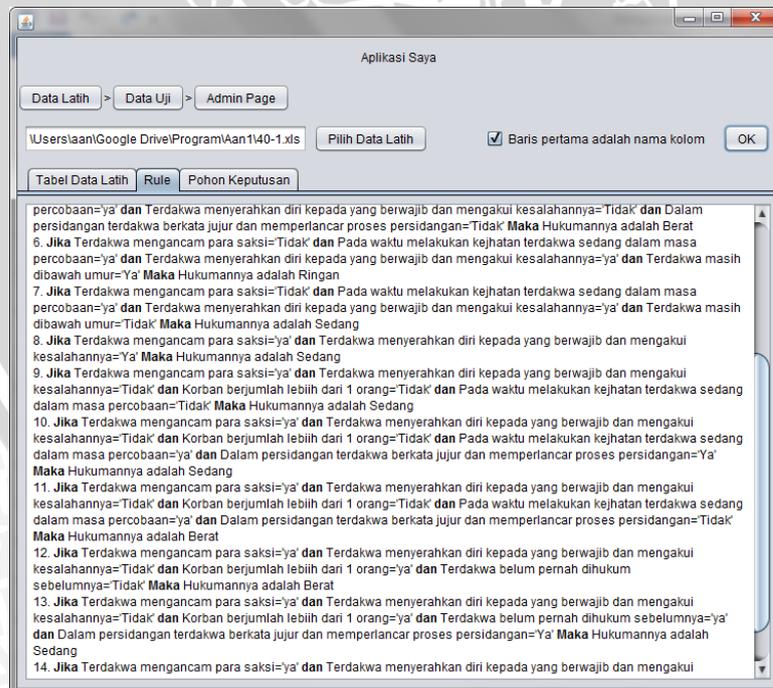
Setelah pakar atau hakim melakukan menekan tombol *login*, kemudian mengarah pada halaman *training*. Halaman ini berisi proses perhitungan metode ID3 yang menghasilkan *Rule* untuk menentukan lama hukuman.

4.3.3 Antarmuka Training

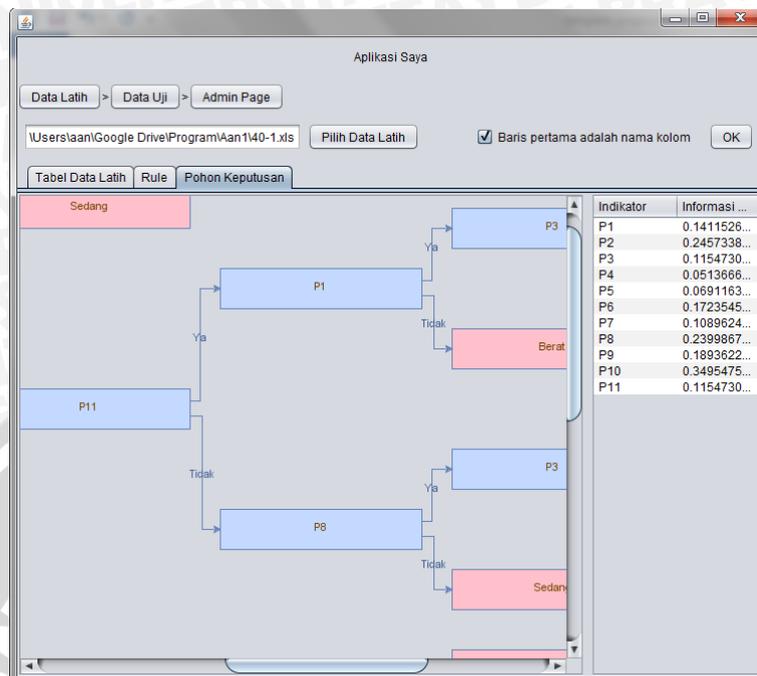


Gambar 4.3 Antarmuka Training

Antarmuka training menampilkan proses perhitungan ID3 tree. admin memilih file data latih kemudian menekan tombol OK untuk melakukan perhitungan ID3, antarmuka training terdapat tabel data latih, Rule dan pohon keputusan.



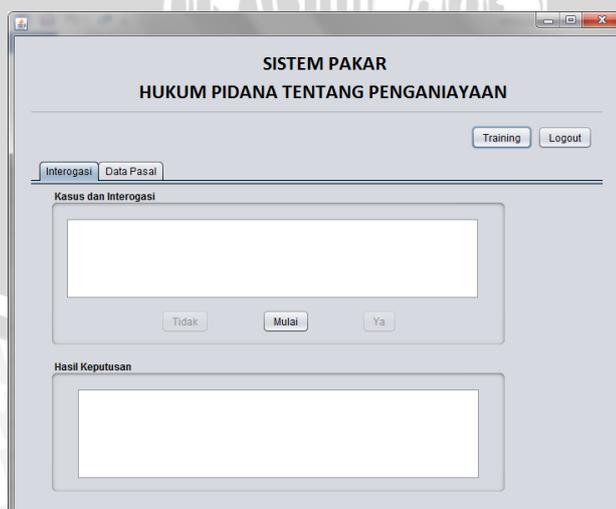
Gambar 4.4 Antarmuka Rule



Gambar 4.5 Antarmuka Pohon Keputusan

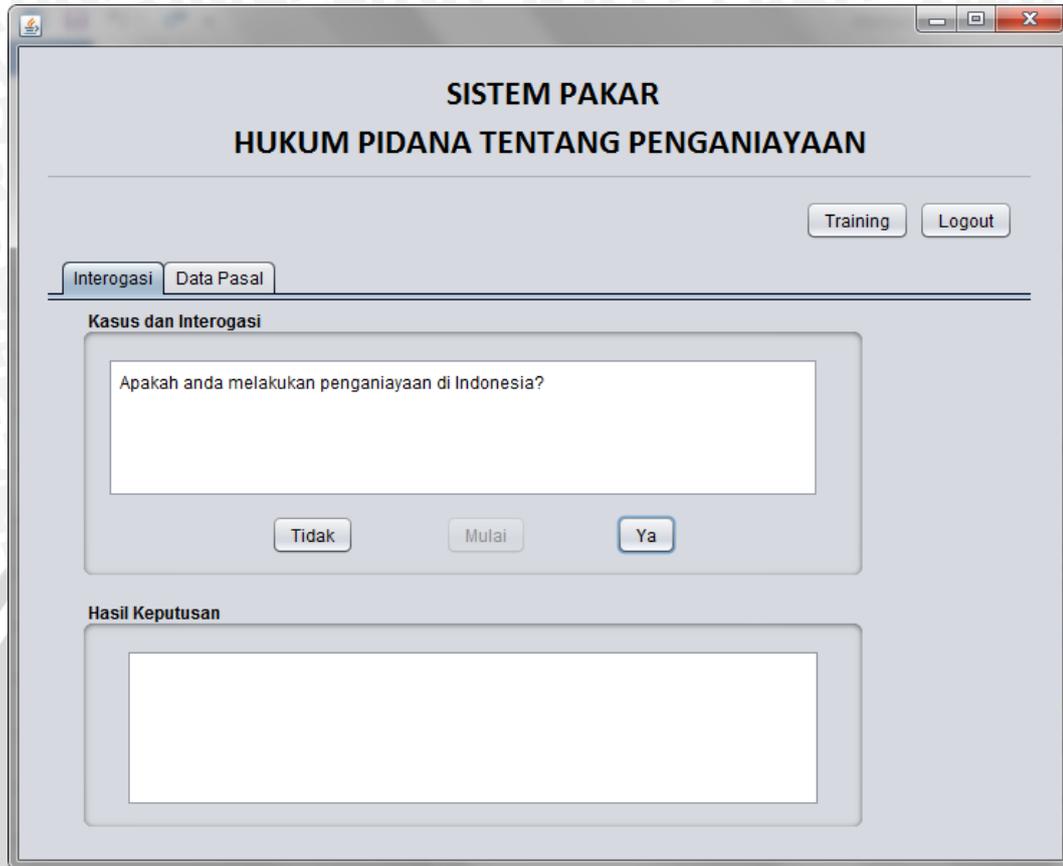
4.3.4 Homepage

Implementasi antarmuka *Home page* berisi tombol *interogasi*, *data pasal*, *training*, *logout* dan *mulai*. Tombol *interogasi* untuk memulai interogasi berupa menjawab pertanyaan dengan menekan jawaban ya atau tidak sesuai dengan tombol ya dan tidak yang disediakan. Tombol *data pasal* berisi tentang penjelasan pasal-pasal yang ada dalam bab penganiayaan. Tombol *training* untuk memasukkan data training guna membentuk *Rule* baru. Kemudian tombol *logout* untuk *logout*, tombol ini tidak tampil ketika sebelumnya tidak ada *login*.



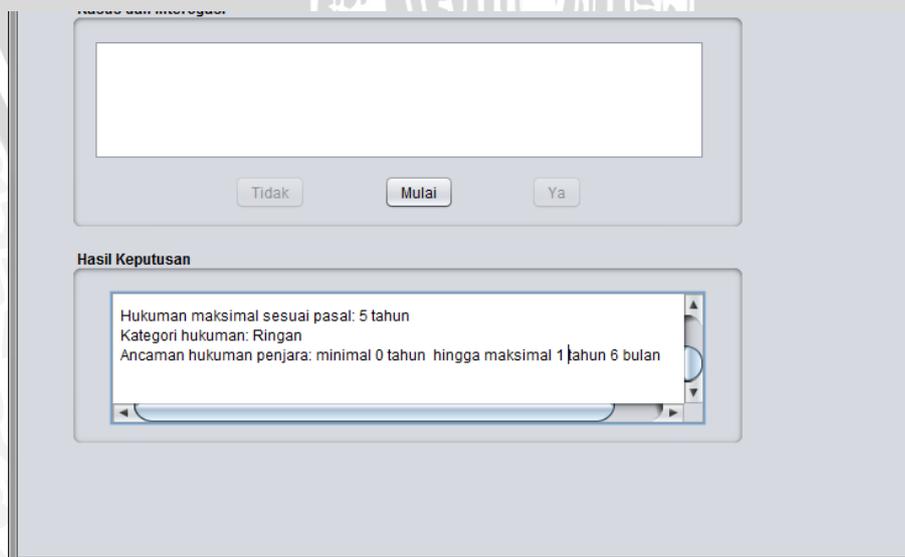
Gambar 4.6 Antarmuka homepage

4.3.5 Proses Interogasi



Gambar 4.7 Proses Interogasi

Antarmuka proses interogasi menampilkan pertanyaan-pertanyaan kasus yang dilanggar. Di antarmuka proses interogasi terdapat tombol *ya* dan *tidak*, ketika *user* menekan selesai melakukan proses interogasi maka akan menampilkan hasil keputusan lama hukuman dan pasal yang dilanggar seperti pada gambar 4.8.



Gambar 4.8 Hasil Keputusan

BAB 5 PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian dan analisis terhadap sistem pakar penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan. Pengujian yang dilakukan untuk penelitian ini adalah pengujian akurasi. Pengujian akurasi dilakukan dengan cara memeriksa akurasi dari output apakah sesuai atau tidak.

5.1 Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan cara menguji apakah sistem bisa memberikan hasil sesuai rancangan dan desain yang telah dibuat. Pengujian yang dilakukan adalah:

1. Melakukan login sebagai admin/hakim.
2. Menambah, menampilkan, menghapus dan mengubah data user.
3. Memasukkan data latih dengan format .xls
4. Memproses data untuk menghasilkan *Rule* dengan menggunakan metode Iterative Dichotomiser *Tree*.
5. Menampilkan vonis berupa pasal dan lama hukuman

Analisa hasil pengujian :

1. Sistem berhasil melakukan login dengan baik sebagai, admin/hakim.
2. Sistem dapat melakukan penambahan, menampilkan, menghapus dan mengubah data user.
3. Sistem berhasil melakukan proses import file dengan format .xls
4. Dengan perhitungan metode Iterative Dichotomiser *Tree*, sistem dapat menghasilkan *Rule* berdasarkan data latih yang ada.
5. Sistem berhasil menampilkan vonis berupa pasal dan lama hukuman sesuai dengan perhitungan metode.

Tabel 5.1 hasil pengujian fungsionalitas

No	Kasus Uji	Status
1	Kasus Uji <i>Login</i>	Valid
2	Kasus Uji <i>Tambah User</i>	Valid
3	Kasus Uji <i>Hapus User</i>	Valid
7	Kasus Uji <i>Import Data latih</i>	Valid
8	Kasus Uji <i>Tampil Vonis</i>	Valid
9	Kasus Uji <i>Logout</i>	Valid

5.2 Pengujian Akurasi

Pengujian akurasi dilakukan dengan cara membandingkan hasil dari pakar dengan hasil *output* dari program. Data training dimasukkan terlebih dahulu untuk membentuk *Rule*. Lalu kemudian sejumlah data uji dimasukkan dalam sistem lalu diproses dengan metode *Iterative Dichotomiser Tree*. Kemudian data uji tersebut dicocokkan dengan data aktual dari pakar.

5.2.1 Hasil Uji Coba

Pada pengujian akurasi ini dilakukan untuk membandingkan akurasi dari *tree* dan aturan yang terbentuk. Pengujian ini dilakukan secara berulang-ulang untuk mengetahui tingkat akurasi dan aturan yang terbentuk pada variasi data latih. Pengujian dilakukan dengan 25 data uji untuk setiap kali pengujian, yang berbeda hanya jumlah data latih dan variasi data latih yaitu 25 dan 40 dan masing-masing dilakukan variasi sebanyak 4 kali, hal ini dilakukan untuk mempermudah pengambilan kesimpulan pada proses analisa. Hasil dari uji coba tersebut ditunjukkan pada tabel 5.2.

Tabel 5.2 Hasil Pengujian Akurasi

Uji Coba	Aturan Yang Terbentuk	Akurasi	Aturan Yang Terbentuk	Akurasi
	25 Data Latih		40 Data Latih	
Uji coba 1	10	92	14	92
Uji coba 2	11	92	14	88
Uji coba 3	8	76	17	92
Uji coba 4	10	68	16	92

Berikut contoh hasil *Rule* yang didapatkan dari 25 data latih pada uji coba ke 3 :

1. Jika Terdakwa masih dibawah umur='Ya' Maka Hukumannya adalah Ringan
2. Jika Terdakwa masih dibawah umur='Tidak' dan Terdakwa berusia lanjut dan sering sakit-sakitan='Tidak' dan Terdakwa berbelit-belit dan mempersulit jalannya persidangan='Ya' Maka Hukumannya adalah Berat
3. Jika Terdakwa masih dibawah umur='Tidak' dan Terdakwa berusia lanjut dan sering sakit-sakitan='Tidak' dan Terdakwa berbelit-belit dan mempersulit jalannya persidangan='Tidak' dan Dalam persidangan terdakwa berkata jujur dan memperlancar proses persidangan='Tidak' Maka Hukumannya adalah Ringan
4. Jika Terdakwa masih dibawah umur='Tidak' dan Terdakwa berusia lanjut dan sering sakit-sakitan='Tidak' dan Terdakwa berbelit-belit dan mempersulit jalannya persidangan='Tidak' dan Dalam persidangan

terdakwa berkata jujur dan memperlancar proses persidangan='ya' dan Terdakwa belum pernah dihukum sebelumnya='Ya' Maka Hukumannya adalah Sedang

5. Jika Terdakwa masih dibawah umur='Tidak' dan Terdakwa berusia lanjut dan sering sakit-sakitan='Tidak' dan Terdakwa berbelit-belit dan mempersulit jalannya persidangan='Tidak' dan Dalam persidangan terdakwa berkata jujur dan memperlancar proses persidangan='ya' dan Terdakwa belum pernah dihukum sebelumnya='Tidak' dan Terdakwa sedang mengandung kurang lebih 3 bulan='Ya' Maka Hukumannya adalah Sedang

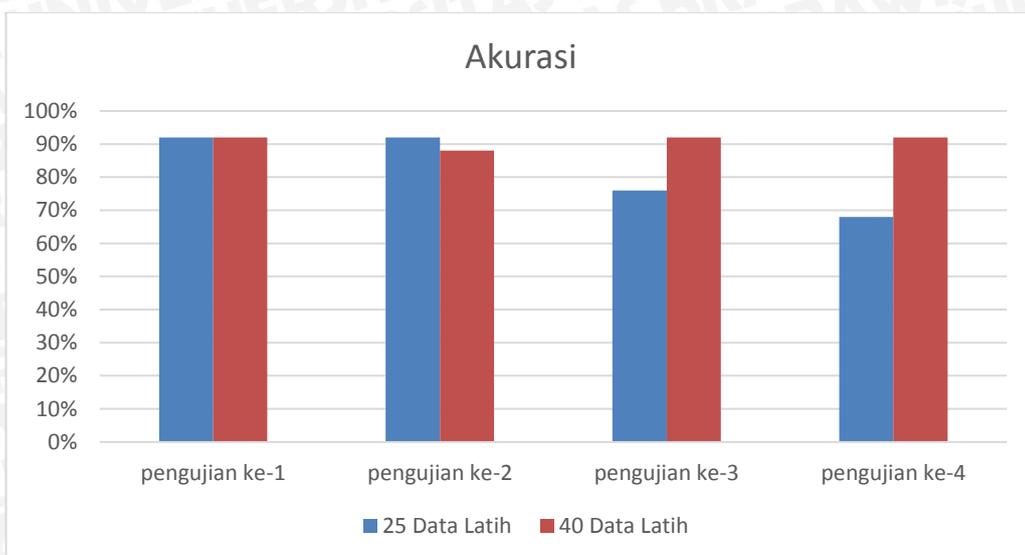
6. Jika Terdakwa masih dibawah umur='Tidak' dan Terdakwa berusia lanjut dan sering sakit-sakitan='Tidak' dan Terdakwa berbelit-belit dan mempersulit jalannya persidangan='Tidak' dan Dalam persidangan terdakwa berkata jujur dan memperlancar proses persidangan='ya' dan Terdakwa belum pernah dihukum sebelumnya='Tidak' dan Terdakwa sedang mengandung kurang lebih 3 bulan='Tidak' Maka Hukumannya adalah Berat

7. Jika Terdakwa masih dibawah umur='Tidak' dan Terdakwa berusia lanjut dan sering sakit-sakitan='ya' dan Dalam persidangan terdakwa berkata jujur dan memperlancar proses persidangan='Ya' Maka Hukumannya adalah Ringan

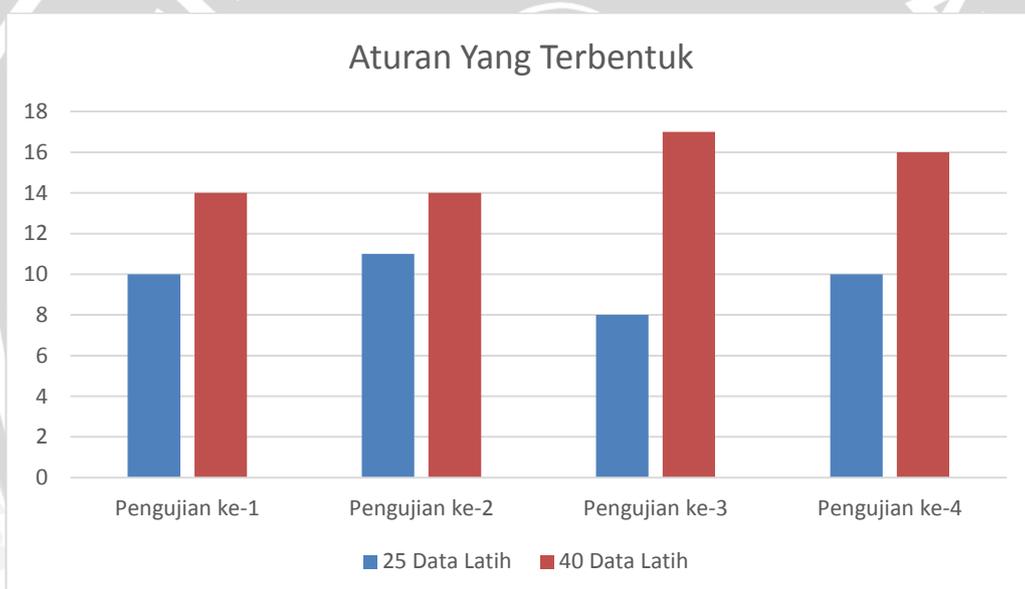
8. Jika Terdakwa masih dibawah umur='Tidak' dan Terdakwa berusia lanjut dan sering sakit-sakitan='ya' dan Dalam persidangan terdakwa berkata jujur dan memperlancar proses persidangan='Tidak' Maka Hukumannya adalah Sedang

Pada hasil pengujian 1 terbentuk aturan pada 25 data latih sebanyak 10 aturan dan 14 aturan untuk 40 data latih. Pada pengujian ke-2 diperoleh 11 aturan untuk 25 data latih dan 14 untuk 40 data latih. Pada pengujian 3 didapat 8 aturan untuk 25 data latih dan 17 aturan untuk 40 data latih. Sedangkan pada pengujian ke-4 didapatkan sebanyak 10 aturan untuk data 25 data latih dan 16 aturan untuk 40 data latih. Perbedaan jumlah aturan yang terbentuk dikarenakan variasi data yang berbeda-beda sehingga hasil perhitungan *Information Gain* berbeda dan *tree* yang terbentuk berbeda pula. Semakin banyak data latih akan membentuk *tree* yang semakin semakin besar dan juga aturan yang semakin banyak.

Akurasi yang didapat dari hasil uji coba menunjukkan variasi data pada pengujian ke-1 mendapatkan hasil terbaik dengan tingkat akurasi sebesar 92% untuk 25 data latih dan 92% untuk 40 data latih. Berdasarkan hasil pengujian ini dan pengamatan pada variasi data latih yang digunakan bisa diambil kesimpulan kenapa pada pengujian ke-1 bisa menghasilkan akurasi yang terbaik, hal ini dikarenakan variasi data pada pengujian ke-1 data latih lebih mencirikan *classnya* serta tingkat kemiripan variasi data latih dengan data ujinya lebih besar dibandingkan data latih yang lain. Berikut grafik perbandingan jumlah aturan yang terbentuk dan akurasi dapat dilihat pada gambar 5.1 dan 5.2.



Gambar 5.1 Hasil Akurasi



Gambar 5.2 Aturan Yang Terbentuk

BAB 6 PENUTUP

Bagian ini memuat kesimpulan dan saran terhadap skripsi. Kesimpulan dan saran disajikan secara terpisah, dengan penjelasan sebagai berikut:

6.1 Kesimpulan

Berdasarkan hasil perancangan dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Aplikasi sistem pakar penentuan pasal dan lama hukuman untuk permasalahan tindak pidana penganiayaan dapat diterapkan sesuai perancangan dan dapat membantu *user* menentukan pasal sesuai kasus yang dilanggar dan lama hukuman yang didapatkan.
2. Sistem ini mampu membuat *Rule* berdasarkan pada pengetahuan pakar dengan menggunakan metode *Iterative Dichtomiser Tree*.
3. Berdasarkan hasil pengujian, variasi dan jumlah banyak data sangat berpengaruh terhadap *tree* dan aturan yang terbentuk serta akurasi yang didapat. Jika variasi pada latih tidak berubah maka hasil aturan dan *tree* yang terbentuk pun tidak akan berubah.
4. Jumlah aturan terbanyak yang didapatkan adalah 17 aturan yakni pada pengujian ke-3 dengan jumlah data latih sebanyak 40 data latih, sedangkan untuk yang paling sedikit adalah 8 aturan yang didapatkan pada 25 data latih pada pengujian ke-3. Untuk nilai akurasi tertinggi sebesar 92% terdapat pada beberapa data latih sedangkan akurasi terendah sebesar 68% pada pengujian ke-4 dengan jumlah 25 data latih.

6.2 Saran

Saran untuk pengembangan Sistem Pakar Penentuan Pasal dan Lama Hukuman Untuk Permasalahan Tindak Pidana Penganiayaan antara lain :

1. Dalam pengembangan selanjutnya diharapkan dapat menghasilkan sistem yang lebih kompleks dengan menambah bab yang lain dalam kuhp , misalnya bab pembunuhan, bab pemerkosaan yang sedang marak terjadi akhir-akhir ini.
2. Dalam pengembangan selanjutnya dapat dilakukan penambahan metode lain agar akurasi sistem meningkat.

DAFTAR PUSTAKA

- [HAN-01] Han, Jiawei dan Micheline Kamber. 2001. "Data Mining : Concepts and Technique", Morgan Kaufmann Publisher, San Francisco, USA.
- [KUS-03] Kusumadewi, Sri. 2003. Artificial Intelligence (Teknik dan Aplikasinya).Yogyakarta : Graha Ilmu.2010
- [TAN-06] Tan P. N. 2006. "Introduction to Data Mining", Addison Wesley.
- [WYD-09] Wahyudin. 2009. "Metode Iterative Dichotomizer 3 (ID3) Untuk Penyeleksian Penerimaan Mahasiswa Baru", IC2T, Volume 2 Nomor 2, Bandung.
- [KOH-99] Kohavi, R., Quinlan. 1999. "Decision Tree Discovery", AAAI and The MIT, Pres, 1-16.
- [SEL-15] Isamuddin, Selly Johansyah. 2015. "Klasifikasi Keluarga Sejahtera Menggunakan Metode *Decision Tree* Dengan Algoritma *Iterative Dichotomiser 3* (Id3) Pada Kecamatan Batu", Universitas Brawijaya, Malang.
- [SUK-11] Suknovic, M., et al. 2011, "Reusable Component in *Decision Tree* Induction Algorithms", Springer-Verlag.
- [LIG-05] Liang, G. 2005. "A Comparative Study of Three *Decision Tree* Algorithms: ID3, Fuzzy ID3 and Probabilistic Fuzzy ID3", Informatics & Economics Erasmus University Rotterdam, The Netherlands.
- [MIT-97] Mitchell, Tom. 1997. "Machine Learning", Singapore, McGraw-Hill.
- [ARH-05] Arhami, Muhammad. 2005. "Konsep Dasar Sistem Pakar", Penerbit Andi,Jakarta
- [ONG-04] Ongko, Fendhy . 2004. "PERANCANGAN DAN PEMBUATAN APLIKASI SISTEM PAKAR UNTUK PERMASALAHAN TINDAK PIDANA TERHADAP HARTA KEKAYAAN", Universitas Kristen Petra.
- [ADP-10] Pranajaya, Aditya. 2010. "APLIKASI SISTEM PAKAR UNTUK PERMASALAHAN TINDAK PIDANA YANG MENAKIBATKAN KEHILANGAN NYAWA BERBASIS WEB ",Universitas Komputer Indonesia, Bandung.
- [YSM-11] Youllia Indrawaty N., Subarjah Mulyadi. 2011. "PERANCANGAN DAN REALISASI (KITAB SISTEM PAKAR IUKUM KUIIP UNDANG-UNDANG HUKUM PI NANA) TENTANG PELANGGARAN DI INDONESIA", Institut Teknologi Nasional, Malang.
- [FTM-13] Frizka Tri Murti Suci Megarani. 2013." PEMBUATAN APLIKASI SISTEM PAKAR HUKUM PIDANA DAN PERDATA BERBASIS SISTEM OPERASI ANDROID", Sekolah Tinggi Manajemen Informatika Dan Komputer Amikom, Yogyakarta.

LAMPIRAN

Kasus Hirarki

No	Kasus							Pasal
1	K003=Ya							352A2 atau 351A5
2	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Ya	K007=Ya/K008=Ya/K009=Ya	K011=Ya	355A1 jo 356
3	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Ya	K007=Ya/K008=Ya/K009=Ya	K012=Ya	355A2 jo 356
4	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Ya	K007=Tidak/K008=Tidak/K009=Tidak	K011=Ya	355A1
5	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Ya	K007=Tidak/K008=Tidak/K009=Tidak	K012=Ya	355A2
6	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K010=Ya	353A1 jo 356
7	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K011=Ya	353A2 jo 356
8	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K012=Ya	353A3 jo 356
9	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K010=Ya	353A1
10	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K011=Ya	353A2
11	K001=Ya	K002=Ya	K004=Ya	K005=Ya	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak	K012=Ya	353A3
12	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Ya	K007=Ya/K008=Ya/K009=Ya	K011=Ya	354A1 jo 356
13	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Ya	K007=Ya/K008=Ya/K009=Ya	K012=Ya	354A2 jo 356
14	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Ya	K007=Tidak/K008=Tidak/K009=Tidak	K011=Ya	354A1
15	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Ya	K007=Tidak/K008=Tidak/K009=Tidak	K012=Ya	354A2
16	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K010=Ya	351A1 jo 356
17	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K011=Ya	351A2 jo 356
18	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K012=Ya	351A3 jo 356
19	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Ya/K008=Ya/K009=Ya	K013=Ya	351A4 jo 356

(lanjutan)

No	Kasus							Pasal	
20	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Ya/K008=Ya/K009=Ya		K014=Ya	352A1 jo 356
21	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak		K010=Ya	351A1
22	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak		K011=Ya	351A2
23	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak		K012=Ya	351A3
24	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak		K013=Ya	351A4
25	K001=Ya	K002=Ya	K004=Ya	K005=Tidak	K006=Tidak	K007=Tidak/K008=Tidak/K009=Tidak		K014=Ya	352A1
26	K001=Ya	K002=Ya	K004=Tidak			K007=Ya/K008=Ya/K009=Ya		K011=Ya	358-1 jo 356
27	K001=Ya	K002=Ya	K004=Tidak			K007=Ya/K008=Ya/K009=Ya		K012=Ya	358-2 jo 356
28	K001=Ya	K002=Ya	K004=Tidak			K007=Tidak/K008=Tidak/K009=Tidak		K011=Ya	358-1
29	K001=Ya	K002=Ya	K004=Tidak			K007=Tidak/K008=Tidak/K009=Tidak		K012=Ya	358-2

Keterangan :

No	Kasus	Keterangan
1	K001	Apakah anda melakukan penganiayaan?
2	K002	Apakah anda melakukan penganiayaan di Indonesia?
3	K003	Apakah anda hanya sekedar melakukan percobaan penganiayaan?
4	K004	Apakah penganiayaan tersebut anda lakukan secara individu?
5	K005	Apakah penganiayaan tersebut anda lakukan secara berencana?
6	K006	Apakah ada unsur kesengajaan dalam penganiayaan yang anda lakukan?
7	K007	Apakah anda memiliki hubungan kekeluargaan dengan korban?
8	K008	Apakah korban merupakan anggota pejabat?
9	K009	Apakah anda menggunakan racun dalam penganiayaan tersebut?
10	K010	Apakah penganiayaan tersebut mengakibatkan luka ringan?

(lanjutan)

No	Kasus	Keterangan
11	K011	Apakah penganiayaan tersebut mengakibatkan luka berat?
12	K012	Apakah penganiayaan tersebut mengakibatkan kematian?
13	K013	Apakah penganiayaan tersebut merusak kesehatan?
14	K014	Apakah penganiayaan tersebut mengakibatkan penyakit?

Dataset

NO	Nama	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	Hukuman
1	AS	1	1	0	0	0	0	0	0	1	0	0	Ringan
2	DR	0	1	1	0	0	0	0	0	0	0	0	Ringan
3	FG	1	0	1	1	1	0	0	0	0	0	1	Ringan
4	MLB	0	1	0	0	0	1	0	1	1	0	0	Ringan
5	KDH	1	1	1	0	0	1	0	0	0	1	0	Ringan
6	ARN	0	1	1	0	0	0	0	0	0	0	0	Ringan
7	POL	1	0	1	0	1	1	0	0	0	0	0	Ringan
8	LEN	0	0	1	1	1	1	0	0	0	0	0	Ringan
9	NM	0	0	1	0	1	0	0	0	0	0	0	Ringan
10	RH	1	0	1	1	0	0	0	0	0	0	0	Ringan
11	LM	1	0	1	0	0	0	1	0	0	0	0	Ringan
12	KH	1	0	1	1	1	1	0	0	0	0	1	Ringan
13	TY	1	0	0	0	0	0	1	0	1	0	0	Ringan
14	UD	0	0	1	1	1	1	0	1	0	0	0	Ringan
15	FA	0	0	1	1	1	1	0	1	0	0	1	Ringan
16	YRP	0	0	0	0	0	0	0	0	0	0	0	Ringan

(lanjutan)

NO	Nama	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	Hukuman
17	AWP	1	1	1	0	0	1	0	0	0	0	1	Ringan
18	IDP	1	0	0	1	0	1	0	0	1	0	0	Ringan
19	MI	1	0	0	1	0	1	0	0	0	0	1	Ringan
20	DC	1	1	1	0	0	0	0	0	0	0	1	Ringan
21	NB	0	0	0	0	0	1	1	1	0	0	1	Sedang
22	MAL	0	0	0	1	1	0	0	0	1	0	1	Sedang
23	LHM	0	0	1	0	0	1	1	0	0	1	1	Sedang
24	DNS	1	0	0	1	0	0	0	0	1	1	0	Sedang
25	BN	0	0	0	0	1	0	0	0	1	1	0	Sedang
26	MW	1	0	1	1	1	0	0	0	0	1	1	Sedang
27	SF	1	0	0	1	1	0	0	0	1	1	0	Sedang
28	BW	0	0	0	0	1	1	0	1	1	0	0	Sedang
29	WU	0	0	0	1	1	0	0	1	0	1	0	Sedang
30	GR	1	0	1	1	0	0	0	0	0	1	1	Sedang
31	HD	0	0	1	0	0	1	1	1	0	0	1	Sedang
32	JK	1	0	0	1	1	0	0	0	1	1	0	Sedang
33	SB	1	0	0	0	1	0	0	0	1	0	1	Sedang
34	KR	1	0	0	1	0	1	0	0	1	1	0	Sedang
35	PS	0	0	1	0	0	0	1	1	0	1	0	Sedang
36	KP	1	0	1	1	0	1	0	0	0	1	1	Sedang
37	LO	0	0	1	1	1	0	0	1	0	0	1	Sedang
38	KF	0	0	0	1	0	1	0	1	1	0	0	Sedang
39	HOU	0	0	0	1	0	0	0	1	1	0	1	Berat
40	KHS	0	0	0	0	1	0	0	0	1	1	1	Berat

(lanjutan)

NO	Nama	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	Hukuman
41	MBS	0	0	0	1	0	0	0	1	1	0	1	Berat
42	NH	0	0	1	0	0	0	0	1	0	1	1	Berat
43	AB	0	0	1	1	0	0	0	1	0	1	1	Berat
44	VN	0	0	0	0	0	0	0	1	1	1	0	Berat
45	LP	0	0	0	1	0	0	0	0	1	1	1	Berat
46	DG	1	0	0	1	0	0	0	0	1	1	1	Berat
47	SDR	0	0	0	1	0	0	0	1	1	1	0	Berat
48	EG	0	0	0	1	0	0	0	1	1	0	1	Berat
49	RR	0	0	0	1	0	0	0	1	1	0	1	Berat
50	MS	0	0	1	0	0	0	0	1	0	1	1	Berat
51	MH	1	0	0	1	0	0	0	0	1	1	1	Berat
52	DY	0	0	0	1	0	0	0	1	1	1	0	Berat

Keterangan :

Kode	Deskripsi Parameter
P1	Terdakwa belum pernah dihukum sebelumnya
P2	Terdakwa masih dibawah umur
P3	Dalam persidangan terdakwa berkata jujur dan memperlancar proses persidangan
P4	Terdakwa mempunyai istri dan beberapa anak kecil
P5	Terdakwa berusia lanjut dan sering sakit-sakitan
P6	Terdakwa menyerahkan diri kepada yang berwajib dan mengakui kesalahannya
P7	Terdakwa sedang mengandung kurang lebih 3 bulan
P8	Pada waktu melakukan kejahatan terdakwa sedang dalam masa percobaan
P9	Terdakwa berbelit-belit dan mempersulit jalannya persidangan

(lanjutan)

Kode	Deskripsi Parameter
P10	Terdakwa mengancam para saksi
P11	Korban berjumlah lebih dari 1 orang

Kategori Hukuman	Lama Hukuman
Ringan	0-30% dari hukuman maksimal
Sedang	31%-65% dari hukuman maksimal
Berat	66%-100% dari hukuman maksimal

Kediri, Mei 2016
Pakar / Hakim

Bondan Supodo, S.H