BAB II
TINJAUAN PUSTAKA

II.1 Tinjauan Pustaka

Algoritma ID3(Induction of Decision Tree) ditemukan oleh Quinlan (1979), sejak akhir dekade 70-an. Pengembangan ini merupakan cikal bakal algoritma C4.5 pada tahun 1993. Keunggulan dari algoritma ID3 adalah tahap belajar yang cepat, *time complexity* yang rendah, ketelitian klasifikasi yang tinggi. Tujuan algoritma ID3 adalah mendapatkan *decision tree* yang merupakan salah satu bentuk *"Classification Models"* yang terbaik. Salah satu masalah yang dapat dipecahkan menggunakan algoritma ini yakni upaya untuk mendapatkan *decision tree* terbaik yakni pendekatan seminimal mungkin yang konsisten dari sekumpulan data, termasuk dalam kategori algoritma NP-Hard/Completeness. Mekanisme Pembangunan algoritma ini yaitu dilakukan secara *top-down* kemudian diawali dengan pertanyaan *"Attribute* mana yang harus diperiksa pada *root* dari *decision tree"* kemudian pembentukan dilakukan dengan mempartisi *training examples*. Kekuatan utama pada algoritma adalah fungsi *heuristic information gain* untuk memilih atribut terbaik, mewujudkan *Greedy Heuristic Search* antara lain *Hill-Climbing* tanpa *Backtracking*.

Masalah pembelajaran dapat didistribusikan dan diringkas sebagai berikut: data yang didistribusikan antara berbagai situs dan tugas pembelajaran adalah menemukan beberapa pengetahuan yang bermanfaat (misalnya, hipotesis dalam bentuk *decision tree* yang mengklasifikasikan contoh-contoh). Ini dapat dicapai dengan agen pembelajaran yang mengunjungi situs yang

berbeda untuk mengumpulkan informasi yang dibutuhkan untuk menghasilkan hipotesis yang cocok dari data. Serial yang didistribusikan sesuai dengan skenario pembelajaran yang ditunjukkan pada Gambar.

Clementine menawarkan berbagai cara untuk menilai dan meningkatkan kinerja *decision tree*, yaitu seberapa baik pohon mengklasifikasikan contoh-contoh baru. *Cross-validation* yang tersedia dalam C5 memberikan perkiraan berdasarkan klasifikasi *tree* kembali menggunakan satu *sampel*. Algoritma menggunakan validasi silang untuk memperkirakan kurva belajar untuk C5, yaitu bagaimana meningkatkan keakuratan klasifikasi pelatihan ditetapkan dengan ukuran. Pemangkasan berbagai pilihan yang tersedia di C5 untuk menghapus cabang-cabang yang lebih rendah yang mengandung frekuensi kecil. Pilihan *Pruning severity* menentukan tingkat *post-pruning tree* yang dibangun. Semakin besar angka ini, semakin besar tingkat pemangkasan. *Run-time* menekan pemangkasan dengan mengurangi catatan Minimum per cabang ke nol. Percobaan dengan pengaturan yang berbeda dari pilihan *Pruning severity* (semakin besar, semakin besar tingkat pemangkasan). dibandingkan hasil dari dua sampel, pelatihan dan pengujian. Pada titik apakah tree dilakukan pemangkasan lebih dari versi *Favour Accuracy* (Sharma dkk, 2000).

Tujuan dalam peningkatan pembelajaran antara lain untuk menemukan kebijakan yang optimal. Kebijakan optimal adalah pemetaan dari *states* untuk tindakan-tindakan yang dapat memaksimalkan jumlah *rewards*. Nilai dari suatu *states* yang didefinisikan sebagai jumlah dari hadiah yang diterima ketika mulai dalam keadaan itu dan mengikuti kebijakan *terminal state*. Nilai fungsi dapat diperkirakan dengan menggunakan

approximator fungsi umum seperti *neural network, look-up table, or decision tree*. Tabel *lookup* yang langsung dengan metode *subdivides* ruang *input* ke *intervals*. Diantara bagian dari ruang *states* memiliki resolusi yang sama. Pendekatan yang lebih baik akan memungkinkan resolusi tinggi hanya apabila diperlukan. Beberapa usaha telah dilakukan dengan menggunakan resolusi variabel menggunakan tabel, dengan sukses terbatas. *Decision tree* memungkinkan ruang untuk dibagi dengan berbagai tingkat resolusi. Pohon dapat digunakan untuk memetakan sebuah *input* vektor ke salah satu simpul daun, yang sesuai dengan daerah dalam ruang *states*. Pembelajaran dapat digunakan untuk mengasosiasikan sebuah nilai pada masing-masing daerah. *G-learning* menggunakan *decision tree* untuk belajar mengompakan representasi dari nilai fungsi pada masalah dengan *input biner*. Pendekatan yang kami lakukan untuk memperluas metode algoritma yang menangani nilai riil. Tujuannya adalah untuk menyediakan konvergensi yang kuat dan skalabilitas yang baik. Decision tree dipelajari bersama dengan kebijakan (Pyeat dan Addele, 1998).

Pentingnya aplikasi dalam logika AI, dan skala aplikasi ini, mewakili sebuah metodologi baru untuk satu logika yang mungkin akan terjadi tanpa penalaran mekanik. Metodologi ini memaksa ahli teori untuk memikirkan masalah pada skala baru dan pada tingkat *detail* baru, pada gilirannya memiliki efek mendalam pada teori-teori yang dihasilkan. Efek dari metodologi ini akan digambarkan dalam bagian di bawah ini, berurusan dengan berbagai topik di AI logis. Tapi masalahnya digambarkan dengan baik oleh penalaran tentang tindakan dan perubahan. Topik ini diselidiki

dalam filsafat literatur. Penalaran tentang perubahan, setidaknya, bagian dari logika tense (*tense logic*), dan konsekuensi tindakan diselidiki dalam literatur tentang "*seeing to it that”* Walaupun hal itu merupakan perkembangan yang menarik dalam filsafat logika, skala pencapaian sangat berbeda dari tradisi penelitian dalam logika AI. Dalam tradisi ini *The formalisms* tidak hanya mendukung formalisasi yang kompleks, realistis masalah perencanaan, tetapi memberikan wawasan penalaran yang baru tentang efek dalam tindakan kausal, *the persistence of states*, dan *the interactions* antara *actions dan continuous physical processes*. Perkembangan seperti ini akan mungkin terjadi tanpa interaksi antara teori-teori logis dan berskala besar, aplikasi praktis dalam perencanaan otomatis(Belnap, 1996).