

BAB II LANDASAN TEORI

2.1 Artificial Intelligence

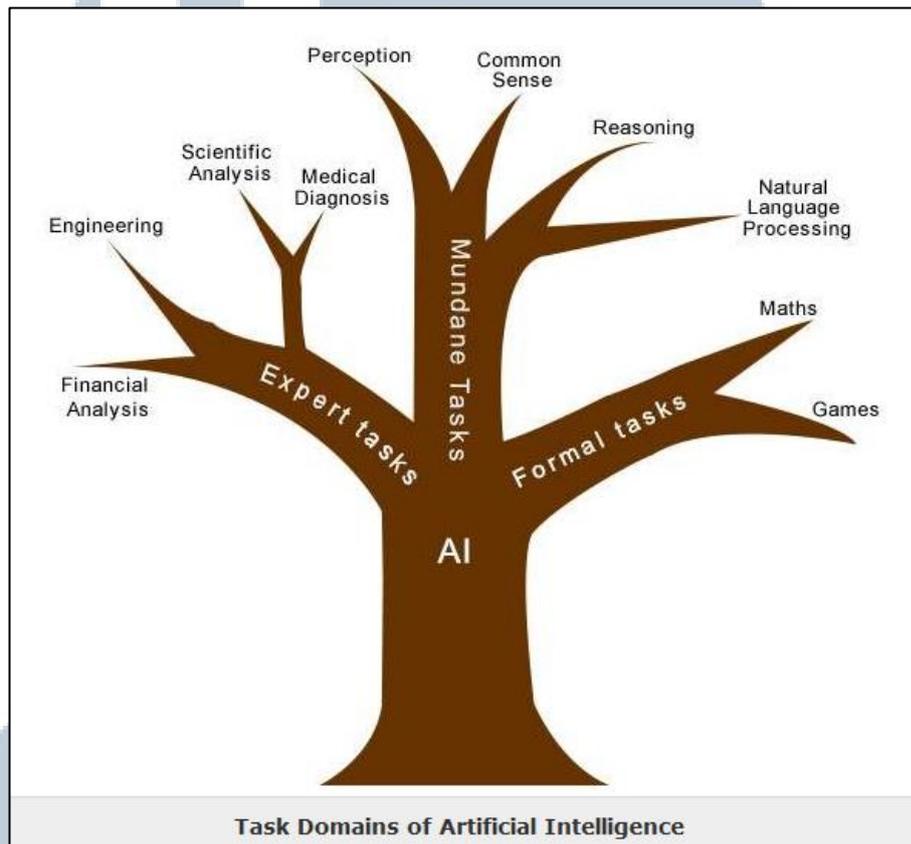
Artificial intelligence (AI) menurut John McCarthy merupakan suatu ilmu dan teknik dalam menciptakan mesin yang bersifat cerdas, terutama dalam menciptakan program atau aplikasi komputer cerdas. AI adalah suatu langkah untuk menciptakan komputer, robot, atau aplikasi atau program yang bekerja secara cerdas, layaknya seperti manusia (McCarthy, 2007).

Tujuan diciptakannya AI itu sendiri untuk:

- a. Menciptakan suatu sistem pakar, yakni suatu sistem yang dapat melakukan perilaku cerdas, belajar, mendemonstrasikan, menjelaskan, dan menyarankan *user*.
- b. Untuk mengimplementasikan kecerdasan daripada manusia ke dalam mesin, menciptakan suatu sistem yang dapat mengerti, berpikir, belajar, dan berperilaku seperti manusia.

Hal-hal yang berkontribusi untuk AI itu sendiri diantaranya adalah bidang Ilmu Komputer, Biologi, Psikologi, Bahasa, Matematika, dan Teknik. Salah satu langkah besar dalam menciptakan komputer yang berhubungan dengan kecerdasan buatan adalah, berpikir dengan logika, belajar, dan menyelesaikan permasalahan. Teknik yang digunakan oleh AI dalam menyelesaikan permasalahan dengan merapikan suatu informasi dan pengetahuan sehingga dapat diakses dan dipahami dengan mudah oleh *user*, dapat dengan mudah dimodifikasi untuk memperbaiki *error*, dan dapat berguna di berbagai situasi walaupun masih belum sempurna atau akurat.

Klasifikasi *domain* dari AI tergolong menjadi *Formal tasks*, *Mundane tasks*, dan *Expert tasks*, digambarkan pada Gambar 2.1 (http://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_research_areas.htm, 2016)



Gambar 2.1 *Task Domains of Artificial Intelligence*

(Sumber: http://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_research_areas.htm)

Sistem yang dibangun pada skripsi ini digolongkan dalam domain *expert tasks*, dengan cabang *engineering*. Untuk menciptakan sistem tersebut digunakan teknik *learning*, suatu teknik yang secara otomatis menemukan *rules* yang dapat berlaku umum untuk data-data yang banyak dan belum pernah diketahui sebelumnya, serta memiliki kelebihan dimana sistem dapat mempelajari data baru yang ada, dimana

seiring semakin banyak data yang dipelajari maka semakin meningkat pula *performance measure*-nya.

2.2 Expert System

Expert system atau sistem pakar merupakan suatu program atau aplikasi dalam komputer yang merepresentasikan pengetahuan dari beberapa ahli atau pakar tentang suatu bidang, dan menghasilkan suatu pandangan untuk penyelesaian permasalahan ataupun saran dan solusi (Jackson, 1999).

Sistem pakar memerlukan akses yang efisien kedalam domain *knowledge base* dan *reasoning mechanism* untuk menerapkan *knowledge* atau informasi ke dalam permasalahan yang diberikan, dan biasanya sistem pakar juga dapat menjelaskan kepada *user* yang menggunakan sistem bagaimana suatu hasil keputusan dapat diperoleh oleh sistem pakar (Bullinaria, 2005).

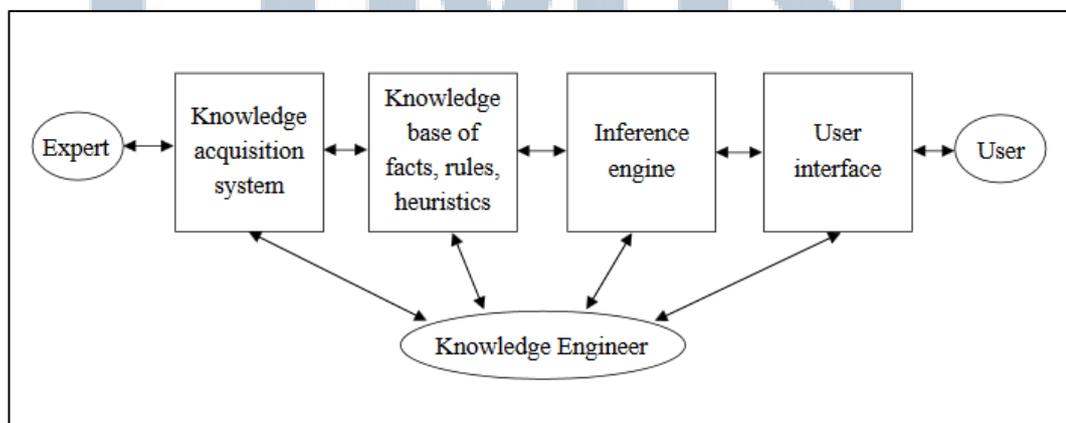
Sistem pakar sendiri umumnya digunakan untuk:

- a. Menginterpretasikan *data*, seperti pengukuran data sonar dan geofisikal,
- b. Diagnosa suatu malfungsi, seperti kerusakan pada suatu alat atau barang atau penyakit pada manusia,
- c. Analisa struktural atau konfigurasi dari objek yang bersifat kompleks, seperti senyawa kimia atau sistem komputer,
- d. Perencanaan urutan suatu tindakan, seperti sesuatu yang bisa dilakukan oleh robot,
- e. Melakukan prediksi akan sesuatu yang akan terjadi, seperti prediksi cuaca, saham.

Sistem pakar dapat dibedakan dengan sistem yang bersifat konvensional dengan melihat beberapa karakteristik daripada sistem pakar itu sendiri (Bullinaria, 2005), diantaranya:

- a. Sistem pakar mensimulasikan penalaran manusia tentang domain suatu masalah, daripada mensimulasikan domain masalah itu sendiri,
- b. Sistem pakar melakukan penalaran sebagai representasi dari pengetahuan manusia, sebagai tambahan melakukan perolehan data dan perhitungan,
- c. Permasalahan cenderung diselesaikan dengan menggunakan *heuristic* (aturan praktis) atau metode perkiraan atau metode probabilitas, dimana solusi yang dihasilkan tidak dijamin tepat atau optimal,
- d. Sistem pakar biasanya menyediakan penjelasan dan pembenaran dari solusi yang dihasilkan/rekomendasi untuk meyakinkan user bahwa penalaran yang dihasilkan oleh sistem benar adanya.

Proses dalam pembuatan sistem pakar sering disebut dengan *knowledge engineering*, dimana berkaitan dengan seluruh komponen dalam sistem pakar, hal ini digambarkan pada Gambar 2.2 dibawah (Bullinaria, 2005).

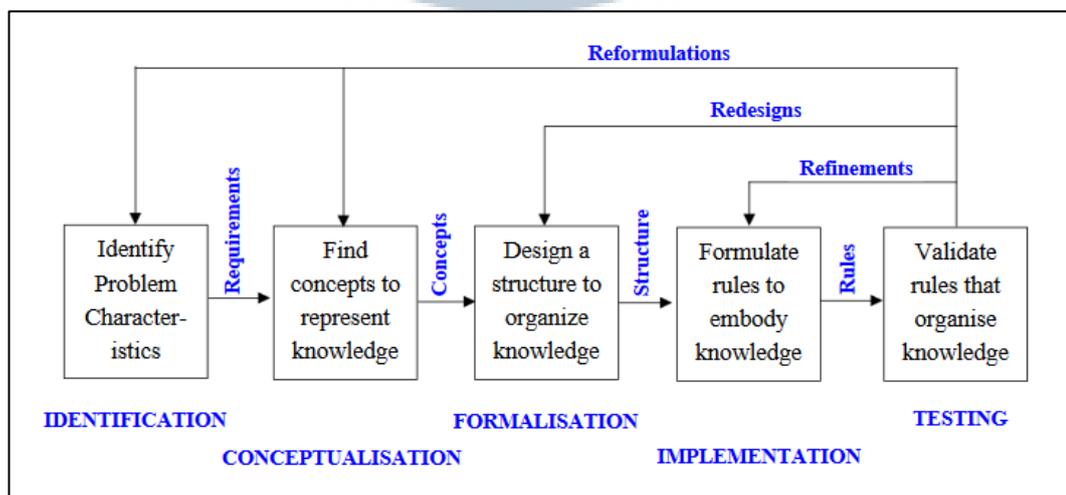


Gambar 2.2 The Architecture of Expert Systems
(Sumber: Bullinaria, 2005)

Proses perolehan informasi dari pakar ke dalam sistem pakar disebut dengan komponen *knowledge acquisition*. *Knowledge acquisition* sendiri biasanya terbagi menjadi tiga tahapan:

- a. *Knowledge elicitation* merupakan interaksi antara pakar dengan pembuat sistem terjadi untuk memperoleh pengetahuan pakar dalam beberapa cara yang sistematis,
- b. Kemudian informasi biasanya disimpan dalam *intermediate representation* yang sifatnya *human friendly*,
- c. Lalu *intermediate representation* daripada informasi tersebut di-*compile* ke dalam *executable form* dimana *inference engine* dapat melakukan proses.

Tahapan-tahapan dalam *knowledge acquisition* dijelaskan lebih rinci dalam Gambar 2.3 (Bullinaria, 2005).



Gambar 2.3 *Stages of Knowledge Acquisition*
(Sumber: Bullinaria, 2005)

Dari gambar 2.3 dapat dijelaskan pada setiap tahapannya sebagai berikut:

- a. *Knowledge identification: interview* yang mendalam dimana informasi dari pakar digali secara mendalam oleh pembuat sistem. Pembuat sistem harus

mengerti *domain* secara baik untuk mengetahui apa saja yang perlu dibicarakan dengan pakar yang bersangkutan,

- b. *Knowledge conceptualization*: mencari konsep yang sifatnya *primitive* dan relasi antara konsep-konsep daripada domain permasalahan,
- c. *Epistemological analysis*: mengungkap sifat struktural pengetahuan yang konseptual, seperti hubungan taksonomi (klasifikasi),
- d. *Logical analysis*: memutuskan bagaimana melakukan penalaran dalam masalah *domain*, dimana informasi seperti ini cenderung sulit didapatkan,
- e. *Implementation analysis*: melakukan prosedur yang sistematis untuk implementasi dan *testing* daripada sistem.

Tahapan selanjutnya setelah *knowledge acquisition* adalah merepresentasikan *knowledge* atau informasi tersebut, dimana secara umum informasi yang diperoleh dari pakar akan dirumuskan menjadi dua cara (Bullinaria, 2005), yaitu:

- a. *Intermediate representation*: informasi terstruktur dimana pembuat sistem dan pakar dapat bekerja secara efisien,
- b. *Production system*: rumusan dimana *inference engine* dari sistem pakar dapat melakukan proses secara efisien

Knowledge/informasi sendiri dibedakan menjadi 2 macam yaitu:

- a. *Domain knowledge*: merupakan pengetahuan daripada pakar yang dapat berupa rules, general/default value, dan sebagainya,
- b. *Case knowledge*: pengetahuan/fakta spesifik tentang suatu kasus, termasuk pengetahuan tentang kasus-kasus tertentu.

Sistem pakar ini akan memiliki *domain knowledge* secara *built-in*, dan harus terintegrasi dengan *case knowledge* yang berbeda dimana akan tersedia setiap kali sistem digunakan. Knowledge based dari sistem ini diperoleh dari data tentang evaluasi kasus kerusakan ban dari pakar dan reasoning mechanism yang didasarkan pada buku panduan pemahaman ban yaitu Bridgestone *Tire Advisor Handbook* (PT. Bridgestone Tire Indonesia, 2008) dan Bridgestone *Consumer Selling Kit* (PT. Bridgestone Tire Indonesia, 2010). Proses pemilihan atribut atau variabel dalam proses pembuatan sistem didasari fakta yang didapat dari buku panduan tersebut kemudian didiskusikan dengan pakar sehingga didapatkan sebanyak 12 atribut atau variabel yaitu tekanan angin, sudut keselarasan roda, kondisi suspensi, cara berkendara, kondisi beban, kondisi jalan, terkena benda tajam atau tumpul, cara pemasangan ban, kecepatan berkendara, penggunaan velg, rotasi ban, dan *balancing* ban. Knowledge based serta *reasoning mechanism* dari sistem ini dijelaskan dengan *rule* sebagai berikut:

1. IF tekanan angin = tidak sesuai THEN rusak irregular wear,
2. IF tekanan angin = tidak sesuai & kondisi beban = overloading & kecepatan berkendara = tinggi THEN rusak regular wear,
3. IF tekanan angin = tidak sesuai & kondisi beban = overloading & penggunaan velg = tidak sesuai THEN rusak regular wear,
4. IF tekanan angin = tidak sesuai & terkena benda tajam atau tumpul = pernah THEN rusak regular wear,
5. IF tekanan angin = tidak sesuai & kondisi beban = overloading & cara pemasangan ban = tidak tepat THEN rusak regular wear,

6. IF cara berkendara = kasar & kondisi jalan = buruk & tekanan angin = tidak tepat & kondisi beban = overloading THEN rusak regular wear,
7. IF tekanan angin = tidak sesuai & kondisi beban = overloading & kondisi suspensi = tidak baik & cara berkendara = kasar & penggunaan velg = tidak sesuai THEN rusak regular wear,
8. IF rotasi ban = jarang & tekanan angin = tidak sesuai THEN rusak irregular wear,
9. IF terkena benda tajam atau tumpul = pernah & kondisi jalan = rusak & cara mengemudi = kasar THEN rusak regular wear,
10. IF tekanan angin = tidak sesuai & kondisi beban = overloading & cara pemasangan ban = kasar THEN rusak regular wear,
11. IF tekanan angin = tidak sesuai & kondisi beban = overloading & terkena benda tajam atau tumpul = pernah & kecepatan berkendara = tinggi THEN rusak regular wear,
12. IF terkena benda tajam atau tumpul = pernah & cara berkendara = kasar THEN rusak regular wear,
13. IF kondisi suspensi = tidak baik & kondisi berkendara = kasar THEN rusak irregular wear,
14. IF kondisi jalan = rusak & kondisi beban = overloading THEN rusak regular wear,
15. IF tekanan angin = tidak sesuai & kecepatan berkendara = tinggi & cara berkendara = kasar THEN rusak regular wear,
16. IF cara berkendara = kasar THEN rusak regular wear,

17. IF tekanan angin = tidak sesuai & terkena benda tajam atau tumpul = pernah & kondisi jalan = rusak THEN rusak regular wear,
18. IF terkena benda tajam atau tumpul = pernah & tekanan angin = tidak sesuai & kondisi beban = overloading & cara mengemudi = kasar & kecepatan berkendara = tinggi THEN rusak regular wear,
19. IF tekanan angin = tidak sesuai & kondisi beban = overloading & cara berkendara = kasar THEN rusak regular wear,
20. IF sudut keselarasan roda = tidak sesuai THEN rusak irregular wear,
21. IF cara pemasangan ban = tidak tepat & kondisi beban = overloading & cara berkendara = kasar THEN rusak regular wear,
22. IF tekanan angin = tidak sesuai & sudut keselarasan roda = tidak pas & kondisi suspensi = tidak baik THEN rusak irregular wear.

Kemudian tahapan berikutnya dalam sistem pakar adalah tahapan *inference engine* dimana merupakan proses yang mengandung mekanisme daripada pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu permasalahan yang diterapkan kedalam sistem.

Tahapan yang terakhir dalam suatu sistem pakar adalah *user interface*, dimana biasanya terbagi menjadi dua komponen, *the interviewer component* merupakan komponen berupa dialog dengan *user* agar sistem dapat memperoleh atau membaca *data* yang dimasukkan oleh *user* ke dalam sistem, dan *the explanation component* dimana memberikan penjelasan tentang cara kerja bagaimana suatu solusi yang dihasilkan oleh sistem pakar dapat terjadi (Bullinaria, 2005)

Sistem pakar yang dibangun dalam skripsi ini menarik simpulan berdasarkan *rules*, dalam memperoleh *rules* tersebut didasarkan pada pengumpulan fakta-fakta

dari pakar. Fakta-fakta yang diperoleh kemudian diproses dengan metode *learning*, sehingga dapat menghasilkan suatu solusi.

2.3 Data Mining

Data mining merupakan suatu proses untuk menemukan pola atau pengetahuan yang bermanfaat secara otomatis dari sekumpulan data yang berjumlah banyak. *Data mining* sendiri sering dianggap sebagai *Knowledge Discovery in Database* (KDD), dimana proses untuk mencari pengetahuan yang bermanfaat dari data (Sunjana, 2010), proses daripada KDD sendiri dapat dijelaskan sebagai berikut:

- a. *Data selection*, proses seleksi data dari sekumpulan data operasional yang perlu dilakukan sebelum proses penggalian informasi dalam KDD dimulai. Data hasil seleksi yang akan digunakan disimpan terpisah dari basis data operasional,
- b. *Pre-processing* atau *cleaning*, setelah proses seleksi data, dilakukan proses *cleaning* pada data, yang merupakan focus dari KDD. Proses *cleaning* meliputi membuang duplikasi data, memeriksa konsistensi data, serta memperbaiki kesalahan pada data. Pada proses *cleaning* dilakukan pula proses *enrichment*, proses memperkaya data dengan informasi lain yang relevan dan diperlukan untuk KDD,
- c. *Transformation*, proses transformasi pada data yang telah dipilih sehingga data tersebut sesuai untuk digunakan dalam proses *data mining*, proses ini disebut dengan *coding*. Proses *coding* sendiri merupakan suatu proses yang bersifat kreatif, sangat tergantung dengan jenis atau pola informasi yang akan dicari dalam basis *data*,

- d. *Interpretation* atau *evaluation*, tahap pemeriksaan apakah suatu pola atau informasi yang ditemukan berbeda dengan fakta atau hipotesa yang ada sebelumnya. Pola informasi yang dihasilkan daripada proses *data mining* perlu ditampilkan dalam bentuk yang mudah dipahami (Sunaja, 2010).

2.4 Klasifikasi

(Han, Kamber, dan Pei, 2012) dalam bukunya yang berjudul “*Data Mining: Concepts and Techniques*” menjelaskan ada beberapa teknik atau pola yang digunakan untuk *data mining* yaitu:

- a. *Class* atau *concept description (characterization dan discrimination)*,
- b. *Frequent patterns, associations, correlation*,
- c. *Classification dan Regression*
- d. *Cluster Analysis*
- e. *Outliner Analysis*

Dari lima teknik di atas, klasifikasi merupakan teknik atau pola yang digunakan dalam penelitian ini. Klasifikasi sendiri merupakan proses untuk menemukan suatu model atau fungsi yang mendeskripsikan dan membedakan suatu kelas atau konsep data. Model itu sendiri dibentuk dari analisis beberapa *set data training*. Kemudian model ini digunakan untuk memprediksikan label daripada suatu objek yang belum diketahui labelnya.

Model tersebut direpresentasikan dalam berbagai bentuk diantaranya adalah *classification rules, decision trees, mathematical formula*, atau *neural network*. Dalam penelitian ini model yang digunakan untuk klasifikasi adalah *decision tree*.

2.5 Decision Tree

Decision Tree merupakan suatu metode klasifikasi dan prediksi yang sangat terkenal dalam performa kinerjanya serta tingkat keakuratannya. Metode pohon keputusan dapat mengubah fakta dalam jumlah yang sangat besar menjadi suatu pohon keputusan yang merepresentasikan aturan, dimana aturan dapat dengan mudah dipahami dengan bahasa alami, serta juga dapat diekspresikan dalam bentuk bahasa basis data seperti *Structured Query Language* (SQL) untuk mencari *record* pada kategori tertentu.

Untuk membuat suatu pohon keputusan tersebut diperlukan suatu algoritma yang dapat mengklasifikasikan suatu *dataset* menjadi suatu pohon keputusan berdasarkan kelas-kelasnya (Hastuti, 2012). Dalam penelitian ini algoritma yang digunakan untuk membuat pohon keputusan tersebut adalah algoritma C 4.5.

2.6 Algoritma C 4.5

Algoritma C 4.5 sendiri merupakan algoritma yang bersifat *machine learning*, dimana sistem akan diberikan suatu kelompok data untuk dipelajari yang disebut *learning dataset*, yang kemudian hasil daripada pembelajaran tersebut digunakan untuk mengolah *data* yang baru disebut sebagai *test dataset* (Hastuti, 2012).

Secara umum (Irma, 2013) algoritma C 4.5 terdiri dari dua tahap utama, dimana yang pertama adalah pemilihan atribut dengan nilai *gain* tertinggi daripada atribut-atribut yang ada, dirumuskan sebagai berikut:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} \times Entropy(S_i) \quad \dots (2.1)$$

Keterangan:

S : himpunan kasus

A : atribut

n : jumlah partisi atribut A

|S_i| : jumlah kasus pada partisi ke-i

|S| : jumlah kasus dalam S

Perhitungan dari Rumus 2.1 di atas akan menghasilkan nilai *gain* dari atribut yang paling tinggi, *gain* sendiri merupakan salah satu *attribute selection measure* yang digunakan untuk memilih *test* atribut tiap *node* pada pohon. Atribut dengan *information gain* tertinggi akan dipilih sebagai *test* atribut dari suatu *node*. Perhitungan daripada nilai *entropy* sendiri dapat dirumuskan sebagai berikut:

$$Entropy(S) = \sum_{i=1}^n -p_i \times \log_2 p_i \quad \dots (2.2)$$

Keterangan:

S : himpunan kasus

A : atribut

n : jumlah partisi S

p_i : proporsi dari S_i terhadap S

Dari dua tahap perhitungan di atas kemudian dilanjutkan dengan menghitung nilai dari *split info*, dirumuskan sebagai berikut:

$$SplitInfo(S, A) = - \sum_{i=1}^n \frac{S_i}{S} \log_2 \frac{S_i}{S} \quad \dots (2.3)$$

Keterangan:

S : ruang (data) *sample* yang digunakan untuk *training*

A : atribut

S_i : jumlah *sample* untuk atribut i

Kemudian setelah menghitung *split info* maka dapat diperoleh nilai *gain ratio*, dengan menggunakan rumus berikut:

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInfo(S, A)} \quad \dots (2.4)$$

Setelah melakukan perhitungan sampai selesai, kemudian dilakukan pembuatan *decision tree* yang nantinya akan dibentuk menjadi *rules* dengan menggunakan kaidah IF-THEN-ELSE.

2.7 Precision, Recall, and Accuracy

Precision merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan hasil jawaban yang diberikan oleh sistem, sedangkan *recall* merupakan tingkat keberhasilan daripada sistem dalam menemukan kembali sebuah informasi. *Accuracy* adalah tingkat kedekatan antara nilai prediksi dengan nilai *actual* (sebenarnya) (Raharjo, 2011). Ketiga metode ini digunakan untuk menghitung kinerja daripada sistem, dirumuskan dengan:

$$Precision = \frac{TP}{TP + FP} \quad \dots (2.5)$$

$$Recall = \frac{TP}{TP + FN} \quad \dots (2.6)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \dots (2.7)$$

Keterangan:

TP : *True Positive (Correct Result)*

FP : *False Positive (Unexpected Result)*

FN : *False Negative (Missing Result)*

TN : *True Negative (Correct absence of result)*

2.8 Pengetahuan Ban

Ban pada dasarnya diklasifikasikan menjadi dua struktur yaitu:

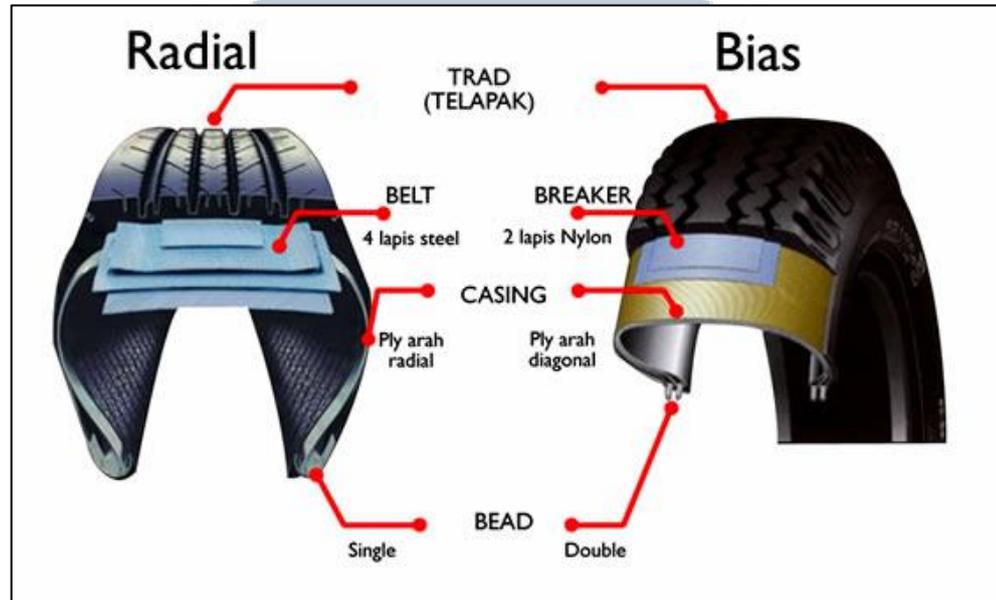
1. Struktur bias

Ban yang dibuat dari banyak lembar *cord* yang digunakan sebagai rangka daripada ban. *Cord* ditenun dengan cara zig-zag membentuk sudut 40 sampai 65 derajat terhadap keliling lingkaran ban.

2. Struktur Radial

Dalam ban radial konstruksi *carcass cord* membentuk 90 derajat sudut terhadap keliling lingkaran ban. Jika dilihat dari samping konstruksi *cord* adalah dalam arah radial terhadap pusat atau *crown* daripada ban. Bagian dari ban berhubungan langsung dengan permukaan jalan diperkuat oleh semacam sabuk pengikat yang dinamakan *breaker* atau *belt*. Ban ini hanya menderita sedikit *deformasi* dalam bentuknya dari gaya sentrifugal bahkan dalam kecepatan tinggi sekalipun. Ban radial juga memiliki *rolling resistance* yang relatif kecil.

Hal tentang struktur daripada ban ini dijelaskan lebih lanjut pada Gambar 2.4 di bawah ini (PT. Bridgestone Tire Indonesia, 2013)



Gambar 2.4 Perbedaan ban radial dan bias
(Sumber:http://www.bridgestone.co.id/tire-information_pengetahuan-ban.html)

Jenis-jenis kerusakan ban secara umum dikategorikan menjadi dua, yaitu:

1. Irregular Wear

Kerusakan yang menyebabkan permukaan ban aus secara tidak merata, sehingga usia pakai daripada ban berkurang secara drastis. Beberapa hal yang menyebabkan kerusakan ini adalah sudut keselarasan roda yang tidak sesuai, tekanan ban yang tidak sesuai, sistem pengereman yang rusak, suspensi yang tidak baik, dan cara berkendara yang kasar. Solusi atas kerusakan jenis ini adalah perlunya untuk menjaga tekanan angin agar tetap selalu stabil, melakukan *sporing* dan *balancing*, melakukan rotasi ban secara berkala, menjaga kondisi pengereman dan suspensi selalu pada kondisi baik.

2. Regular wear

Kerusakan yang diakibatkan oleh masa pakai dari ban tersebut sudah habis. Adapun beberapa faktor yang menyebabkan kerusakan jenis ini lebih cepat terjadi walaupun usia pemakaian daripada ban tergolong pendek diantaranya adalah beban kendaraan yang berlebih, kondisi jalan yang dilalui rusak, pernah terkena benda tajam atau tumpul, serta sering berkendara dengan kecepatan tinggi disertai dengan cara mengemudi yang kasar. Solusi daripada kerusakan jenis ini perlunya mengubah cara berkendara, tidak membawa beban berlebih, melakukan penambalan, dan kemungkinan terburuk adalah perlunya untuk mengganti ban tersebut. (PT. Bridgestone Tire Indonesia, 2010).

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA