

MODUL KECERDASAN BUATAN

Prodi Informatika
UNIVERSITAS ALMA ATA



Daftar ISI

Modul Pembelajaran Mata Kuliah Kecerdasan Buatan	5
BAB 1: Pengantar Kecerdasan Buatan dalam Perspektif Islam.....	5
Definisi dan Ruang Lingkup AI.....	5
Sejarah dan Perkembangan AI	6
AI dalam Perspektif Islam	7
Etika AI dalam Islam.....	8
Studi Kasus: Teknologi AI dan Etika Islam.....	8
BAB 2: Representasi Pengetahuan	9
Pendekatan Deklaratif dan Prosedural.....	10
Frame, Semantic Net, dan Production Rules	10
Ontologi dan Logika Deskriptif	11
Representasi Fakta dan Inferensi	12
BAB 3: Probabilitas Bayes.....	13
Teorema Bayes	14
Naïve Bayes Classifier	14
Bayesian Belief Networks (BBN).....	15
Aplikasi dalam Klasifikasi dan Prediksi.....	16
BAB 4: Logika Fuzzy	17
Fuzzy Sets dan Fuzzy Rules	17
Fungsi Keanggotaan.....	18
Inferensi Fuzzy dan Defuzzifikasi	19
Aplikasi dalam Sistem Kontrol dan Pengambilan Keputusan	21
BAB 5: Version Space	22
Definisi Version Space	23
Contoh Representasi:	23
Generalisasi dan Spesialisasi	24
a. Generalisasi.....	24
b. Spesialisasi	24

Tujuannya:	24
Algoritma Candidate Elimination.....	24
Tujuan:.....	24
Langkah-langkah Algoritma:	24
Contoh Ilustrasi:	25
Referensi Terpercaya.....	25
BAB 6: Sistem Pakar	26
Pengertian Sistem Pakar.....	27
Arsitektur Sistem Pakar	27
1. Antarmuka Pengguna (User Interface)	27
2. Basis Pengetahuan (Knowledge Base)	27
3. Mesin Inferensi (Inference Engine)	27
4. Basis Fakta (Fact Base / Working Memory)	27
5. Modul Penjelas (Explanation Facility).....	28
6. Modul Akuisisi Pengetahuan (Knowledge Acquisition).....	28
BAB 7: Neural Network.....	29
Perceptron dan MLP	29
Struktur Perceptron:	30
Kelebihan dan Keterbatasan Perceptron:	30
Multilayer Perceptron (MLP).....	30
Definisi:	30
Struktur MLP:	31
Fungsi Aktivasi Umum dalam MLP:.....	31
Proses Pembelajaran:.....	31
Keunggulan MLP dibanding Perceptron:	31
Backpropagation	32
Contoh Sederhana.....	32
Kelebihan dan Kekurangan Backpropagation.....	32
Referensi.....	33
BAB 8: Algoritma Genetika	33
Materi Algoritma Genetika.....	33
Kromosom dan Populasi Awal	33
Seleksi, Crossover, dan Mutasi	34

Fungsi Fitness.....	35
Aplikasi Algoritma Genetika.....	35
Referensi.....	35
BAB 9: Natural Language Processing.....	36
Tokenisasi, Stemming, dan POS Tagging.....	37
Tokenisasi	37
Stemming.....	37
POS Tagging (Part-of-Speech Tagging).....	37
Parsing dan Semantic Analysis.....	37
Parsing.....	37
Semantic Analysis	37
Word2Vec dan GloVe.....	38
Word2Vec.....	38
GloVe (Global Vectors for Word Representation)	38
Aplikasi NLP	38
1. Chatbot	38
2. Analisis Sentimen.....	39
Referensi.....	39
BAB 10: Computer Vision.....	39
Pengolahan Citra Digital	40
Pengertian.....	40
Ruang Warna	40
Tahapan Pengolahan Citra	40
Segmentasi dan Ekstraksi Fitur	40
Segmentasi Citra.....	40
Ekstraksi Fitur	40
Deteksi Objek (Pengantar YOLO)	41
Deteksi Objek.....	41
YOLO (You Only Look Once) – Pengantar	41
Referensi.....	42
BAB 11: Robotika.....	42
Pengolahan Citra Digital	43
Segmentasi dan Ekstraksi Fitur	43

Deteksi Objek (YOLO - Pengantar)	44
Struktur Robot dan Sensor	44
Navigasi dan SLAM.....	45
Sistem Kontrol dan Keputusan.....	45

Modul Pembelajaran Mata Kuliah Kecerdasan Buatan

Program Studi: Informatika

Modul Pembelajaran Mata Kuliah Kecerdasan Buatan

BAB 1: Pengantar Kecerdasan Buatan dalam Perspektif Islam

Deskripsi: Mengenalkan dasar-dasar AI serta meninjau perkembangannya dari perspektif ilmiah dan nilai-nilai Islam.

Tujuan Pembelajaran: Setelah mempelajari bab ini, mahasiswa mampu:

1. Menjelaskan definisi dan ruang lingkup kecerdasan buatan (AI).
2. Menelusuri sejarah dan perkembangan AI dari masa ke masa.
3. Mengkaji konsep AI dari perspektif Islam berdasarkan Al-Qur'an dan Hadis.
4. Menganalisis isu etika AI dalam sudut pandang nilai-nilai Islam.
5. Menerapkan prinsip-prinsip etika Islam dalam studi kasus penggunaan AI.

Materi:

Definisi dan Ruang Lingkup AI

Kecerdasan Buatan (Artificial Intelligence/AI) adalah cabang ilmu komputer yang berfokus pada pengembangan sistem atau mesin yang mampu melakukan tugas-tugas yang umumnya memerlukan kecerdasan manusia. Tugas-tugas ini meliputi proses berpikir logis, pengambilan keputusan, pemecahan masalah, pembelajaran dari data, serta memahami dan berkomunikasi menggunakan bahasa alami. John McCarthy, salah satu pelopor AI, mendefinisikan AI sebagai “ilmu dan rekayasa dalam membuat mesin cerdas” (McCarthy, 2007). AI bukan hanya meniru perilaku manusia, tetapi juga

menciptakan cara baru dalam menyelesaikan masalah yang kompleks melalui pendekatan komputasional.

Ruang lingkup AI sangat luas dan mencakup berbagai sub-bidang yang terus berkembang. Di antaranya adalah machine learning, deep learning, natural language processing (NLP), computer vision, robotics, dan expert systems. Masing-masing sub-bidang ini memiliki fungsi dan aplikasi tersendiri, seperti pengenalan wajah dalam computer vision, penerjemahan bahasa otomatis dalam NLP, serta diagnosis penyakit dalam sistem pakar medis. Menurut Russell dan Norvig (2021), pendekatan terhadap AI dapat diklasifikasikan menjadi empat kategori: sistem yang berpikir seperti manusia, bertindak seperti manusia, berpikir secara rasional, dan bertindak secara rasional. Pendekatan ini menggambarkan betapa beragamnya metode dalam mengembangkan AI, dari yang berbasis aturan (symbolic AI) hingga pembelajaran dari data (sub-symbolic AI).

Saat ini, AI telah diterapkan di berbagai sektor, termasuk industri, kesehatan, transportasi, keuangan, dan pendidikan. Teknologi seperti asisten virtual, sistem rekomendasi, mobil otonom, dan analisis prediktif hanyalah sebagian kecil dari penerapan AI dalam kehidupan sehari-hari. Dengan kemampuannya untuk belajar dan beradaptasi, AI dipandang sebagai teknologi masa depan yang dapat meningkatkan efisiensi dan inovasi dalam berbagai aspek kehidupan manusia. Namun demikian, tantangan etis dan sosial juga perlu menjadi perhatian dalam pengembangan dan penerapan AI, terutama terkait privasi data, bias algoritma, dan tanggung jawab hukum.

- McCarthy, J. (2007). *What is Artificial Intelligence?*. Stanford University. <http://jmc.stanford.edu/articles.html>
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Rich, E., & Knight, K. (2010). *Artificial Intelligence* (3rd ed.). McGraw-Hill.

Sejarah dan Perkembangan AI

Perkembangan kecerdasan buatan (AI) berawal dari gagasan klasik mengenai mesin berpikir yang dapat meniru kecerdasan manusia. Pada awal abad ke-20, ide tentang mesin cerdas mulai muncul dalam karya-karya fiksi ilmiah dan pemikiran filosofis. Namun, tonggak penting sejarah AI dimulai pada tahun 1950 ketika **Alan Turing** menerbitkan makalah berjudul "*Computing Machinery and Intelligence*" dan mengajukan tes Turing sebagai cara untuk mengukur kecerdasan mesin. Lalu pada tahun 1956, istilah "*Artificial Intelligence*" secara resmi diperkenalkan oleh **John McCarthy** dalam konferensi di Dartmouth College, yang dianggap sebagai kelahiran resmi bidang AI sebagai disiplin ilmu.

Pada dekade-dekade berikutnya, AI mengalami pasang surut. Periode awal 1960-an hingga 1970-an ditandai dengan optimisme tinggi, ditandai dengan pengembangan

program simbolik seperti *General Problem Solver* dan sistem pakar pertama. Namun, keterbatasan dalam komputasi dan kurangnya data menyebabkan kekecewaan yang memunculkan periode yang disebut sebagai *AI Winter* pada akhir 1970-an hingga awal 1990-an. Kemajuan mulai terlihat kembali ketika pendekatan berbasis statistik dan pembelajaran mesin mulai dikembangkan, khususnya dengan ketersediaan data besar dan peningkatan daya komputasi.

Memasuki abad ke-21, perkembangan AI mengalami lonjakan pesat dengan kemunculan **machine learning** dan **deep learning**, yang didorong oleh kekuatan GPU dan big data. Penerapan AI menjadi sangat luas, mulai dari asisten virtual seperti Siri dan Google Assistant, sistem rekomendasi di platform digital, hingga kendaraan otonom. Kini, AI telah menjadi bagian penting dari transformasi digital global dan terus berkembang menuju kecerdasan buatan umum (AGI) yang diharapkan mampu menyaingi atau melampaui kemampuan kognitif manusia di berbagai bidang.

- McCarthy, J. (2007). *What is Artificial Intelligence?* Stanford University. <http://jmc.stanford.edu/articles.html>
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236), 433–460. <https://doi.org/10.1093/mind/LIX.236.433>

AI dalam Perspektif Islam

Dalam perspektif Islam, kecerdasan buatan dipandang sebagai hasil dari kemampuan akal dan ilmu pengetahuan yang dianugerahkan Allah SWT kepada manusia. Al-Qur'an mendorong umat Islam untuk mengeksplorasi dan memanfaatkan ilmu pengetahuan serta teknologi demi kemaslahatan umat manusia, sebagaimana dalam QS. Al-Mujadilah [58]:11 yang menyatakan bahwa Allah akan meninggikan derajat orang-orang yang beriman dan berilmu. AI, sebagai produk teknologi modern, merupakan salah satu bentuk ikhtiar manusia dalam mengelola pengetahuan dan menciptakan solusi yang bermanfaat. Selama pengembangan dan penggunaannya tidak bertentangan dengan prinsip-prinsip syariat, AI dapat menjadi alat untuk memperkuat nilai-nilai kebaikan, seperti meningkatkan pelayanan kesehatan, pendidikan, dan keadilan sosial.

Namun demikian, Islam juga menekankan pentingnya akhlak dan tanggung jawab dalam pengembangan teknologi. Pemanfaatan AI harus memperhatikan nilai-nilai etika, keadilan, dan tidak menimbulkan kerusakan (mafsadat) di muka bumi. Prinsip *maslahah* (kemanfaatan) dan *la dharar wa la dhirar* (tidak membahayakan dan tidak saling membahayakan) menjadi landasan penting dalam mengevaluasi apakah suatu teknologi, termasuk AI, layak untuk dikembangkan dan diterapkan. Oleh karena itu, umat Islam perlu terlibat aktif dalam pengembangan AI, tidak hanya sebagai pengguna, tetapi juga sebagai pihak yang memberikan arah moral dan etis agar AI berkembang sesuai dengan tujuan kemanusiaan dan nilai-nilai Islam.

- Al-Qur'an. (2008). *Tafsir Al-Mishbah: Pesan, Kesan dan Keserasian Al-Qur'an* (M. Quraish Shihab, Ed.). Lentera Hati. (QS. Al-Mujadilah [58]:11)

- Al-Attas, S. M. N. (1998). *Islam and Secularism*. ISTAC.
- Ahmad, M., & Hussain, S. (2020). “Artificial Intelligence and Islamic Ethics: A Framework for Responsible AI.” *Journal of Islamic Ethics*, 4(2), 115–133.
- Diker, E., & Kaya, A. (2019). “Ethical and Legal Aspects of Artificial Intelligence from an Islamic Perspective.” *Islamic Studies Journal*, 57(1), 75–98.

Etika AI dalam Islam

Dalam Islam, etika penggunaan dan pengembangan AI berlandaskan prinsip *maṣlaḥah* (kemaslahatan) dan *lā ḍarār wa lā ḍirār* (tidak membahayakan dan tidak saling membahayakan). Teknologi AI harus dirancang dan diimplementasikan sedemikian rupa sehingga memberikan manfaat besar—seperti meningkatkan kesejahteraan sosial, keadilan akses, dan peningkatan kualitas hidup—tanpa menimbulkan kerugian, baik secara fisik maupun spiritual (Ahmad & Hussain, 2020). Selain itu, AI hendaknya menghormati hak asasi manusia, termasuk privasi, martabat, dan kebebasan individu, sesuai prinsip Al-Qur’an yang mendorong perlindungan hak setiap insan (QS. Al-Hujurāt [49]:13) dan aturan syariat yang melarang perbuatan zalim (Al-Mā’idah [5]:8).

Aspek transparansi, akuntabilitas, dan keadilan (fairness) juga menjadi pilar utama etika AI dalam perspektif Islam. Pengembang AI wajib menjelaskan logika dan data yang digunakan agar pengguna memahami batas kemampuan dan potensi bias sistem (Diker & Kaya, 2019). Apabila suatu algoritma menimbulkan diskriminasi atau eksklusi, hal ini bertentangan dengan ajaran Islam tentang persamaan hak dan keadilan (‘adl) (Al-An‘ām [6]:152). Oleh karena itu, komunitas Muslim diharapkan berperan aktif dalam menetapkan kerangka regulasi dan pedoman implementasi AI yang tidak hanya mengedepankan inovasi teknis, tetapi juga memprioritaskan nilai-nilai moral dan tanggung jawab sosial sesuai tuntunan syariat.

- Ahmad, M., & Hussain, S. (2020). “Artificial Intelligence and Islamic Ethics: A Framework for Responsible AI.” *Journal of Islamic Ethics*, 4(2), 115–133.
- Diker, E., & Kaya, A. (2019). “Ethical and Legal Aspects of Artificial Intelligence from an Islamic Perspective.” *Islamic Studies Journal*, 57(1), 75–98.

Studi Kasus: Teknologi AI dan Etika Islam

Studi Kasus: Robo-Advisor Halal untuk Investasi Syariah

Pada tahun 2023, sebuah perusahaan fintech di Indonesia meluncurkan layanan **robo-advisor** yang secara otomatis merekomendasikan portofolio investasi sesuai prinsip syariah, dengan memanfaatkan algoritma machine learning untuk memfilter

saham dan instrumen keuangan yang bebas riba (bunga), gharar (ketidakjelasan), serta kegiatan usaha haram. Sistem ini memproses data keuangan emiten secara real-time, menilai kepatuhan syariah berdasarkan kriteria Dewan Syariah Nasional, dan menyesuaikan alokasi investasi pelanggan secara dinamis. Dari perspektif etika Islam, model ini mendukung *maṣlahah* (kemanfaatan) karena mempermudah masyarakat awam mengakses instrumen halal, serta meningkatkan keadilan (*adl*) dengan menyediakan akses layanan investasi dengan biaya rendah.

Namun, tantangan etis muncul terkait **transparansi** dan **akuntabilitas** algoritma. Pelanggan—termasuk kaum milenial Muslim—seringkali tidak memahami bagaimana bobot kriteria syariah diterapkan dalam model, sehingga menimbulkan potensi keraguan terhadap kepatuhan fatwa. Selain itu, jika data pasar yang digunakan mengandung bias (misalnya undervaluasi pada emiten UMKM syariah), rekomendasi robo-advisor dapat secara tidak sengaja mengecualikan perusahaan kecil, bertentangan dengan prinsip *lā ḍarār wa lā ḍirār* (tidak saling membahayakan) yang mendorong pemberdayaan ekonomi umat. Untuk mengatasi ini, perusahaan bermitra dengan ulama dan auditor syariah independen, serta menerbitkan white-paper terbuka yang menjelaskan logika pemilihan dan bobot algoritma, guna memastikan akuntabilitas dan memperkuat kepercayaan pengguna.

- Ahmad, M., & Hussain, S. (2020). “Artificial Intelligence and Islamic Ethics: A Framework for Responsible AI.” *Journal of Islamic Ethics*, 4(2), 115–133.
- Diker, E., & Kaya, A. (2019). “Ethical and Legal Aspects of Artificial Intelligence from an Islamic Perspective.” *Islamic Studies Journal*, 57(1), 75–98.

Latihan dan Evaluasi:

1. Jelaskan ruang lingkup AI dan perannya dalam kehidupan modern.
2. Uraikan bagaimana Islam memandang penggunaan teknologi cerdas.
3. Identifikasi satu studi kasus AI dan analisislah dari perspektif etika Islam.
4. Diskusikan tantangan etis yang muncul dalam pengembangan AI dan bagaimana Islam memberikan panduan dalam menanggapinya.

BAB 2: Representasi Pengetahuan

Deskripsi: Pembahasan teknik representasi pengetahuan sebagai dasar pengambilan keputusan dalam AI.

Tujuan Pembelajaran: Setelah mempelajari bab ini, mahasiswa mampu:

1. Menjelaskan pendekatan deklaratif dan prosedural dalam representasi pengetahuan.

2. Mendeskripsikan berbagai teknik representasi pengetahuan seperti frame, semantic net, dan production rules.
3. Menerapkan ontologi dan logika deskriptif dalam pengorganisasian pengetahuan.
4. Menjelaskan proses representasi fakta dan inferensi dalam sistem AI.

Uraian Materi:

Pendekatan Deklaratif dan Prosedural

Pendekatan Deklaratif dalam representasi pengetahuan menekankan pada “**apa**” yang diketahui, tanpa menyertakan urutan langkah untuk memprosesnya. Model deklaratif biasa menggunakan logika formal (mis. logika predikat pertama), kerangka (frames), atau jaringan semantik (semantic networks) untuk merepresentasikan fakta dan relasi di dunia nyata. Keuntungan utama pendekatan ini adalah kemudahannya dalam **penalaran otomatis**: mesin dapat menarik kesimpulan baru melalui inferensi logika, misalnya modus ponens atau unifikasi, tanpa diprogram ulang. Menurut Russell & Norvig (2021), representasi berbasis logika memfasilitasi verifikasi kebenaran pengetahuan dan mendukung penalaran yang dapat dibuktikan kebenarannya – sebuah kebutuhan penting dalam aplikasi seperti sistem pakar medis atau verifikasi perangkat lunak (Russell & Norvig, 2021).

Sebaliknya, **Pendekatan Prosedural** berfokus pada “**bagaimana**” pengetahuan itu digunakan, dengan menyematkan prosedur atau algoritma langsung di dalam representasi pengetahuan. Contohnya adalah **production rules** (aturan-produksi) dalam sistem pakar klasik, di mana tiap aturan “jika-kondisi maka-aksi” memuat kode yang dijalankan saat kondisi terpenuhi. Begitu pula **scripts** dan **plans** menyertakan struktur prosedural untuk menangani urutan peristiwa atau tindakan. Keunggulan prosedural terletak pada kemampuan kontrol alur kerja yang eksplisit dan efisiensi eksekusi, terutama dalam domain di mana langkah-langkah penyelesaian tugas sudah baku (Buchanan & Shortliffe, 1984). Namun, prosedural sering kali kurang fleksibel untuk penalaran mendalam atau pembuktian formal, karena logika inferensinya tersembunyi di balik kode procedural.

- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Brachman, R. J., & Levesque, H. J. (2004). *Knowledge Representation and Reasoning*. Morgan Kaufmann.

Frame, Semantic Net, dan Production Rules

Frame merupakan struktur data untuk merepresentasikan pengetahuan dalam bentuk objek dan atribut-atributnya (slots). Setiap frame menggambarkan suatu konsep atau entitas dunia nyata, lengkap dengan **default values**, **constraints**, dan **procedural attachment** (metode yang dijalankan saat slot diakses). Misalnya, frame “Mobil” dapat memiliki slot “merk”, “warna”, “kecepatan maksimum”, dan prosedur untuk menghitung waktu tempuh berdasarkan jarak. Pendekatan frame memudahkan

inheritance: frame “Mobil Listrik” dapat mewarisi semua slot dari frame “Mobil” dan menambahkan slot khusus seperti “kapasitas baterai” (Minsky, 1975; Brachman & Levesque, 2004).

Semantic Net (jaringan semantik) merepresentasikan pengetahuan sebagai **graf berlabel**: simpul (nodes) melambangkan konsep atau objek, sedangkan sisi (arcs) menunjukkan relasi antar konsep (misalnya *ISA*, *PART-OF*, *CAUSES*). Jaringan ini intuitif dan memudahkan visualisasi struktur pengetahuan. Contohnya, simpul “Anjing” terhubung via *ISA* ke simpul “Mamalia”, dan via *HAS-PROPERTY* ke simpul “Bersuara: Menggonggong”. Dengan semantic net, mesin dapat melakukan penelusuran graf untuk inferensi sederhana seperti menemukan subclass atau bagian suatu objek (Sowa, 1984; Russell & Norvig, 2021).

Production Rules (aturan-produksi) adalah representasi pengetahuan berbasis **if-then**, di mana setiap aturan terdiri dari **condition** (left-hand side) dan **action** (right-hand side). Sistem pakar klasik seperti **MYCIN** menggunakan ratusan aturan produksi untuk diagnosis medis: misalnya, “IF demam tinggi AND ruam merah, THEN kemungkinan penyakit X”. Mesin inferensi (inference engine) menerapkan strategi seperti **forward chaining** atau **backward chaining** untuk mengeksekusi aturan yang relevan. Kelebihan produksi rules adalah fleksibilitas dalam menambah/mengubah pengetahuan tanpa memodifikasi logika inferensi; namun, skala besar dapat menimbulkan **conflict resolution** yang kompleks (Buchanan & Shortliffe, 1984; Newell & Simon, 1976).

- Minsky, M. (1975). *A framework for representing knowledge*. MIT-AI Laboratory Memo 306.
- Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

Ontologi dan Logika Deskriptif

Ontologi dalam kecerdasan buatan adalah spesifikasi eksplisit mengenai konsep-konsep dalam suatu domain dan hubungan antar-konsep tersebut. Ontologi memungkinkan berbagi dan penggunaan ulang pengetahuan dengan menyediakan struktur formal yang mendukung interoperabilitas antarsistem. Ontologi umumnya ditulis menggunakan bahasa seperti OWL (Web Ontology Language), yang mendasarkan semantik pada logika deskriptif sehingga memungkinkan alat reasoning mendeteksi inkonsistensi, mengklasifikasikan konsep, dan menarik kesimpulan baru secara otomatis. Ontologi banyak diterapkan dalam aplikasi seperti Semantic Web, bioinformatika (misalnya Gene Ontology), dan manajemen pengetahuan perusahaan, di mana pemahaman bersama atas istilah-istilah kunci sangat krusial (Gruber, 1993; Studer, Benjamins, & Fensel, 1998).

Logika Deskriptif (Description Logic, DL) adalah keluarga logika formal yang dirancang untuk merepresentasikan struktur ontologi dengan keseimbangan antara

ekspresivitas dan keterputusan (decidability). DL menyediakan konstruksi logis—seperti konjungsi, disjungsi, eksistensial, dan universal quantifier—untuk membangun konsep kompleks dari konsep dasar. Mesin reasoning berbasis DL, seperti tableau-based reasoners, dapat melakukan subsumption checking (mengecek apakah satu konsep merupakan spesialisasi konsep lain), instance checking, dan consistency checking. Karena sifat decidable-nya, DL menjadi landasan formal bagi OWL-DL, varian OWL yang menjamin hasil reasoning yang lengkap dan benar, esensial untuk aplikasi semantik web dan sistem pakar modern (Baader et al., 2003; Baader, Horrocks, & Sattler, 2008).

- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199–220.
- Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, 25(1–2), 161–197.
- Guarino, N. (1998). Formal ontology in information systems. In N. Guarino (Ed.), *Proceedings of FOIS 1998* (pp. 3–15). IOS Press.
- Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (Eds.). (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.
- Baader, F., Horrocks, I., & Sattler, U. (2008). Description logics as ontology languages for the Semantic Web. In J. Heflin (Ed.), *Semantic Web Technologies* (pp. 228–241). Springer.

Representasi Fakta dan Inferensi

Representasi Fakta dalam kecerdasan buatan merujuk pada cara-cara formal untuk menyimpan informasi dasar tentang dunia agar dapat diolah mesin. Salah satu metode yang paling umum adalah **logika predikat pertama**, di mana fakta dinyatakan sebagai predikat dengan argumen—misalnya `Parent(ali, budi)` menyatakan “Ali adalah orang tua Budi”. Selain itu, **triple RDF** (Resource Description Framework) memodelkan fakta sebagai “subjek-predikat-objek”, misalnya `(Ali, isParentOf, Budi)`, yang banyak digunakan di Semantic Web. Representasi lain mencakup **frame** (objek dengan slot dan nilai), **jaringan semantik** (graf konsep dan relasi), serta **basis fakta** dalam sistem pakar berbasis aturan. Keunggulan logika dan struktur ini adalah kemampuannya untuk memverifikasi konsistensi, mendeteksi duplikasi, serta memudahkan penambahan fakta baru tanpa merombak keseluruhan sistem (Russell & Norvig, 2021; Brachman & Levesque, 2004).

Inferensi adalah proses penarikan kesimpulan baru dari fakta-fakta yang ada menggunakan mekanisme logis. Dalam sistem berbasis aturan, **forward chaining** (rangkainan maju) memulai dari fakta yang diketahui dan menerapkan aturan “jika–maka” untuk menambah fakta hingga tujuan tercapai; sedangkan **backward chaining** (rangkainan mundur) memulai dari hipotesis dan bekerja mundur untuk memeriksa apakah fakta pendukung tersedia. Metode lain, seperti **resolusi** pada logika predikat, menggunakan unifikasi dan aturan refutation untuk membuktikan kebenaran atau mendeteksi kontradiksi. Mesin inferensi pada sistem pakar (inference engine)

mengelola konflik antar-aturan dengan strategi seperti specificity ordering atau priority, memastikan inferensi dilakukan secara efisien dan konsisten (Buchanan & Shortliffe, 1984; Genesereth & Nilsson, 1987).

- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Brachman, R. J., & Levesque, H. J. (2004). *Knowledge Representation and Reasoning*. Morgan Kaufmann.
- Buchanan, B. G., & Shortliffe, E. H. (1984). *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.
- Genesereth, M. R., & Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann.
- Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, 25(1–2), 161–197.

Latihan dan Evaluasi:

1. Jelaskan perbedaan antara pendekatan deklaratif dan prosedural.
 2. Buatlah contoh semantic net untuk menggambarkan hubungan antara konsep "mobil", "kendaraan", dan "mesin".
 3. Tuliskan 3 contoh production rules dalam domain sistem pakar kesehatan.
 4. Jelaskan bagaimana ontologi dapat digunakan dalam pengembangan sistem informasi.
-

BAB 3: Probabilitas Bayes

Deskripsi: Teori probabilistik untuk pengambilan keputusan dan pemodelan ketidakpastian.

Tujuan Pembelajaran: Setelah mempelajari bab ini, mahasiswa mampu:

1. Menjelaskan konsep dasar probabilitas dan Teorema Bayes.
2. Mengimplementasikan algoritma Naïve Bayes untuk klasifikasi.
3. Mendeskripsikan struktur dan fungsi Bayesian Belief Networks.
4. Menerapkan model probabilistik untuk klasifikasi dan prediksi dalam AI.

Uraian Materi:

Teorema Bayes

Bayes' Theorem adalah suatu aturan dalam teori probabilitas yang digunakan untuk menghitung probabilitas bersyarat sebuah peristiwa A diberikan peristiwa B telah terjadi. Secara matematis, ia dirumuskan sebagai:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

di mana

- $P(A)$ adalah probabilitas a priori terjadinya A,
- $P(B)$ adalah probabilitas a priori terjadinya B,
- $P(B|A)$ adalah probabilitas terjadinya B apabila A benar,
- dan $P(A|B)$ adalah probabilitas A setelah diketahui B.

Dengan Teorema Bayes, kita dapat “membalik” kondisi probabilitas—misalnya dari $P(B|A)$ menjadi $P(A|B)$ —yang sangat berguna dalam aplikasi seperti diagnosis medis (menghitung probabilitas penyakit berdasarkan gejala), pemrosesan bahasa alami, atau sistem klasifikasi dalam machine learning.

Referensi

- Bayes, T. (1763). *An Essay towards solving a Problem in the Doctrine of Chances*. Philosophical Transactions of the Royal Society of London, 53, 370–418.

Naïve Bayes Classifier

Naïve Bayes Classifier adalah metode klasifikasi probabilistik yang didasarkan pada Teorema Bayes dengan asumsi *independensi bersyarat* antar fitur. Artinya, untuk suatu kelas C dan fitur-fitur $\{x_1, x_2, \dots, x_n\}$, Naïve Bayes menganggap bahwa setiap x_i bersifat independen satu sama lain jika diketahui C . Probabilitas posterior suatu kelas diberikan observasi fitur dihitung sebagai:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

Meskipun asumsi independensi ini jarang terpenuhi sepenuhnya dalam praktik, Naïve Bayes sering kali memberikan kinerja klasifikasi yang sangat baik, terutama pada data berdimensi tinggi seperti teks dan dokumen.

Naïve Bayes banyak digunakan dalam aplikasi seperti **spam filtering**, **analisis sentimen**, dan **klasifikasi dokumen**, di mana estimasi probabilitas sederhana dan efisien menjadi keunggulannya. Terdapat beberapa varian, misalnya Gaussian Naïve Bayes (untuk fitur kontinu yang diasumsikan berdistribusi normal), Multinomial Naïve Bayes (untuk data hitung/frekuensi, seperti kata dalam teks), dan Bernoulli Naïve Bayes (untuk fitur biner). Pelatihan model hanya memerlukan penghitungan frekuensi atau statistik sederhana untuk $P(C)P(C)P(C)$ dan $P(x_i|C)P(x_i|C)P(x_i|C)$, sehingga Naïve Bayes sangat cepat dibangun dan skalabel ke dataset besar.

- Domingos, P., & Pazzani, M. (1997). “On th

Domingos, P., & Pazzani, M. (1997). “On the optimality of the simple Bayesian classifier under zero–one loss.” *Machine Learning*, 29(2–3), 103–130.

Bayesian Belief Networks (BBN)

Bayesian Belief Networks (BBN), juga dikenal sebagai *Bayesian networks* atau *probabilistic graphical models*, adalah model representasi probabilistik yang menggambarkan sekumpulan variabel acak beserta ketergantungan kausal atau kondisional di antara mereka melalui struktur graf berarah tanpa siklus (DAG). Setiap simpul dalam graf mewakili satu variabel, dan setiap busur terarah mengindikasikan hubungan probabilistik—umumnya kausal—dari variabel induk ke variabel anak. Untuk setiap simpul, ditentukan *conditional probability table* (CPT) yang menyatakan $P(X|\text{Parents}(X))$. Struktur ini memungkinkan pemecahan kompleksitas representasi distribusi multivariat: jaringan yang terstruktur secara lokal menurunkan kebutuhan menyimpan dan menghitung probabilitas secara eksponensial besar menjadi komputasi yang terfaktorasi sesuai konektivitas graf—pendekatan yang sangat efisien untuk domain dengan banyak variabel saling terkait (Pearl, 1988; Koller & Friedman, 2009).

Inferensi dalam BBN dapat dilakukan dengan berbagai algoritma, seperti *exact inference* (misalnya variable elimination, junction tree) dan *approximate inference* (misalnya Monte Carlo sampling, loopy belief propagation), yang memungkinkan perhitungan $P(X|e)$ untuk variabel kueri X given evidence e (Jensen & Nielsen, 2007). Bayesian networks banyak diterapkan dalam diagnosis medis, peramalan kegagalan mesin, sistem pakar, dan analisis risiko keuangan, karena kemampuannya untuk menggabungkan pengetahuan domain (struktur kausal) dengan data empiris (parameter probabilitas), serta menangani ketidakpastian secara eksplisit. Selain itu, BBN memberikan kerangka formal untuk pembelajaran struktur dan parameter dari data, memungkinkan otomatisasi pembangunan model probabilistik yang mencerminkan pola kausal dalam dataset besar (Heckerman, 1995; Russell & Norvig, 2021).

- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.
- Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

- Jensen, F. V., & Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs* (2nd ed.). Springer.

Aplikasi dalam Klasifikasi dan Prediksi

Dalam kecerdasan buatan, **klasifikasi** adalah tugas menetapkan label diskrit ke instance data—misalnya, menentukan apakah sebuah email merupakan spam atau bukan, mengidentifikasi jenis tumor sebagai jinak atau ganas, atau mengenali tulisan tangan sebagai angka tertentu. Berbagai algoritma seperti **Naïve Bayes**, **Support Vector Machines (SVM)**, **Decision Trees**, dan **Random Forests** banyak digunakan karena kemampuannya mempelajari pola dari data berlabel dan menggeneralisasi ke data baru (Murphy, 2012; Russell & Norvig, 2021). Di bidang kesehatan, misalnya, model klasifikasi dapat memproses citra radiologi untuk mendeteksi kelainan dini; di bidang keamanan siber, klasifikasi membantu mendeteksi serangan atau malware; sedangkan dalam e-commerce, sistem rekomendasi produk sering kali dibangun di atas model klasifikasi perilaku pengguna agar dapat menyajikan barang yang relevan.

Sementara itu, **prediksi** (regression) berkaitan dengan peramalan nilai kontinu berdasarkan variabel input—seperti meramalkan harga saham, permintaan energi, atau jumlah pelanggan di masa mendatang. Algoritma seperti **Linear Regression**, **Decision Tree Regression**, dan **Neural Networks** (terutama arsitektur LSTM untuk data urutan) digunakan untuk menangkap hubungan kompleks antar variabel (Bishop, 2006; Goodfellow, Bengio, & Courville, 2016). Aplikasi prediksi meliputi peramalan cuaca, pengelolaan rantai pasok melalui prediksi permintaan produk, hingga pemeliharaan prediktif di industri manufaktur yang memanfaatkan sensor IoT untuk memperkirakan kegagalan mesin sebelum terjadi kerusakan serius. Dengan memadukan teknik klasifikasi dan prediksi, AI kini memungkinkan sistem cerdas yang tidak hanya mengenali pola masa lalu, tetapi juga memberikan gambaran proaktif untuk pengambilan keputusan yang lebih baik di berbagai domain.

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

Latihan dan Evaluasi:

1. Jelaskan konsep Teorema Bayes dan berikan contoh penerapannya.
2. Implementasikan algoritma Naïve Bayes untuk klasifikasi dataset sederhana.
3. Gambarkan struktur Bayesian Belief Network untuk kasus diagnosa penyakit.

4. Bandingkan kelebihan dan keterbatasan Naïve Bayes dibandingkan metode klasifikasi lain.

BAB 4: Logika Fuzzy

Deskripsi: Pengenalan logika fuzzy dan aplikasinya dalam sistem cerdas.

Tujuan Pembelajaran: Setelah mempelajari bab ini, mahasiswa mampu:

1. Menjelaskan konsep dasar logika fuzzy dan perbedaannya dengan logika klasik.
2. Menggunakan fuzzy sets dan aturan fuzzy dalam representasi pengetahuan.
3. Merancang sistem inferensi fuzzy dan menerapkan proses defuzzifikasi.
4. Mengimplementasikan logika fuzzy dalam sistem kontrol dan pengambilan keputusan.

Fuzzy Sets dan Fuzzy Rules

Fuzzy Sets adalah perluasan dari teori himpunan klasik yang diperkenalkan oleh Lotfi A. Zadeh (1965) untuk menangani ketidakpastian dan ketidakjelasan (vagueness) dalam representasi pengetahuan. Pada himpunan klasik, suatu elemen **a** hanya memiliki

keanggotaan biner—“ya” (1) atau “tidak” (0)—dalam suatu himpunan AAA. Sebaliknya, pada **fuzzy set** $A^{\wedge}$, setiap elemen xxx memiliki derajat keanggotaan $\mu_{A^{\wedge}}(x)$ dalam rentang $[0,1]$, di mana nilai mendekati 1 menunjukkan keanggotaan kuat dan nilai mendekati 0 menunjukkan keanggotaan lemah. Fungsi keanggotaan ini (membership function) dapat berbentuk segitiga, trapesium, Gaussian, atau bentuk lain yang sesuai untuk memodelkan konsep linguistik seperti “tinggi”, “sedang”, atau “rendah” dalam sistem kontrol maupun pengambilan keputusan

Fuzzy Rules (aturan fuzzy) memanfaatkan fuzzy sets untuk merumuskan penalaran berbasis bahasa alami melalui format “IF–THEN”. Misalnya:

```

vbnet
SalinEdit
IF suhu IS tinggi AND kelembapan IS rendah
THEN kecepatan_kipas IS cepat

```

Di sini, kondisi “suhu IS tinggi” dan “kelembapan IS rendah” dievaluasi dengan derajat keanggotaan pada fungsi membership masing-masing, lalu digabungkan (biasanya dengan operator *min* untuk AND atau *max* untuk OR). Hasil inferensi ini kemudian di-*defuzzification*—mengubah nilai fuzzy kembali menjadi sinyal kontrol konkret, misalnya nilai PWM untuk kipas. Pendekatan ini memungkinkan sistem mengolah informasi yang tidak pasti secara mirip dengan pemikiran manusia, sehingga banyak diaplikasikan pada kontrol mesin industri, peralatan elektronik (AC, kamera), dan sistem pakar fuzzy (Zadeh, 1965; Mamdani & Assilian, 1975).

- Zadeh, L. A. (1965). “Fuzzy Sets.” *Information and Control*, 8(3), 338–353.
- Mamdani, E. H., & Assilian, S. (1975). “An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller.” *International Journal of Man–Machine Studies*, 7(1), 1–13.
- Ross, T. J. (2010). *Fuzzy Logic with Engineering Applications* (3rd ed.). Wiley.
- Klir, G. J., & Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall.

Fungsi Keanggotaan

Fungsi Keanggotaan (Membership Function) dalam logika fuzzy adalah fungsi matematis yang memetakan setiap elemen xxx dari himpunan semesta XXX ke derajat keanggotaan $\mu_{A^{\wedge}}(x)$ dalam rentang $[0,1]$. Derajat ini mencerminkan seberapa kuat xxx mewakili konsep linguistik tertentu—misalnya “tinggi”, “sedang”, atau “rendah”. Bentuk umum fungsi keanggotaan mencakup:

- **Segitiga (Triangular):**

$$\mu(x) = \begin{cases} 0, & x \leq a \text{ or } x \geq c, \\ \frac{x-a}{b-a}, & a < x < b, \\ \frac{c-x}{c-b}, & b \leq x < c, \end{cases}$$

dengan parameter (a,b,c)(a,b,c)(a,b,c) menentukan titik awal, puncak, dan akhir¹.

- **Trapezium (Trapezoidal):**

$$\mu(x) = \begin{cases} 0, & x \leq a \text{ or } x \geq d, \\ \frac{x-a}{b-a}, & a < x < b, \\ 1, & b \leq x \leq c, \\ \frac{d-x}{d-c}, & c < x < d, \end{cases}$$

dengan empat parameter (a,b,c,d)(a,b,c,d)(a,b,c,d) untuk mengatur landai naik, plateau, dan landai turun².

- **Gaussian:**

$$\mu(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

di mana c adalah mean (pusat) dan σ adalah simpangan baku yang mengontrol lebar bell-curve³.

Pemilihan bentuk fungsi keanggotaan tergantung pada kebutuhan aplikasi—simplen dan cepat hitung seperti segitiga/trapesium, atau halus dan diferensiabel seperti Gaussian—untuk memodelkan ketidakjelasan konsep secara tepat.

1. Zadeh, L. A. (1965). "Fuzzy Sets." *Information and Control*, 8(3), 338–353.
2. Klir, G. J., & Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall.
3. Ross, T. J. (2010). *Fuzzy Logic with Engineering Applications* (3rd ed.). Wiley.

Inferensi Fuzzy dan Defuzzifikasi

Inferensi Fuzzy adalah proses penalaran berbasis aturan fuzzy yang mengubah input crisp menjadi output fuzzy sebelum akhirnya di-defuzzifikasi. Secara garis besar, langkah-langkahnya meliputi:

1. **Fuzzifikasi:** Mengubah nilai input crisp menjadi derajat keanggotaan pada setiap himpunan fuzzy terkait menggunakan fungsi keanggotaan.
2. **Evaluasi Aturan (Rule Evaluation):** Menerapkan setiap aturan IF–THEN dengan menggabungkan derajat keanggotaan kondisi (antecedent) menggunakan operator

fuzzy (biasanya *min* untuk AND, *max* untuk OR) untuk mendapatkan derajat firing (kekuatan) masing-masing aturan.

3. **Kompilasi Konsekuen (Aggregation):** Menggabungkan efek dari semua aturan yang menyasar himpunan output yang sama menjadi satu himpunan fuzzy hasil dengan memilih nilai maksimum atau metode lain untuk menyatukan kontribusi aturan-aturan tersebut.
Metode inferensi yang paling umum digunakan adalah **Mamdani**—yang mengaplikasikan min–max untuk implication dan max untuk agregasi—serta **Sugeno**—yang menggunakan fungsi output berbentuk polinomial sederhana sehingga menghasilkan inferensi yang lebih mudah di-defuzzifikasi.

Defuzzifikasi adalah langkah terakhir yang mentransformasikan himpunan fuzzy hasil inferensi menjadi nilai crisp yang dapat digunakan sistem kontrol atau pengambilan keputusan. Beberapa metode defuzzifikasi yang sering diaplikasikan antara lain:

- **Centroid (Center of Gravity):** Menghitung titik pusat area di bawah kurva keanggotaan hasil agregasi:

$$y^* = \frac{\int y \mu_{\text{out}}(y) dy}{\int \mu_{\text{out}}(y) dy}$$

- **Mean of Maxima (MOM):** Rata-rata dari semua nilai y yang memiliki derajat keanggotaan maksimum.
- **Bisector:** Titik nilai y yang membagi area di bawah kurva hasil menjadi dua bagian sama besar.
- **Smallest/Largest of Maxima:** Memilih nilai terkecil atau terbesar dari semua y dengan derajat keanggotaan maksimum.
Pemilihan metode defuzzifikasi berdampak langsung pada respons sistem; misalnya, centroid memberikan keluwesan tetapi lebih mahal komputasinya, sedangkan MOM atau bisector lebih cepat namun kadang kurang presisi dalam kasus kurva yang kompleks.
- Mamdani, E. H., & Assilian, S. (1975). "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller." *International Journal of Man–Machine Studies*, 7(1), 1–13.
- Ross, T. J. (2010). *Fuzzy Logic with Engineering Applications* (3rd ed.). Wiley.
- Klir, G. J., & Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall.
- Zadeh, L. A. (1973). "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes." *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(1), 28–44.

Aplikasi dalam Sistem Kontrol dan Pengambilan Keputusan

Dalam sistem kendali, AI memungkinkan perancangan **controller cerdas** yang dapat menyesuaikan parameter kontrol secara real time berdasarkan umpan balik lingkungan. Misalnya, **Pengendali Fuzzy** (Fuzzy Controller) memanfaatkan aturan linguistik untuk mengatur variabel proses—seperti suhu, kecepatan motor, atau tekanan—tanpa memerlukan model matematis yang rumit, sehingga banyak diaplikasikan pada AC, mesin industri, dan kendaraan otonom (Mamdani & Assilian, 1975; Ross, 2010). Di sisi lain, **Reinforcement Learning (RL)**—dengan algoritma seperti Q-Learning dan Deep Q-Networks—mengoptimalkan kebijakan kontrol melalui trial-and-error, memungkinkan robot dan sistem otomatis belajar mengatasi dinamika kompleks tanpa pemrograman eksplisit (Sutton & Barto, 2018; Lillicrap et al., 2016). Keuntungan utama AI dalam kendali adalah kemampuannya beradaptasi terhadap perubahan kondisi dan gangguan, serta mengotomatisasi tuning parameter kontrol untuk performa optimal.

Dalam pengambilan keputusan, AI diimplementasikan melalui **Decision Support Systems (DSS)** dan **Expert Systems** yang mengintegrasikan basis pengetahuan, mesin inferensi, dan antarmuka pengguna untuk membantu manusia membuat keputusan kritis. Contohnya, sistem pakar medis MYCIN menggunakan aturan produksi untuk merekomendasikan terapi berdasarkan gejala pasien (Buchanan & Shortliffe, 1984), sedangkan **Bayesian Decision Networks** memodelkan ketidakpastian dan trade-off antara risiko dan manfaat dalam perencanaan keuangan atau manajemen risiko (Pearl, 1988; Koller & Friedman, 2009). Dengan memanfaatkan teknik seperti **optimisasi multi-kriteria** dan **analisis what-if**, AI dapat mempercepat proses evaluasi alternatif dan memberikan rekomendasi yang terukur, akuntabel, dan transparan kepada pengambil keputusan di berbagai domain—mulai dari bisnis, kesehatan, hingga pemerintahan.

- Buchanan, B. G., & Shortliffe, E. H. (1984). *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley.
- Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.
- Lillicrap, T. P., et al. (2016). "Continuous Control with Deep Reinforcement Learning." *arXiv preprint arXiv:1509.02971*.

Latihan dan Evaluasi:

1. Jelaskan perbedaan antara logika klasik dan logika fuzzy.
2. Buatlah tiga fuzzy rule untuk sistem kontrol suhu.
3. Gambarkan fungsi keanggotaan fuzzy untuk variabel "suhu".

4. Implementasikan sistem fuzzy sederhana menggunakan metode Mamdani dan centroid defuzzification.

BAB 5: Version Space

Deskripsi: Pembelajaran konsep-konsep dalam pencarian hipotesis dengan pendekatan version space.

Tujuan Pembelajaran: Setelah mempelajari bab ini, mahasiswa mampu:

1. Menjelaskan konsep version space dan representasi hipotesis dalam pembelajaran mesin.
2. Membedakan generalisasi dan spesialisasi dalam proses pembelajaran.
3. Mengimplementasikan algoritma Candidate Elimination untuk pencarian hipotesis.

4. Menganalisis kelebihan dan keterbatasan pendekatan version space dalam pembelajaran konsep.

Uraian Materi:

Definisi Version Space

Version Space adalah himpunan dari semua hipotesis dalam ruang hipotesis H yang **konsisten dengan data pelatihan** D . Artinya, semua hipotesis dalam Version Space tidak membuat kesalahan pada data pelatihan yang diberikan.

$$VS_{H,D} = \{h \in H \mid \forall (x,y) \in D, h(x) = y\}$$

Version Space berguna untuk **mempersempit pencarian** terhadap hipotesis yang benar seiring bertambahnya data pelatihan.

Representasi Hipotesis

Hipotesis direpresentasikan dalam bentuk **kondisi-kondisi atribut** terhadap data.

Contoh Representasi:

Misalkan data adalah tentang buah dengan atribut:

- Warna: {merah, kuning, hijau}
- Ukuran: {besar, kecil}
- Bentuk: {bulat, lonjong}

Contoh hipotesis:

- $h_1 = (\text{merah}, ?, \text{bulat})$
 - Artinya: semua buah berwarna merah dan berbentuk bulat (ukuran tidak penting).
- $h_2 = (?, ?, ?)$
 - Artinya: menerima semua buah (hipotesis paling umum).
- $h_3 = (\text{merah}, \text{kecil}, \text{lonjong})$
 - Hipotesis paling spesifik.

Simbol yang umum:

- $?$ → menerima semua nilai atribut (general)
- nilai spesifik (contoh: "merah") → membatasi (spesifik)

Generalisasi dan Spesialisasi

Untuk memperbarui Version Space, kita menggunakan dua operasi penting:

a. Generalisasi

- Membuat hipotesis **lebih umum**.
- Contoh: dari $(\text{merah}, \text{kecil}, \text{bulat})$ menjadi $(?, \text{kecil}, \text{bulat})$

b. Spesialisasi

- Membuat hipotesis **lebih spesifik**.
- Contoh: dari $(?, ?, \text{bulat})$ menjadi $(\text{merah}, ?, \text{bulat})$

Tujuannya:

- Menyesuaikan hipotesis agar tetap **konsisten dengan data pelatihan baru**.
- Mengeliminasi hipotesis yang terlalu umum (tidak cocok dengan contoh negatif) atau terlalu spesifik (tidak cocok dengan contoh positif).

Algoritma Candidate Elimination

Tujuan:

Mempersempit ruang hipotesis dengan menyimpan dua batas:

- **G (General boundary)**: himpunan hipotesis paling umum yang masih konsisten.
- **S (Specific boundary)**: himpunan hipotesis paling spesifik yang masih konsisten.

Langkah-langkah Algoritma:

1. **Inisialisasi:**
 - G diisi dengan hipotesis paling umum $(? ? ? \dots ?)$
 - S diisi dengan hipotesis paling spesifik $(\emptyset \emptyset \emptyset \dots \emptyset)$
2. **Untuk setiap contoh pelatihan (x, y):**
 - Jika **positif**:
 - Hapus hipotesis di G yang tidak mencakup xxx
 - Ubah S agar mencakup xxx, lalu **generalisasi** S seminimal mungkin agar konsisten
 - Hapus hipotesis di S yang **tidak lebih spesifik** dari hipotesis lain di S

- Jika **negatif**:
 - Hapus hipotesis di S yang mencakup xxx
 - Ubah G agar tidak mencakup xxx, lalu **spesialisasi** seminimal mungkin agar konsisten
 - Hapus hipotesis di G yang **tidak lebih umum** dari hipotesis lain di G
3. **Ulangi** sampai semua data pelatihan diproses.

Contoh Ilustrasi:

Misalkan data pelatihan:

- Positif: (merah,kecil,bulat)(\text{merah}, \text{kecil}, \text{bulat})(merah,kecil,bulat)
- Negatif: (kuning,kecil,bulat)(\text{kuning}, \text{kecil}, \text{bulat})(kuning,kecil,bulat)

Algoritma akan memperbarui:

- $G \rightarrow$ mengecualikan hipotesis yang menyetujui contoh negatif
- $S \rightarrow$ memperumum hipotesis agar mencakup contoh positif

Referensi Terpercaya

1. **Mitchell, T. M. (1997).** *Machine Learning*. McGraw-Hill.
2. **Russell, S., & Norvig, P. (2021).** *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
3. **Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983).** *Machine Learning: An Artificial Intelligence Approach*. Springer.

Latihan dan Evaluasi:

1. Jelaskan perbedaan antara generalisasi dan spesialisasi dalam version space.
2. Berikan contoh 5 hipotesis dan evaluasi konsistensinya terhadap 3 data pelatihan.
3. Simulasikan proses Candidate Elimination untuk dataset kecil.
4. Diskusikan kelebihan dan kekurangan pendekatan version space dalam sistem pembelajaran berbasis data.

BAB 6: Sistem Pakar

Deskripsi: Sistem pakar sebagai aplikasi utama AI untuk simulasi keputusan pakar.

Tujuan Pembelajaran: Setelah mempelajari bab ini, mahasiswa mampu:

1. Menjelaskan konsep dan komponen utama sistem pakar.
2. Mendeskripsikan arsitektur dan alur kerja sistem pakar.
3. Mengembangkan basis pengetahuan dan mesin inferensi sederhana.
4. Mengimplementasikan sistem pakar menggunakan bahasa CLIPS atau Prolog.

Uraian Materi:

Pengertian Sistem Pakar

Sistem Pakar (Expert System) adalah sistem komputer yang **meniru kemampuan pengambilan keputusan seorang pakar manusia** dalam suatu domain tertentu. Sistem ini dirancang untuk menyelesaikan masalah kompleks dengan **penalaran logis berdasarkan pengetahuan yang telah diprogramkan**.

Arsitektur Sistem Pakar

Arsitektur sistem pakar secara umum terdiri dari beberapa **komponen utama** yang bekerja sama, yaitu:

1. Antarmuka Pengguna (User Interface)

- Berfungsi sebagai **penghubung antara pengguna dan sistem pakar**.
- Pengguna memasukkan informasi atau pertanyaan, dan sistem memberikan saran atau jawaban.

2. Basis Pengetahuan (Knowledge Base)

- Menyimpan **fakta dan aturan (rules)** dari domain tertentu.
- Contoh format aturan:
IF kondisi THEN kesimpulan.

Contoh:

IF suhu > 38°C AND batuk = ya THEN kemungkinan = demam

3. Mesin Inferensi (Inference Engine)

- Merupakan "**otak**" **sistem pakar** yang:
 - Menginterpretasikan dan mengevaluasi fakta
 - Melakukan penalaran logis berdasarkan aturan dalam basis pengetahuan
- Dua pendekatan umum:
 - **Forward chaining** (penalaran maju)
 - **Backward chaining** (penalaran mundur)

4. Basis Fakta (Fact Base / Working Memory)

- Menyimpan data atau informasi yang diberikan oleh pengguna.
- Data ini digunakan oleh mesin inferensi untuk menarik kesimpulan.

5. Modul Penjelas (Explanation Facility)

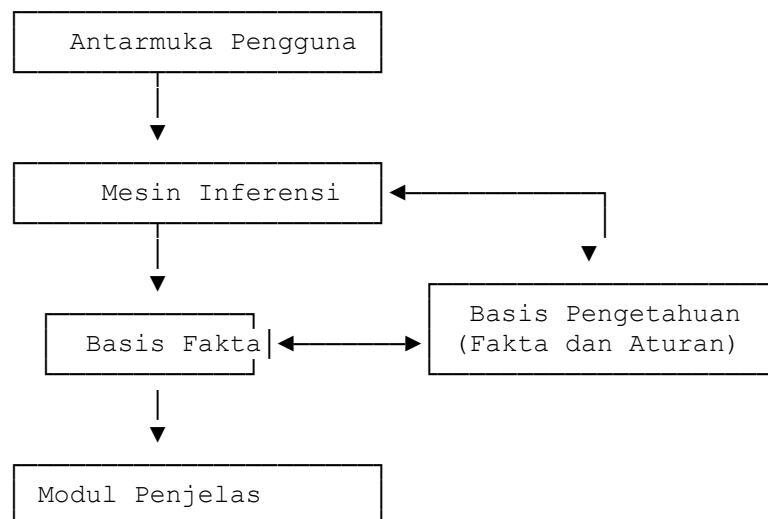
- Menjelaskan **alasan di balik keputusan atau kesimpulan** yang diambil oleh sistem.
- Menjawab pertanyaan seperti "Mengapa sistem mengambil kesimpulan itu?"

6. Modul Akuisisi Pengetahuan (Knowledge Acquisition)

- Memungkinkan penambahan, pengeditan, atau penghapusan pengetahuan dari basis pengetahuan.
- Bisa dilakukan oleh pakar atau knowledge engineer.

Diagram Umum Arsitektur Sistem Pakar

markdown
SalinEdit



3. Referensi Buku dan Literatur

1. **Durkin, J. (1994).** *Expert Systems: Design and Development*. Macmillan Publishing.
 - Referensi utama tentang desain arsitektur sistem pakar secara komprehensif.
2. **Turban, E., Aronson, J. E., & Liang, T. P. (2005).** *Decision Support Systems and Intelligent Systems* (7th ed.). Pearson.
 - Menjelaskan sistem pakar dalam konteks sistem pendukung keputusan.

Latihan dan Evaluasi:

1. Jelaskan arsitektur dasar sistem pakar.
 2. Buatlah 3 aturan IF-THEN yang dapat digunakan dalam sistem pakar akademik.
 3. Rancang alur inferensi dari sistem pakar diagnosa penyakit sederhana.
 4. Bangun sistem pakar sederhana menggunakan CLIPS atau Prolog.
-

BAB 7: Neural Network

Deskripsi: Konsep dan arsitektur jaringan saraf tiruan serta aplikasinya.

Tujuan Pembelajaran:

1. Memahami prinsip kerja jaringan saraf tiruan.
2. Menjelaskan arsitektur perceptron dan multilayer perceptron (MLP).
3. Mengimplementasikan algoritma backpropagation.
4. Mengenal dasar CNN dan RNN.

Perceptron dan MLP

Perceptron adalah **model jaringan saraf tiruan (neural network)** paling sederhana yang digunakan untuk **klasifikasi linier**. Diperkenalkan oleh **Frank Rosenblatt** pada tahun 1958, perceptron meniru cara kerja neuron biologis untuk memproses input dan menghasilkan output biner (0 atau 1).

Struktur Perceptron:

Perceptron tunggal memiliki:

- **Input:** $x_1, x_2, \dots, x_n$
- **Bobot (weight):** $w_1, w_2, \dots, w_n$
- **Fungsi aktivasi** (biasanya fungsi langkah/heaviside)

Rumus dasar:

$$y = \begin{cases} 1, & \text{jika } \sum_{i=1}^n w_i x_i + b > 0 \\ 0, & \text{lainnya} \end{cases}$$

Atau dalam bentuk umum:

$$y = f(w \cdot x + b)$$

Di mana f adalah fungsi aktivasi (step function).

Kelebihan dan Keterbatasan Perceptron:

Kelebihan	Keterbatasan
Sederhana dan cepat	Hanya bisa menyelesaikan masalah linier (misalnya OR, AND)
Mudah diimplementasikan	Tidak bisa memproses fungsi non-linier seperti XOR

Multilayer Perceptron (MLP)

Definisi:

Multilayer Perceptron (MLP) adalah pengembangan dari perceptron tunggal dengan menambahkan **satu atau lebih lapisan tersembunyi (hidden layers)**, sehingga mampu mempelajari dan memetakan **hubungan non-linear**.

MLP adalah jenis jaringan saraf **feedforward** yang dilatih menggunakan algoritma **backpropagation**.

Struktur MLP:

MLP biasanya terdiri dari:

1. **Input Layer**
2. **Hidden Layer(s)**
3. **Output Layer**

Setiap neuron di satu lapisan **terhubung ke semua neuron di lapisan berikutnya** (fully connected).

Fungsi Aktivasi Umum dalam MLP:

- **ReLU** (Rectified Linear Unit): $f(x) = \max(0, x)$
 - **Sigmoid**: $f(x) = \frac{1}{1+e^{-x}}$
 - **Tanh**: $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
-

Proses Pembelajaran:

1. **Forward Propagation** – data masuk, dihitung hingga output.
 2. **Backward Propagation** – error dihitung dan bobot diperbarui untuk mengurangi kesalahan.
-

Keunggulan MLP dibanding Perceptron:

Perceptron	MLP
Hanya satu lapisan	Multi-layer (lebih kompleks)
Hanya bisa klasifikasi linier	Bisa menangani data non-linier (misalnya XOR)
Tidak pakai backpropagation	Menggunakan backpropagation untuk pelatihan

1. **Haykin, S. (2009).** *Neural Networks and Learning Machines* (3rd ed.). Pearson.
2. **Goodfellow, I., Bengio, Y., & Courville, A. (2016).** *Deep Learning*. MIT Press.

Backpropagation

Backpropagation (Propagation Balik) adalah algoritma yang digunakan untuk melatih jaringan saraf tiruan (Artificial Neural Network/ANN), terutama jaringan **feedforward multilayer**. Tujuan utama dari backpropagation adalah meminimalkan kesalahan output jaringan dengan cara **menyesuaikan bobot** koneksi antar neuron melalui perhitungan **gradien dari fungsi loss (fungsi kerugian)**.

Algoritma ini bekerja dengan dua fase utama:

1. **Forward Pass:** Data input dimasukkan ke jaringan, dan dihitung output hingga lapisan terakhir.
2. **Backward Pass (Backpropagation):** Kesalahan (error) antara output aktual dan target dihitung, lalu disebarkan mundur melalui jaringan. Dalam proses ini, bobot-bobot diperbarui menggunakan **metode gradient descent** untuk meminimalkan error.

Langkah-langkah Backpropagation

1. **Inisialisasi bobot secara acak.**
2. **Hitung output jaringan (forward propagation).**
3. **Hitung error:**

$$E = \frac{1}{2} \sum (t - o)^2$$

di mana t = target dan o = output jaringan.

4. **Backpropagate error** ke setiap neuron:
 - Hitung delta ($\partial E / \partial \text{net}$) untuk setiap neuron.
 - Perbarui bobot dengan rumus:

$$w_{ij} = w_{ij} - \eta \cdot \frac{\partial E}{\partial w_{ij}}$$

di mana η adalah **learning rate**.

5. **Ulangi proses** sampai error cukup kecil atau mencapai jumlah epoch maksimum.

Contoh Sederhana

Misalkan jaringan memiliki 2 neuron input, 1 neuron hidden, dan 1 neuron output. Jika error pada output diketahui, maka error ini dihitung balik ke neuron hidden untuk memperbaiki bobot secara bertahap, agar pada iterasi berikutnya output mendekati target.

Kelebihan dan Kekurangan Backpropagation

Kelebihan:

- Dapat melatih jaringan dengan banyak lapisan (deep learning).

- Efisien untuk dataset besar.

Kekurangan:

- Dapat terjebak di **local minima**.
 - Sensitif terhadap pemilihan learning rate dan inisialisasi bobot.
-

Referensi

1. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). **Learning representations by back-propagating errors**. *Nature*, 323(6088), 533–536. <https://doi.org/10.1038/32>

Latihan dan Evaluasi:

1. Jelaskan struktur dasar MLP.
 2. Implementasikan backpropagation sederhana.
 3. Bandingkan CNN dan RNN dari sisi arsitektur dan aplikasi.
-

BAB 8: Algoritma Genetika

Deskripsi: Algoritma evolusioner untuk optimasi dan pencarian solusi.

Tujuan Pembelajaran:

1. Memahami prinsip algoritma genetika.
2. Mendesain representasi kromosom.
3. Mengaplikasikan seleksi, crossover, dan mutasi.
4. Menyusun fungsi fitness untuk evaluasi solusi.

Uraian Materi:**Materi Algoritma Genetika**

Kromosom dan Populasi Awal

Kromosom dalam Algoritma Genetika (AG) adalah representasi solusi potensial terhadap suatu permasalahan. Biasanya direpresentasikan dalam bentuk string biner (0 dan 1), tetapi bisa juga berupa bilangan real atau karakter lainnya tergantung konteks masalah.

Contoh representasi kromosom:

- Masalah penjadwalan: kromosom = urutan pekerjaan (P1, P2, P3,...)
- Masalah optimasi: kromosom = string biner 10110011

Populasi awal adalah kumpulan kromosom yang dibangkitkan secara acak atau dengan metode tertentu sebagai solusi awal. Populasi ini akan mengalami evolusi melalui generasi untuk mendapatkan solusi terbaik.

Seleksi, Crossover, dan Mutasi

Seleksi

Seleksi bertujuan untuk memilih kromosom yang akan dikawinkan (reproduksi). Kromosom yang memiliki nilai fitness lebih tinggi memiliki peluang lebih besar untuk terpilih.

Metode seleksi umum:

- Roulette Wheel Selection
- Tournament Selection
- Rank Selection

Crossover (Penyilangan)

Crossover adalah proses pertukaran gen antara dua induk (parent) untuk menghasilkan keturunan (offspring) baru.

Jenis-jenis crossover:

- Single-point crossover
- Two-point crossover
- Uniform crossover

Mutasi

Mutasi dilakukan dengan mengubah satu atau beberapa gen dalam kromosom secara acak untuk menjaga keberagaman genetik dan menghindari stagnasi di local optimum.

Contoh mutasi biner:

Kromosom sebelum: 101010

Setelah mutasi: 101110 (bit ke-4 berubah)

Fungsi Fitness

Fungsi fitness (fitness function) mengukur kualitas solusi (kromosom) terhadap masalah yang dipecahkan. Semakin baik solusi, semakin tinggi nilai fitness-nya.

Contoh fungsi fitness:

Jika memaksimalkan fungsi $f(x) = x^2$, dan kromosom adalah representasi dari x , maka fitness-nya bisa langsung menggunakan nilai $f(x)$.

Ciri fungsi fitness yang baik:

- Merepresentasikan tujuan optimasi secara akurat
- Memberikan perbedaan yang cukup signifikan antar individu
- Menghindari nilai ekstrem yang membuat seleksi tidak adil

Aplikasi Algoritma Genetika

Algoritma Genetika digunakan di berbagai bidang karena kemampuannya menyelesaikan masalah optimasi kompleks.

Contoh aplikasi:

- **Penjadwalan tugas** (scheduling): seperti pada sistem produksi atau penjadwalan dosen
- **Optimasi rute**: seperti *Traveling Salesman Problem (TSP)*
- **Desain jaringan**: optimasi topologi jaringan komputer
- **Pemodelan keuangan**: optimasi portofolio
- **AI dalam game**: pengembangan perilaku karakter non-pemain (NPC)

Referensi

1. Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
2. Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press.

Latihan dan Evaluasi:

1. Desain kromosom untuk kasus optimasi.

2. Simulasikan proses evolusi sederhana.
 3. Implementasikan satu studi kasus penggunaan algoritma genetika.
-

BAB 9: Natural Language Processing

Deskripsi: Pengolahan bahasa alami untuk pemahaman dan produksi bahasa manusia.

Tujuan Pembelajaran:

1. Memahami proses tokenisasi, stemming, POS tagging.
2. Melakukan parsing dan analisis semantik.
3. Menggunakan word embeddings.
4. Menerapkan NLP pada aplikasi nyata seperti chatbot.

Uraian Materi:

Tokenisasi, Stemming, dan POS Tagging

Tokenisasi

Tokenisasi adalah proses memecah teks menjadi unit-unit kecil yang disebut *token*. Token bisa berupa kata, frasa, atau simbol.

- **Contoh:**

Teks: "Saya belajar NLP." → Token: ["Saya", "belajar", "NLP", "."]

Stemming

Stemming adalah proses mengubah kata ke bentuk dasarnya dengan menghapus imbuhan.

- **Contoh:**

"berlari", "lari-lari", "pelari" → "**lari**"

- Algoritma populer:

- Porter Stemmer (untuk bahasa Inggris)
- Nazief–Adriani (untuk bahasa Indonesia)

POS Tagging (Part-of-Speech Tagging)

POS Tagging adalah proses menentukan kelas kata (kata benda, kata kerja, adjektiva, dsb) untuk setiap token.

- **Contoh:**

"Saya makan nasi." → Saya/PRP makan/VB nasi/NN

Parsing dan Semantic Analysis

Parsing

Parsing adalah proses menganalisis struktur sintaksis dari kalimat menggunakan aturan tata bahasa (*grammar*), biasanya berupa **parse tree** atau **dependency tree**.

- **Jenis parsing:**

- Constituency parsing (struktur frase)
- Dependency parsing (relasi antar kata)

Semantic Analysis

Tujuan analisis semantik adalah memahami **makna** dari teks. Ini mencakup:

- **Word Sense Disambiguation:** Menentukan arti kata yang ambigu.
 - **Named Entity Recognition (NER):** Menemukan entitas seperti nama orang, tempat, waktu.
 - **Semantic Role Labeling (SRL):** Menandai fungsi semantik seperti siapa pelaku, objek, dan alat.
-

Word2Vec dan GloVe

Word2Vec

Word2Vec adalah metode representasi kata (word embedding) dalam bentuk vektor, yang dilatih berdasarkan konteks.

- **Model:**
 - CBOW (Continuous Bag of Words)
 - Skip-gram
- Kata-kata yang memiliki makna mirip akan memiliki vektor yang dekat di ruang dimensi.

GloVe (Global Vectors for Word Representation)

GloVe adalah pendekatan word embedding berbasis **frekuensi ko-occurrence kata** di seluruh korpus.

- Keunggulan GloVe: menggabungkan pendekatan statistik global (matriks) dan lokal (window).
 - **Contoh:**
Hubungan semantik →
$$\text{vec}(\text{"king"}) - \text{vec}(\text{"man"}) + \text{vec}(\text{"woman"}) \approx \text{vec}(\text{"queen"})$$
-

Aplikasi NLP

1. Chatbot

Chatbot adalah sistem percakapan otomatis menggunakan NLP untuk memahami dan merespons input pengguna.

- Pendekatan:
 - Rule-based chatbot
 - AI-based chatbot (menggunakan NLP dan Machine Learning)
- Contoh: ChatGPT, Google Assistant, customer support bot.

2. Analisis Sentimen

Analisis sentimen bertujuan mengetahui **emosi atau opini** pengguna dari teks.

- **Klasifikasi umum:**
 - Positif
 - Negatif
 - Netral
 - **Teknik:** Naive Bayes, SVM, LSTM, BERT, dsb.
 - **Contoh aplikasi:**
Analisis ulasan produk, tanggapan pelanggan di media sosial, opini politik.
-

Referensi

1. Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing* (3rd Ed. draft). Stanford University.
<https://web.stanford.edu/~jurafsky/slp3/>
2. Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Latihan dan Evaluasi:

1. Buat pipeline NLP sederhana.
 2. Implementasi analisis sentimen sederhana.
-

BAB 10: Computer Vision

Deskripsi: Teknik AI untuk pemrosesan dan pemahaman citra digital.

Tujuan Pembelajaran:

1. Memahami dasar pengolahan citra.
2. Melakukan segmentasi dan ekstraksi fitur.
3. Menerapkan CNN untuk pengenalan objek.

Pengolahan Citra Digital

Pengertian

Pengolahan Citra Digital (Digital Image Processing) adalah teknik untuk memanipulasi citra digital dengan tujuan meningkatkan kualitas atau mengekstrak informasi penting darinya.

Ruang Warna

- **Grayscale:** hanya memiliki satu kanal intensitas (hitam-putih)
- **RGB:** tiga kanal warna (Red, Green, Blue)
- **HSV, YCbCr:** digunakan untuk segmentasi dan pengolahan lanjutan

Tahapan Pengolahan Citra

1. **Akuisisi citra:** mendapatkan gambar dari kamera atau sensor
 2. **Prapemrosesan:** filtering, noise removal, konversi warna
 3. **Segmentasi:** memisahkan objek dari latar
 4. **Ekstraksi fitur:** mendapatkan informasi seperti tepi, bentuk, warna
 5. **Pengenalan pola atau klasifikasi**
-

Segmentasi dan Ekstraksi Fitur

Segmentasi Citra

Segmentasi adalah proses membagi citra menjadi bagian-bagian bermakna, misalnya objek dan latar belakang.

Metode segmentasi umum:

- Thresholding (Otsu)
- Region Growing
- Edge Detection (Sobel, Canny)
- Clustering (K-means)
- Deep Learning-based (U-Net, Mask R-CNN)

Ekstraksi Fitur

Ekstraksi fitur adalah proses mengidentifikasi atribut atau informasi penting dari objek dalam citra.

Jenis fitur yang sering digunakan:

- **Fitur bentuk:** kontur, aspek rasio, perimeter
 - **Fitur warna:** histogram warna, rata-rata RGB
 - **Fitur tekstur:** GLCM (Gray Level Co-occurrence Matrix), LBP (Local Binary Pattern)
 - **Fitur sudut dan tepi:** SIFT, SURF, ORB
-

Deteksi Objek (Pengantar YOLO)

Deteksi Objek

Deteksi objek adalah proses menemukan dan mengklasifikasi objek dalam gambar serta menentukan lokasi (bounding box)-nya.

YOLO (You Only Look Once) – Pengantar

YOLO adalah algoritma deteksi objek berbasis deep learning yang melakukan deteksi secara real-time.

Konsep Dasar YOLO:

- Menggunakan satu jaringan konvolusi (CNN) yang memproses seluruh gambar sekaligus.
- Gambar dibagi menjadi grid (misalnya 7x7), dan setiap sel memprediksi:
 - Bounding box (x, y, w, h)
 - Confidence score
 - Kelas objek

Kelebihan YOLO:

- Cepat dan efisien (real-time)
- Deteksi multi-objek sekaligus

Versi YOLO yang umum:

- YOLOv3: Akurasi tinggi
- YOLOv4: Cepat dan akurat
- YOLOv5: Populer dan ringan
- YOLOv8: Versi terbaru dengan peningkatan arsitektur dan hasil deteksi

Contoh aplikasi YOLO:

- Deteksi kendaraan di jalan
- Pendeteksi wajah atau helm
- Pemantauan keamanan (CCTV)

Referensi

1. Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson.
2. Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer.
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Latihan dan Evaluasi:

1. Proses edge detection.
 2. Implementasi CNN untuk klasifikasi gambar.
-

BAB 11: Robotika

Deskripsi: Integrasi AI dalam sistem robotik cerdas.

Tujuan Pembelajaran:

1. Menjelaskan struktur dasar robot dan jenis sensor.
2. Memahami navigasi dan pemetaan (SLAM).
3. Merancang kontrol gerakan dan pengambilan keputusan.

Uraian Materi:

Pengolahan Citra Digital

1.1 Pengertian

Pengolahan Citra Digital adalah proses untuk menganalisis dan memanipulasi gambar digital menggunakan algoritma komputer. Tujuannya bisa berupa peningkatan kualitas citra atau ekstraksi informasi penting dari citra.

1.2 Ruang Warna

- **Grayscale:** Intensitas hitam-putih
- **RGB:** Warna dasar (Red, Green, Blue)
- **HSV/YCbCr:** Representasi warna alternatif yang lebih efektif untuk analisis citra tertentu

1.3 Tahapan Pengolahan Citra

1. Akuisisi Citra
 2. Prapemrosesan (filtering, konversi warna, noise removal)
 3. Segmentasi
 4. Ekstraksi Fitur
 5. Klasifikasi / Analisis
-

Segmentasi dan Ekstraksi Fitur

2.1 Segmentasi

Proses memisahkan objek penting dari latar belakang.

Metode Segmentasi:

- Thresholding (Otsu)
- Region Growing
- Edge Detection (Sobel, Canny)
- Clustering (K-Means)
- Deep Learning (U-Net, Mask R-CNN)

2.2 Ekstraksi Fitur

Menentukan ciri khas dari objek yang diolah:

- **Fitur bentuk:** kontur, aspek rasio
- **Fitur warna:** histogram, rata-rata RGB
- **Fitur tekstur:** GLCM, LBP

- **Fitur tepi/sudut:** SIFT, SURF, ORB
-

Deteksi Objek (YOLO - Pengantar)

3.1 Pengertian Deteksi Objek

Deteksi objek adalah proses menemukan serta mengenali objek dalam suatu citra atau video, lengkap dengan posisi (bounding box).

3.2 YOLO (You Only Look Once)

YOLO adalah algoritma deteksi objek yang bekerja secara real-time dengan membagi citra menjadi grid dan memprediksi posisi serta kelas objek sekaligus.

Komponen Prediksi:

- Bounding box (x, y, w, h)
- Confidence score
- Kelas objek

Versi YOLO:

- YOLOv3, YOLOv4, YOLOv5, YOLOv8

Contoh Aplikasi:

- Deteksi kendaraan
 - Keamanan dan CCTV
 - Deteksi objek medis atau industri
-

Struktur Robot dan Sensor

4.1 Struktur Robot

Struktur robot mencakup bagian-bagian fisik seperti:

- **Kerangka mekanik** (rangka, aktuator, roda/lengan)
- **Aktuator** (motor DC, servo, aktuator pneumatik)
- **Penggerak** (roda diferensial, kaki, lengan artikulasi)

4.2 Sensor pada Robot

Sensor digunakan untuk mendeteksi kondisi lingkungan:

- **Sensor jarak** (Ultrasonik, Lidar)
 - **Sensor posisi dan gerakan** (IMU, enkoder)
 - **Sensor visual** (kamera RGB/Depth)
 - **Sensor sentuh dan gaya** (tactile sensor, force sensor)
-

Navigasi dan SLAM

5.1 Navigasi Robot

Navigasi adalah kemampuan robot untuk bergerak menuju tujuan secara mandiri:

- **Navigasi lokal** (menghindari rintangan)
- **Navigasi global** (menentukan jalur dari peta)

5.2 SLAM (Simultaneous Localization and Mapping)

SLAM adalah proses membangun peta lingkungan dan secara bersamaan melacak posisi robot di dalamnya.

Jenis SLAM:

- Visual SLAM (menggunakan kamera)
- Lidar SLAM (menggunakan pemindai laser)
- Graph-based SLAM

Aplikasi SLAM:

- Robot pembersih otomatis
 - Kendaraan tanpa pengemudi
 - Drone eksplorasi
-

Sistem Kontrol dan Keputusan

6.1 Sistem Kontrol

Sistem kontrol mengatur gerakan dan stabilitas robot:

- **Kontrol umpan balik** (PID Controller)

- **Kontrol gerak** (lokomosi, lengan robot)
- **Kontrol adaptif dan prediktif**

6.2 Sistem Pengambilan Keputusan

Robot mengambil keputusan berdasarkan sensor dan pemrograman:

- **Logika IF-THEN**
- **State Machine**
- **Artificial Intelligence** (machine learning, fuzzy logic)
- **Path Planning** (A*, Dijkstra, RRT)

Siciliano, B., & Khatib, O. (2016). *Springer Handbook of Robotics* (2nd ed.). Springer.

Latihan dan Evaluasi:

1. Jelaskan fungsi sensor dalam robot.
2. Simulasikan peta robot menggunakan metode SLAM.
3. Desain algoritma navigasi sederhana untuk robot bergerak.