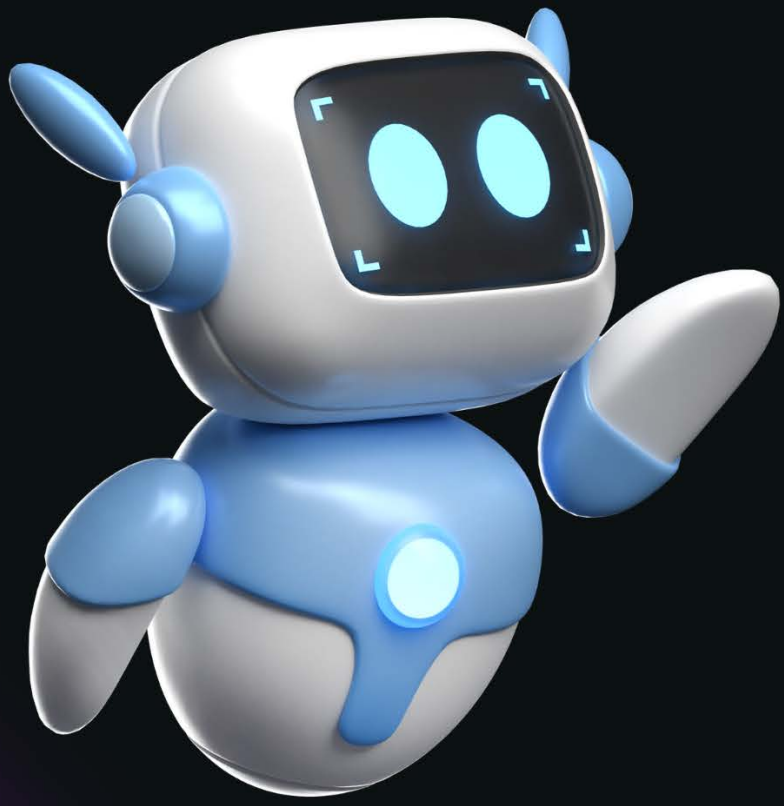
SISTEM CERDAS
Case-Based Reasoning



Assoc. Prof. Dr. Hindayati Mustafidah, S.Si., M.Kom.
Suwarsito, S.Pi., M.Si.
Emhaka Dwi Bestoro, S.Kom.

SISTEM CERDAS

BERBASIS CASE-BASED REASONING

Penulis

Assoc. Prof. Dr. Hindayati Mustafidah, S.Si., M.Kom.

Suwarsito, S.Pi., M.Si.

Emhaka Dwi Bestoro, S.Kom.

Penerbit

Zahira Media Publisher

SISTEM CERDAS

BERBASIS CASE-BASED REASONING

ISBN :

Penulis Naskah : Assoc. Prof. Dr. Hindayati
Mustafidah, S.Si., M.Kom.
Suwarsito, S.Pi., M.Si.
Emhaka Dwi Bestoro, S.Kom.

Desain Sampul : Andrianto

Jumlah Halaman : 80 Halaman

Ukuran Buku : 15,5 x 23 cm

Cetakan 1, (Bulan) 2025

Penerbit “Zahira Media Publisher”

E-mail: zahiramediapublisher@gmail.com

Anggota IKAPI : 191/JTE/2020

PEMASARAN

CV. ZT CORPORA, Jl. Ach Zein No. 97 D Pasir Kidul,
Purwokerto Barat,
Banyumas, Jawa Tengah

E-mail: cv.ztcorpora@gmail.com

KATA PENGANTAR

Segala puji dan syukur kami panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga buku ini dapat disusun dan diselesaikan dengan baik. Buku *Sistem Cerdas Berbasis Case-Based Reasoning* ini hadir sebagai upaya untuk memperkenalkan konsep, prinsip kerja, dan penerapan metode Case-Based Reasoning (CBR) dalam sistem cerdas kepada khalayak luas.

CBR merupakan pendekatan yang meniru cara manusia berpikir dan mengambil keputusan berdasarkan pengalaman sebelumnya. Pendekatan ini menawarkan fleksibilitas dan kemampuan adaptasi yang tinggi, menjadikannya relevan dalam berbagai bidang seperti kesehatan, pendidikan, lingkungan, hingga layanan berbasis teknologi. Melalui buku ini, pembaca akan diajak memahami dasar teori, proses kerja sistem CBR, serta contoh penerapannya dalam dunia nyata.

Kami menyusun buku ini dengan harapan dapat menjadi bahan bacaan yang informatif dan aplikatif, tidak hanya bagi kalangan akademisi, tetapi juga bagi siapa saja yang tertarik pada teknologi dan pemecahan masalah berbasis pengalaman. Buku ini juga diharapkan dapat mendorong lebih banyak pengembangan solusi cerdas yang dekat dengan kebutuhan nyata masyarakat.

Kami menyadari bahwa masih terdapat banyak kekurangan dalam penyusunan buku ini. Untuk itu, kami membuka diri terhadap kritik dan saran yang membangun demi penyempurnaan ke depan. Semoga buku ini dapat memberikan manfaat dan inspirasi bagi para pembaca di mana pun berada.

DAFTAR ISI

Bab 1 – Mengenal Case-Based Reasoning

Apa Itu Case-Based Reasoning?

Case-Based Reasoning (CBR) adalah pendekatan dalam kecerdasan buatan yang meniru cara manusia menyelesaikan masalah berdasarkan pengalaman masa lalu. Ketika seseorang dihadapkan pada situasi baru, secara alami mereka akan mengingat kejadian serupa yang pernah dialami sebelumnya, lalu menggunakan pengalaman tersebut sebagai acuan dalam mengambil keputusan. Konsep inilah yang menjadi dasar utama dari metode CBR.

Berbeda dari pendekatan lain dalam pemrograman atau sistem pakar yang membutuhkan aturan eksplisit, CBR lebih fleksibel karena mengandalkan kasus nyata yang telah terjadi. Setiap kasus biasanya berisi permasalahan, konteks, solusi yang pernah diambil, serta hasil atau evaluasinya. Ketika sebuah kasus baru muncul, sistem akan mencari kemiripan dengan kasus-kasus yang sudah ada untuk memberikan solusi yang paling relevan.

Inti dari CBR terletak pada proses retrieve, yaitu pengambilan satu atau beberapa kasus masa lalu yang paling mirip dengan kondisi sekarang. Setelah itu, sistem melakukan reuse, atau penggunaan solusi dari kasus terdahulu. Namun, solusi

tersebut tidak serta-merta diterapkan begitu saja. Sistem juga melalui tahap revise, yaitu penyesuaian atau koreksi terhadap solusi agar cocok dengan kondisi saat ini. Terakhir, pada tahap retain, sistem menyimpan pengalaman baru tersebut sebagai kasus tambahan di basis data.

CBR cocok digunakan dalam berbagai situasi di mana tidak ada satu jawaban yang benar, tetapi solusi bergantung pada konteks. Contohnya dalam dunia medis, diagnosis penyakit sering kali dilakukan dengan membandingkan gejala pasien saat ini dengan gejala pasien sebelumnya. Semakin mirip kondisinya, semakin besar kemungkinan solusi atau penanganan yang serupa bisa diterapkan.

Selain di dunia kesehatan, CBR juga banyak digunakan dalam bidang pertanian, manajemen bisnis, sistem rekomendasi, hingga teknologi informasi. Keunggulannya terletak pada kemampuan sistem untuk “belajar” dari waktu ke waktu. Setiap kali sistem menghadapi kasus baru dan berhasil memberikan solusi yang tepat, maka pengalaman tersebut akan memperkaya basis kasus dan meningkatkan kualitas keputusan di masa mendatang.

Salah satu kelebihan besar dari pendekatan ini adalah kemudahan dalam pengembangan sistem. Karena tidak membutuhkan pengetahuan dalam bentuk aturan kompleks,

pengembang cukup mengumpulkan data kasus nyata, lalu menyusun struktur data yang tepat untuk menyimpan dan mencarinya. Hal ini membuat CBR menjadi metode yang efisien dan praktis dalam implementasi dunia nyata.

Namun demikian, keberhasilan sistem CBR sangat tergantung pada kualitas dan keragaman basis kasus yang dimilikinya. Jika sistem hanya memiliki sedikit contoh atau kasus yang kurang relevan, maka kualitas solusi yang diberikan juga akan terbatas. Oleh karena itu, proses pengumpulan data dan validasi kasus menjadi aspek penting dalam membangun sistem CBR yang andal.

Dalam praktiknya, CBR sering dipadukan dengan metode lain seperti machine learning atau decision tree untuk meningkatkan ketepatan dan efisiensinya. Misalnya, algoritma pembelajaran mesin dapat digunakan untuk mempercepat proses pencarian kasus atau membantu menyaring kasus-kasus yang paling relevan. Dengan cara ini, sistem menjadi lebih adaptif dan responsif terhadap perubahan kondisi.

Dengan kemampuannya yang menyerupai cara berpikir manusia, CBR menjadi pilihan populer dalam pengembangan sistem cerdas berbasis pengalaman. Bukan hanya menyimpan data, tetapi sistem ini mengingat, memahami, dan belajar dari

pengalaman tersebut. Hal ini menjadikan CBR sebagai pendekatan yang sangat manusiawi dalam dunia komputasi.

Melalui bab-bab selanjutnya, pembaca akan diajak lebih dalam mengenal bagaimana CBR dibangun, dijalankan, dan diterapkan dalam aplikasi dunia nyata, khususnya dalam pengembangan sistem berbasis web. Dengan pemahaman yang baik terhadap prinsip-prinsip CBR, diharapkan pembaca dapat mulai merancang solusi cerdas yang berbasis pengalaman secara mandiri.

Sejarah dan Perkembangan Case-Based Reasoning

Case-Based Reasoning (CBR) pertama kali dikenalkan secara formal sebagai pendekatan dalam kecerdasan buatan pada awal tahun 1980-an. Konsep dasarnya berasal dari pengamatan terhadap bagaimana manusia berpikir dan mengambil keputusan—yakni dengan mengingat pengalaman sebelumnya. Pendekatan ini menjadi populer karena menawarkan cara baru untuk membuat sistem komputer berpikir lebih seperti manusia.

Salah satu pionir dalam pengembangan CBR adalah Roger Schank, seorang peneliti di Yale University. Ia memperkenalkan ide bahwa pengetahuan manusia tidak selalu tersimpan dalam bentuk aturan, melainkan lebih sering dalam

bentuk cerita atau pengalaman. Dari gagasan inilah lahir program CBR awal seperti CYRUS dan CHEF, yang digunakan untuk menjawab pertanyaan dan menyusun resep makanan berdasarkan pengalaman masa lalu.

CBR kemudian berkembang pesat pada akhir 1980-an dan awal 1990-an seiring meningkatnya kebutuhan akan sistem pakar dan aplikasi berbasis pengetahuan. Saat itu, banyak pendekatan kecerdasan buatan masih bergantung pada aturan if-then yang kaku. CBR menjadi alternatif yang lebih fleksibel karena tidak memerlukan formulasi aturan yang rumit. Sistem cukup belajar dari contoh nyata.

Pada dekade 1990-an, berbagai konferensi dan jurnal ilmiah mulai menaruh perhatian serius pada CBR. Salah satu forum internasional yang mewadahi perkembangan ini adalah International Conference on Case-Based Reasoning (ICCBR), yang hingga kini masih rutin diselenggarakan. Perkembangan ini turut mendorong banyak penelitian dan aplikasi CBR di berbagai bidang, dari kesehatan, hukum, manufaktur, hingga pendidikan.

Seiring dengan kemajuan teknologi informasi, CBR tidak hanya dikembangkan sebagai sistem desktop, tapi juga mulai diintegrasikan ke dalam aplikasi berbasis web dan perangkat mobile. Dengan kemudahan akses dan penyimpanan data

yang besar di cloud, basis kasus dalam sistem CBR bisa dikembangkan lebih luas dan dikelola secara kolaboratif oleh banyak pengguna sekaligus.

Perkembangan penting berikutnya terjadi ketika CBR mulai dikombinasikan dengan pendekatan lain seperti machine learning dan data mining. Misalnya, dalam sistem rekomendasi, CBR digunakan untuk mengingat pilihan pengguna sebelumnya, sementara machine learning membantu memprediksi tren berdasarkan data yang besar. Kombinasi ini memperkuat kemampuan sistem untuk memberikan solusi yang personal dan relevan.

Tak hanya dalam ranah teknis, CBR juga berkembang dari sisi konseptual. Para peneliti mulai memperluas pengertian "kasus" menjadi tidak hanya berupa masalah dan solusi, tetapi juga menyertakan konteks sosial, tujuan, dan proses pengambilan keputusan. Hal ini menjadikan CBR semakin relevan untuk diterapkan di lingkungan yang kompleks dan dinamis.

Saat ini, CBR digunakan di berbagai sektor praktis. Di bidang medis, misalnya, sistem CBR digunakan untuk membantu diagnosis dengan membandingkan gejala pasien saat ini dengan kasus pasien sebelumnya. Di bidang hukum, CBR dapat membantu pengacara menemukan preseden kasus

serupa. Bahkan dalam pendidikan, sistem CBR dapat menyesuaikan materi ajar berdasarkan pengalaman belajar siswa sebelumnya.

Pengembangan CBR juga semakin terbantu dengan ketersediaan berbagai perangkat lunak dan pustaka pemrograman. Banyak bahasa pemrograman seperti Python, Java, dan PHP telah menyediakan modul atau framework yang memudahkan implementasi algoritma CBR. Ini menjadikan CBR tidak hanya relevan di dunia akademik, tetapi juga praktis digunakan oleh pengembang perangkat lunak.

Dengan sejarah yang telah berjalan lebih dari empat dekade, CBR kini bukan sekadar pendekatan alternatif dalam kecerdasan buatan, tetapi telah menjadi fondasi penting dalam pengembangan sistem berbasis pengalaman. Kemampuannya untuk terus belajar dari waktu ke waktu menjadikan CBR sebagai metode yang sangat adaptif, terutama di era digital yang bergerak cepat dan penuh perubahan.

CBR dalam Konteks Sistem Cerdas dan AI

Dalam dunia kecerdasan buatan (AI), terdapat berbagai pendekatan untuk menyelesaikan masalah dan meniru kecerdasan manusia. Beberapa yang populer adalah rule-based system, neural networks, decision trees, hingga learning

algorithms. Case-Based Reasoning (CBR) hadir sebagai pendekatan yang unik karena tidak hanya berfokus pada pembelajaran melalui aturan, tetapi lebih menekankan pada kemampuan untuk mengingat dan mengadaptasi pengalaman masa lalu.

Sistem cerdas, secara umum, dirancang untuk meniru cara manusia berpikir, mengambil keputusan, dan memecahkan masalah. Dalam konteks ini, CBR sangat cocok karena pendekatan ini memang dirancang berdasarkan perilaku alami manusia dalam berpikir. Saat seseorang dihadapkan pada situasi baru, ia cenderung mencari kemiripan dengan apa yang pernah dialami sebelumnya. Mekanisme berpikir ini menjadi inti dari sistem CBR.

Berbeda dengan model pembelajaran mesin tradisional yang mengharuskan pelatihan (training) pada data besar dalam sekali waktu, CBR bekerja secara episodik. Artinya, sistem tidak harus mempelajari semua data sekaligus, tetapi dapat terus berkembang seiring dengan bertambahnya kasus baru. Hal ini membuat CBR sangat fleksibel untuk diterapkan pada sistem yang berkembang secara dinamis.

CBR juga sangat berguna dalam situasi di mana solusi tidak selalu pasti atau kebenarannya relatif. Dalam bidang medis, misalnya, diagnosis penyakit tidak selalu hitam-putih.

Beberapa gejala bisa mirip untuk beberapa penyakit, dan dokter sering mengandalkan pengalaman serupa yang pernah mereka tangani. Sistem CBR mampu meniru pendekatan ini, membuatnya sangat relevan untuk sistem cerdas dalam domain tak terstruktur.

Integrasi CBR dengan sistem cerdas lainnya pun terus berkembang. Misalnya, dalam agent-based systems, CBR dapat digunakan untuk membekali agen cerdas dengan kemampuan pengambilan keputusan berdasarkan pengalaman mereka sendiri. Bahkan dalam sistem robotik, CBR bisa membantu robot "mengingat" strategi yang pernah berhasil saat menghadapi situasi tertentu di masa lalu.

Dari sisi teknis, CBR sangat bisa dikombinasikan dengan machine learning. Misalnya, dalam proses retrieve, algoritma klasifikasi atau clustering dapat digunakan untuk mempercepat pencarian kasus yang paling mirip. Sementara dalam proses revise, reinforcement learning dapat digunakan untuk menyempurnakan hasil solusi berdasarkan feedback dari lingkungan.

Kemampuan CBR untuk menyimpan dan memanfaatkan pengalaman menjadikannya bagian penting dalam membangun sistem yang adaptif dan terus belajar. CBR tidak hanya menyelesaikan masalah, tetapi juga membentuk ingatan

kolektif digital yang dapat digunakan untuk pembelajaran jangka panjang. Hal ini sangat sesuai dengan prinsip dari AI modern yang menekankan pada kemampuan sistem untuk berkembang seiring waktu.

Dalam konteks big data, CBR tetap memiliki tempatnya sendiri. Meskipun tidak mengandalkan data dalam jumlah sangat besar seperti deep learning, CBR justru unggul dalam situasi di mana data terbatas namun berkualitas. Bahkan, di tengah tren pemrosesan data skala besar, banyak perusahaan dan organisasi lebih memilih CBR untuk sistem yang perlu menjelaskan alasan keputusan secara transparan dan berbasis pengalaman.

Selain itu, CBR mempermudah proses debugging dan audit sistem. Karena setiap keputusan sistem didasarkan pada kasus nyata yang tersimpan, pengguna dapat menelusuri asal-usul solusi yang diberikan. Hal ini memberi tingkat kepercayaan yang lebih tinggi dibandingkan model AI yang cenderung bersifat “black box”.

Dengan karakteristiknya yang mendekati cara manusia berpikir dan fleksibilitasnya dalam beradaptasi dengan kondisi nyata, CBR adalah salah satu pilar dalam pengembangan sistem cerdas modern. Baik digunakan sendiri maupun dikombinasikan dengan teknologi AI lainnya, CBR

tetap relevan dan bahkan semakin penting di era digital yang menuntut sistem mampu belajar, mengingat, dan menyesuaikan diri dengan cepat.

Empat Langkah Utama: Retrieve, Reuse, Revise, Retain

Inti dari metode Case-Based Reasoning terletak pada empat langkah utama yang secara berurutan membentuk siklus pemecahan masalah: *retrieve*, *reuse*, *revise*, dan *retain*. Keempat langkah ini merepresentasikan bagaimana sistem cerdas bekerja layaknya manusia saat menghadapi situasi baru dengan memanfaatkan pengalaman sebelumnya. Siklus ini dikenal sebagai *CBR cycle* dan menjadi fondasi dari setiap sistem CBR yang baik.

Langkah pertama adalah **retrieve**, yaitu proses mencari dan mengambil satu atau beberapa kasus dalam basis data yang memiliki kemiripan tertinggi dengan kasus baru yang sedang dihadapi. Sistem akan membandingkan parameter-parameter kunci dari kasus baru dengan kasus-kasus lama menggunakan metode perhitungan kemiripan (*similarity measure*). Semakin tinggi nilai kemiripan, semakin besar kemungkinan solusi dari kasus tersebut bisa digunakan kembali.

Setelah kasus yang paling mirip ditemukan, sistem masuk ke tahap **reuse**, yaitu menggunakan solusi dari kasus yang

diambil untuk menyelesaikan kasus baru. Dalam praktiknya, reuse bisa dilakukan dengan langsung menerapkan solusi yang sama, atau dengan melakukan sedikit penyesuaian tergantung pada kompleksitas masalah. Tujuan utama dari tahap ini adalah menyederhanakan proses pencarian solusi tanpa perlu memulai dari nol.

Langkah ketiga adalah **revise**, yaitu mengevaluasi apakah solusi yang diambil benar-benar cocok untuk kasus baru. Dalam dunia nyata, tidak semua solusi dari masa lalu bisa digunakan mentah-mentah. Kadang perlu modifikasi kecil, validasi dari pakar, atau bahkan uji coba langsung. Tahap revise inilah yang memberi fleksibilitas dan kecerdasan pada sistem, karena ia belajar untuk menyesuaikan solusi dengan situasi baru.

Jika solusi yang direvisi ternyata berhasil, maka masuk ke tahap terakhir: **retain**. Pada tahap ini, kasus baru—beserta solusi yang telah diperbaiki—disimpan dalam basis kasus sebagai pengetahuan baru. Dengan demikian, sistem tidak hanya menyelesaikan masalah, tetapi juga memperkaya memorinya. Semakin sering digunakan, semakin luas pula pengalaman yang dimiliki oleh sistem.

Empat langkah ini bekerja secara berulang. Kasus-kasus baru yang masuk akan terus memperkuat kemampuan sistem dalam

menghadapi variasi masalah di masa depan. Dengan mekanisme retain yang baik, sistem menjadi adaptif—tidak hanya meniru, tapi juga berkembang dan belajar dari interaksi.

Yang menarik, proses ini sangat paralel dengan cara manusia belajar. Misalnya, ketika seseorang belajar memasak makanan baru, mereka cenderung mengingat resep yang pernah digunakan sebelumnya (retrieve), mencoba menggunakannya kembali (reuse), mencicipi hasilnya dan menyesuaikan bumbu jika perlu (revise), lalu menyimpan hasil perbaikan tersebut dalam ingatan (retain). CBR menerjemahkan logika alami ini ke dalam bentuk algoritma.

Dalam pengembangan perangkat lunak, setiap langkah dalam siklus CBR bisa dirancang secara modular. Retrieve dapat dioptimalkan dengan algoritma pencarian cepat, reuse bisa diatur dengan template solusi, revise dikembangkan bersama pakar atau melalui feedback pengguna, dan retain diprogram agar menyimpan kasus penting secara otomatis. Dengan desain seperti ini, CBR menjadi kerangka yang fleksibel dan kuat.

Namun, penerapan keempat langkah tersebut juga menimbulkan tantangan tersendiri. Misalnya, bagaimana mengatur agar basis kasus tidak terlalu penuh atau berisi data yang redundan? Atau bagaimana memastikan bahwa solusi

yang direvisi benar-benar lebih baik dari sebelumnya? Pertanyaan-pertanyaan ini membuka ruang untuk pengembangan lanjutan dalam implementasi CBR.

Secara keseluruhan, empat langkah ini bukan hanya prosedur teknis, tetapi juga filosofi kerja CBR yang menjadikannya berbeda dari pendekatan lain. CBR bukan hanya tentang memproses data, tapi tentang *belajar dari pengalaman* dan *bertumbuh seiring waktu*. Itulah yang membuatnya istimewa dalam dunia sistem cerdas.

Bab 2 – Menyusun Solusi dari Pengalaman

Contoh Aplikasi CBR dalam Dunia Nyata

Case-Based Reasoning bukan sekadar teori akademik, melainkan sudah banyak diterapkan dalam berbagai sektor kehidupan nyata. Salah satu bidang yang paling awal dan paling sukses mengadopsi CBR adalah dunia **medis**. Di rumah sakit dan klinik, sistem berbasis CBR digunakan untuk membantu diagnosis penyakit dengan membandingkan gejala pasien saat ini dengan kasus-kasus sebelumnya. Sistem ini mampu merekomendasikan kemungkinan penyakit dan tindakan medis berdasarkan riwayat gejala serupa, yang sering kali mempercepat proses diagnosis dan meningkatkan akurasi.

Dalam **bidang pendidikan**, CBR dimanfaatkan dalam sistem pembelajaran adaptif. Sistem ini menyimpan data riwayat belajar setiap siswa, termasuk kesalahan yang sering dilakukan dan materi yang belum dikuasai. Ketika siswa menghadapi soal baru atau mengalami kesulitan, sistem akan merekomendasikan materi tambahan atau metode belajar yang telah efektif untuk siswa lain dengan profil yang serupa. Ini memungkinkan pembelajaran yang lebih personal dan efisien.

CBR juga sangat berguna dalam **industri manufaktur dan pemeliharaan mesin**. Sistem diagnosis kerusakan mesin berbasis CBR mampu mendeteksi permasalahan mesin

dengan membandingkan gejala saat ini (seperti suara, getaran, atau performa menurun) dengan kasus kerusakan sebelumnya. Dengan solusi yang telah terbukti efektif di masa lalu, teknisi bisa lebih cepat menentukan langkah perbaikan tanpa harus menelusuri masalah dari awal.

Di dunia **pertanian dan peternakan**, CBR digunakan dalam sistem pakar untuk mengidentifikasi serangan hama atau penyakit tanaman berdasarkan gejala visual. Misalnya, petani dapat memasukkan deskripsi daun yang menguning, bercak hitam, atau jenis serangga yang muncul, lalu sistem akan mencocokkan dengan basis kasus yang ada. Sistem ini membantu petani mendapatkan saran pengendalian tanpa harus menunggu kunjungan ahli.

Bidang **e-commerce** dan **rekomendasi produk** juga memanfaatkan pendekatan CBR. Situs belanja online sering kali menyimpan riwayat perilaku pengguna—produk yang diklik, dibeli, atau diabaikan. Berdasarkan pola tersebut, sistem akan merekomendasikan produk lain yang cocok berdasarkan kasus pembeli sebelumnya yang memiliki selera serupa. Ini menjadikan pengalaman berbelanja lebih personal dan meningkatkan konversi penjualan.

CBR juga berperan dalam **layanan pelanggan dan helpdesk**. Sistem dapat menyimpan pertanyaan dan solusi yang telah

diberikan kepada pelanggan sebelumnya. Ketika ada pelanggan baru dengan masalah yang mirip, sistem langsung merekomendasikan solusi yang relevan, menghemat waktu agen layanan dan mempercepat penanganan masalah.

Dalam **sistem hukum**, CBR bisa digunakan untuk mencari preseden atau keputusan hukum masa lalu yang mirip dengan kasus baru. Hal ini membantu pengacara atau hakim memahami bagaimana kasus serupa diselesaikan dan memberikan dasar argumen yang lebih kuat. Di negara-negara dengan sistem hukum berbasis preseden, aplikasi CBR seperti ini sangat bermanfaat.

Bahkan dalam **game dan robotika**, CBR digunakan untuk memberikan perilaku yang adaptif. Karakter dalam game, misalnya, dapat mengingat pola permainan pengguna dan menyesuaikan strategi berdasarkan pengalaman bermain sebelumnya. Robot juga dapat mengingat cara-cara yang paling efisien dalam menyelesaikan tugas tertentu dan menerapkannya kembali saat menghadapi situasi serupa.

Yang menarik, CBR juga telah diterapkan dalam **perencanaan perkotaan** dan **manajemen bencana**. Misalnya, pemerintah dapat menggunakan CBR untuk mengambil keputusan cepat dalam penanggulangan banjir dengan melihat data dari kejadian sebelumnya, seperti pola

hujan, ketinggian air, dan respon yang paling efektif. Pendekatan ini terbukti mempercepat tindakan di lapangan.

Semua contoh di atas menunjukkan bahwa CBR bukanlah metode yang terbatas pada satu disiplin ilmu. Justru sebaliknya, ia bersifat lintas bidang dan sangat fleksibel. Di mana pun ada pengalaman yang bisa disimpan dan dipelajari, di situ CBR bisa diimplementasikan. Kuncinya adalah bagaimana kita menyusun basis kasus yang relevan, menjaga kualitas data, dan memastikan sistem terus belajar dari waktu ke waktu.

Merancang Struktur Data dan Parameter

Agar sistem Case-Based Reasoning (CBR) dapat bekerja dengan optimal, langkah awal yang sangat penting adalah merancang struktur data dan parameter yang tepat. Struktur data ini menjadi kerangka tempat sistem menyimpan, mengakses, dan membandingkan kasus-kasus yang ada. Tanpa struktur yang rapi dan logis, sistem akan kesulitan melakukan proses pencocokan (*retrieve*) yang akurat.

Sebuah kasus dalam sistem CBR biasanya terdiri atas dua bagian utama: *deskripsi masalah* dan *solusi yang diterapkan*. Deskripsi masalah berisi nilai-nilai parameter yang menggambarkan kondisi atau situasi tertentu. Sedangkan

bagian solusi mencatat tindakan atau rekomendasi yang telah diberikan serta hasilnya. Dalam dunia nyata, struktur ini bisa dikembangkan menjadi lebih kompleks tergantung kebutuhan sistem.

Pemilihan **parameter** menjadi hal yang sangat krusial dalam membangun basis kasus. Parameter adalah elemen-elemen yang menjadi titik pembanding antar kasus. Contohnya, jika sistem digunakan untuk diagnosis penyakit, maka parameter bisa berupa gejala seperti demam, batuk, tekanan darah, atau hasil tes laboratorium. Semakin relevan parameter yang dipilih, semakin tinggi akurasi solusi yang dihasilkan.

Setiap parameter idealnya memiliki skala atau nilai yang bisa diukur secara konsisten, baik itu dalam bentuk angka (kuantitatif) maupun kategori (kualitatif). Misalnya, suhu tubuh bisa dinyatakan dalam derajat Celsius, sedangkan gejala batuk bisa dikategorikan sebagai “ringan”, “sedang”, atau “berat”. Parameter-parameter ini nantinya akan diolah dalam perhitungan kemiripan saat proses retrieve.

Dalam merancang struktur data, penting juga untuk menentukan **bobot** pada masing-masing parameter. Tidak semua parameter memiliki pengaruh yang sama terhadap pengambilan keputusan. Misalnya, dalam sistem identifikasi ikan air tawar, pH air mungkin lebih penting dibandingkan

suhu udara. Memberikan bobot yang tepat akan membantu sistem memprioritaskan faktor-faktor yang paling menentukan.

Selain itu, struktur data juga harus fleksibel agar mudah diperluas seiring waktu. Sistem yang baik seharusnya mampu menambahkan parameter baru tanpa harus merombak keseluruhan basis data. Dengan cara ini, sistem dapat terus diperbarui dan disesuaikan dengan kebutuhan pengguna atau perkembangan ilmu pengetahuan yang relevan.

Dalam implementasi teknis, struktur data kasus umumnya disimpan dalam bentuk tabel atau objek dalam basis data. Untuk sistem berbasis web, struktur ini bisa diatur dalam file SQL atau dalam bentuk JSON yang ringan dan mudah dibaca. Yang penting adalah sistem harus mampu membaca, mengolah, dan menampilkan data ini dengan cepat saat proses pencocokan kasus terjadi.

Tak kalah pentingnya adalah mempertimbangkan **kelengkapan dan kualitas data** dalam setiap kasus. Data yang tidak lengkap, ambigu, atau salah input akan berdampak langsung pada hasil pencocokan. Oleh karena itu, validasi saat pengisian kasus dan penggunaan form input yang terstruktur sangat membantu menjaga kualitas data.

Dalam sistem yang lebih kompleks, struktur data kasus bahkan bisa mencakup riwayat perubahan, tanggal pengujian, validasi pakar, hingga feedback pengguna. Informasi tambahan ini sangat berguna untuk proses revise dan retain, di mana sistem memperbaiki dan menyimpan pengalaman baru secara lebih kontekstual.

Dengan perancangan struktur data dan parameter yang tepat, sistem CBR tidak hanya menjadi cerdas, tetapi juga dapat diandalkan dan tumbuh seiring waktu. Itulah yang membedakan sistem CBR dari model-model lain—kemampuannya untuk berkembang dari pengalaman nyata dan menyajikan solusi yang kontekstual dan tepat guna.

Strategi Menyusun Basis Kasus yang Efektif

Basis kasus adalah fondasi dari sistem Case-Based Reasoning. Semakin lengkap, relevan, dan terorganisasi basis kasus yang dimiliki, semakin tinggi pula kualitas solusi yang dihasilkan oleh sistem. Karena itu, menyusun basis kasus tidak bisa dilakukan sembarangan. Diperlukan strategi agar kasus-kasus yang disimpan benar-benar mewakili beragam kondisi nyata dan dapat diandalkan saat sistem diminta mengambil keputusan.

Langkah pertama dalam menyusun basis kasus yang baik adalah mengidentifikasi kebutuhan pengguna dan tujuan sistem. Apakah sistem digunakan untuk diagnosis, rekomendasi, atau klasifikasi? Setiap jenis aplikasi akan menentukan parameter apa yang dibutuhkan, serta bentuk dan format data seperti apa yang paling cocok. Basis kasus untuk diagnosis penyakit tentu berbeda dengan basis kasus untuk rekomendasi produk atau identifikasi kondisi lingkungan.

Selanjutnya, penting untuk mengumpulkan data dari berbagai sumber yang kredibel. Data bisa berasal dari wawancara dengan ahli, laporan teknis, data lapangan, maupun studi literatur. Pada fase awal, data mungkin sedikit, tapi yang terpenting adalah memastikan bahwa setiap kasus benar-benar valid. Sistem CBR sangat sensitif terhadap kualitas data—kasus yang salah atau tidak lengkap bisa menyebabkan rekomendasi yang keliru.

Untuk menjaga keandalan, setiap kasus sebaiknya melalui proses verifikasi atau validasi sebelum dimasukkan ke dalam sistem. Hal ini bisa dilakukan oleh pakar di bidang terkait atau melalui uji coba terbatas. Validasi ini sangat penting terutama jika sistem akan digunakan oleh publik atau dalam situasi kritis, seperti bidang kesehatan atau lingkungan.

Salah satu strategi yang juga penting adalah menghindari redundansi dalam penyimpanan kasus. Jika ada dua kasus yang sangat mirip dan solusinya sama, cukup simpan salah satunya. Redundansi akan memperlambat proses pencarian dan meningkatkan beban sistem tanpa memberikan manfaat tambahan. Untuk itu, perlu dilakukan proses penyaringan dan pengelompokan kasus secara berkala.

Agar basis kasus selalu up to date, sistem sebaiknya dilengkapi dengan fitur pembelajaran berkelanjutan, yaitu kemampuan untuk menyimpan kasus baru setelah melalui proses revise. Kasus baru ini akan menjadi tambahan pengetahuan yang memperkuat sistem. Namun demikian, tidak semua kasus harus langsung disimpan—perlu ada kriteria tertentu, seperti efektivitas solusi atau konfirmasi dari pengguna.

Strategi berikutnya adalah mengelompokkan kasus berdasarkan kategori atau klasifikasi tertentu. Misalnya, dalam sistem pemilihan jenis ikan air tawar, kasus bisa dikelompokkan berdasarkan kondisi wilayah (dataran tinggi atau rendah) atau jenis sumber air. Pengelompokan ini memudahkan sistem dalam mempersempit pencarian dan meningkatkan efisiensi retrieve.

Dalam pengembangan jangka panjang, sebaiknya basis kasus dilengkapi dengan metadata, seperti tanggal input, lokasi kasus, atau siapa yang memasukkan data. Metadata ini membantu dalam pelacakan, audit, dan analisis performa sistem. Kita bisa mengetahui misalnya, kasus mana yang sering muncul atau solusi mana yang paling banyak digunakan.

Untuk sistem berskala besar, sangat disarankan menggunakan database manajemen yang kokoh dan scalable, seperti MySQL, PostgreSQL, atau MongoDB. Ini akan memudahkan pengelolaan ribuan bahkan jutaan data kasus, serta mendukung pencarian cepat dan integrasi dengan antarmuka pengguna berbasis web.

Dengan strategi-strategi tersebut, sistem CBR akan memiliki basis kasus yang kaya, relevan, dan terus berkembang. Inilah yang membedakan sistem yang asal-asalan dengan sistem cerdas yang benar-benar adaptif dan bermanfaat. Basis kasus bukan hanya kumpulan data, melainkan inti dari "ingatan" sistem yang menjadi sumber pengetahuan bagi masa depan.

Menentukan Kemiripan dan Algoritma Similarity

Salah satu keunggulan utama dari sistem Case-Based Reasoning (CBR) adalah kemampuannya membandingkan kasus baru dengan kasus-kasus sebelumnya secara otomatis. Proses ini disebut *similarity matching* atau pencocokan kemiripan, dan merupakan jantung dari langkah pertama dalam siklus CBR: *retrieve*. Untuk membuat sistem benar-benar "cerdas" dalam memilih kasus yang paling relevan, kita perlu menentukan cara yang tepat dalam mengukur kemiripan antar kasus.

Kemiripan tidak selalu berarti "identik". Dalam CBR, sistem cukup mencari kasus yang *mirip* atau memiliki sejumlah nilai parameter yang sama atau mendekati. Misalnya, jika suhu air pada kasus lama adalah 28°C dan kasus baru 27°C, maka sistem akan mengenalinya sebagai cukup dekat. Hal yang sama berlaku untuk parameter lainnya seperti pH, DO (Dissolved Oxygen), atau suhu udara.

Untuk menghitung kemiripan, digunakan rumus-rumus atau algoritma yang disebut *similarity functions*. Rumus yang paling umum digunakan adalah **euclidean distance**, yang menghitung jarak antara dua titik dalam ruang multidimensi. Setiap parameter dianggap sebagai satu dimensi, dan semakin kecil jaraknya, semakin mirip kasus tersebut. Namun, rumus

ini paling cocok digunakan pada data numerik yang berskala seragam.

Jika data bersifat campuran—misalnya gabungan antara angka dan kategori—maka pendekatan lain seperti **weighted similarity** atau **nearest neighbor** sering digunakan. Dalam pendekatan ini, setiap parameter diberi bobot, lalu tingkat kesamaan dihitung berdasarkan seberapa cocok nilai antarparameter pada dua kasus. Hasil akhirnya berupa skor kemiripan (biasanya dalam persentase), yang menentukan urutan kasus dari yang paling relevan ke yang paling tidak relevan.

Penentuan bobot sangat penting agar sistem tahu parameter mana yang lebih berpengaruh dalam pengambilan keputusan. Misalnya, dalam sistem pemilihan jenis ikan air tawar, parameter amonia dan DO mungkin lebih krusial daripada suhu udara. Oleh karena itu, sistem dapat memberikan bobot lebih tinggi pada parameter tersebut, sehingga meskipun suhu udara tidak persis sama, sistem tetap bisa mengenali bahwa dua kasus tersebut sangat mirip.

Beberapa sistem CBR yang lebih canggih bahkan memungkinkan penggunaan algoritma pembelajaran mesin untuk secara otomatis menyesuaikan bobot berdasarkan efektivitas solusi dari waktu ke waktu. Ini membuat sistem

tidak hanya mengandalkan rumus statis, tapi juga belajar dari pengalaman dan umpan balik (*feedback*) dari pengguna.

Selain perhitungan jarak dan bobot, penting juga untuk menangani data yang kosong atau tidak lengkap. Sistem CBR yang baik biasanya memiliki mekanisme *normalisasi* nilai dan toleransi terhadap data hilang. Dengan begitu, sistem tetap bisa menghitung kemiripan meskipun ada satu atau dua parameter yang tidak diisi oleh pengguna.

Setelah skor kemiripan dari semua kasus dihitung, sistem akan mengurutkan kasus dari yang paling mirip ke yang paling berbeda. Biasanya, hanya 1 sampai 5 kasus teratas yang ditampilkan atau digunakan untuk proses selanjutnya. Kasus-kasus inilah yang akan menjadi dasar rekomendasi atau solusi yang diberikan oleh sistem.

Dalam implementasi teknis, perhitungan kemiripan bisa dilakukan menggunakan bahasa pemrograman seperti PHP, Python, atau JavaScript dengan rumus matematika sederhana. Meski terdengar teknis, logika di baliknya sangat manusiawi: kita membandingkan kondisi sekarang dengan pengalaman sebelumnya, lalu menilai mana yang paling relevan.

Dengan perhitungan kemiripan yang tepat, sistem CBR akan mampu mengambil keputusan yang kontekstual dan mendekati solusi ideal. Di sinilah letak kekuatan sistem:

bukan sekadar menyimpan data, tapi benar-benar memahami pola di balik pengalaman dan menerapkannya dalam situasi baru.

Bab 3 – Membangun Sistem Cerdas Berbasis Web

Tahapan Merancang Aplikasi Web Cerdas

Setelah memahami prinsip kerja Case-Based Reasoning (CBR), langkah berikutnya adalah mengubah konsep tersebut menjadi aplikasi nyata yang bisa digunakan oleh pengguna. Salah satu pendekatan paling praktis dan fleksibel untuk mewujudkannya adalah dengan membangun aplikasi berbasis web. Aplikasi web memungkinkan pengguna mengakses sistem dari mana saja, kapan saja, tanpa harus menginstal perangkat lunak tambahan.

Merancang aplikasi web cerdas berbasis CBR memerlukan beberapa tahapan penting. Tahapan ini harus dirancang secara sistematis agar sistem yang dibangun benar-benar dapat menjalankan keempat siklus CBR dengan baik: retrieve, reuse, revise, dan retain. Meski tiap proyek memiliki karakteristik yang berbeda, namun secara umum proses pengembangan aplikasi CBR berbasis web terdiri dari tujuh tahap: analisis kebutuhan, perancangan sistem, desain antarmuka, perancangan basis data, implementasi logika CBR, pengujian, dan pemeliharaan.

Tahap pertama adalah **analisis kebutuhan**, yaitu memahami secara menyeluruh apa tujuan sistem, siapa penggunanya, dan masalah apa yang ingin diselesaikan. Pada tahap ini, pengembang biasanya berdiskusi dengan pemilik ide atau

calon pengguna untuk mengidentifikasi parameter penting, jenis solusi yang dibutuhkan, dan alur penggunaan sistem secara umum.

Setelah kebutuhan dipahami, masuk ke tahap **perancangan sistem**. Di sini, pengembang menentukan bagaimana struktur sistem akan dibangun, termasuk modul-modul yang diperlukan. Misalnya, sistem akan memiliki modul input kasus baru, pencarian kasus serupa, modul revisi solusi, serta penyimpanan kasus baru. Perancangan ini juga mencakup pemilihan teknologi yang akan digunakan, seperti PHP, Python, JavaScript, atau framework tertentu.

Berikutnya adalah **desain antarmuka pengguna (user interface)**. Meskipun sistem di balik layar kompleks, antarmuka harus tetap sederhana dan mudah digunakan. Desain yang baik akan memudahkan pengguna dalam mengisi parameter, membaca rekomendasi, memberikan masukan terhadap solusi, hingga menambahkan kasus baru. Desain antarmuka yang buruk justru bisa membuat pengguna enggan menggunakan sistem.

Tahap selanjutnya adalah **perancangan basis data**, yang menjadi tempat menyimpan semua kasus yang ada. Struktur database harus dirancang agar bisa menyimpan deskripsi masalah, parameter, solusi, hasil evaluasi, dan metadata

lainnya. Basis data ini juga harus dioptimalkan untuk pencarian cepat, karena performa retrieve sangat bergantung pada struktur penyimpanan yang efisien.

Kemudian masuk ke tahap inti: **implementasi logika CBR**. Di tahap ini, pengembang membuat fungsi atau algoritma untuk menghitung kemiripan antar kasus, menampilkan hasil pencocokan, mengatur bobot parameter, serta menambahkan kasus baru ke database. Logika ini biasanya diintegrasikan dalam backend aplikasi dan dihubungkan dengan antarmuka serta database.

Setelah implementasi selesai, sistem harus melalui tahap **pengujian (testing)**. Pengujian dilakukan untuk memastikan sistem berjalan sesuai alur yang dirancang dan memberikan hasil yang akurat. Pengujian dilakukan baik dari sisi fungsionalitas, seperti apakah input dan output sesuai harapan, maupun dari sisi performa, seperti seberapa cepat sistem melakukan retrieve dari ribuan kasus.

Tahap terakhir adalah **pemeliharaan dan pengembangan berkelanjutan**. Sistem harus mampu berkembang seiring waktu, baik dari sisi penambahan kasus, perbaikan logika, maupun penyempurnaan antarmuka. Pemeliharaan yang rutin juga mencegah sistem dari kesalahan data, bug, atau ketidaksesuaian dengan kondisi baru.

Membangun aplikasi web cerdas bukanlah proses sekali jadi, melainkan sebuah proses yang berkelanjutan. Dengan mengikuti tahapan ini secara konsisten dan melibatkan pengguna dalam setiap proses, sistem yang dihasilkan tidak hanya cerdas, tetapi juga benar-benar berguna dan mudah diadopsi oleh penggunanya.

Pengumpulan dan Pengolahan Data Kasus

Salah satu tahapan paling krusial dalam membangun sistem CBR berbasis web adalah pengumpulan dan pengolahan data kasus. Tanpa data yang cukup dan berkualitas, sistem tidak akan mampu memberikan rekomendasi atau solusi yang akurat. Oleh karena itu, proses ini memerlukan perhatian khusus, baik dari segi sumber data, metode pengumpulan, hingga cara memproses dan menyimpannya dalam sistem.

Langkah pertama dalam pengumpulan data kasus adalah **menentukan sumber data yang dapat dipercaya**. Sumber ini bisa berasal dari pengalaman langsung pengguna, laporan teknis, arsip digital, hasil observasi lapangan, hingga wawancara dengan pakar. Idealnya, data yang dikumpulkan adalah kasus nyata yang sudah pernah terjadi dan memiliki dokumentasi solusi serta hasilnya.

Setiap kasus yang dikumpulkan perlu diuraikan secara sistematis menjadi **parameter-parameter terstruktur**. Misalnya, jika sistem digunakan untuk mendukung keputusan dalam lingkungan perairan, maka setiap kasus harus memuat data seperti suhu air, pH, kadar oksigen terlarut, dan sebagainya. Standarisasi ini sangat penting agar sistem dapat membandingkan antar kasus secara konsisten.

Namun, data dari dunia nyata sering kali tidak langsung dalam bentuk yang rapi. Karena itu, diperlukan tahap **pembersihan data (data cleaning)**. Data harus dicek untuk memastikan tidak ada nilai kosong, kesalahan input, atau duplikasi. Data yang tidak valid atau tidak lengkap bisa dihapus atau diisi ulang berdasarkan aturan atau estimasi yang telah ditentukan.

Setelah data dibersihkan, tahap berikutnya adalah **normalisasi**, yaitu menyamakan skala antar parameter agar tidak ada yang mendominasi perhitungan kemiripan secara tidak adil. Misalnya, jika satu parameter dinyatakan dalam rentang 0–100 dan yang lain 0–1, maka skala harus disesuaikan agar perhitungannya seimbang saat dicari nilai similarity.

Langkah selanjutnya adalah **pengkodean atau kategorisasi data**. Untuk parameter yang berbentuk teks atau deskripsi, sistem biasanya akan mengubahnya menjadi nilai numerik

atau simbol yang bisa diproses secara logis. Misalnya, tingkat kekeruhan bisa dikategorikan menjadi “jernih”, “sedang”, dan “keruh” yang kemudian diberi nilai 1, 2, dan 3.

Setelah data siap, setiap kasus dimasukkan ke dalam basis data sistem menggunakan format yang telah ditentukan. Proses ini bisa dilakukan secara manual (melalui form input dalam antarmuka web) atau otomatis (melalui impor file Excel, CSV, atau API). Penting untuk menyertakan **tanggal, identitas sumber, dan catatan tambahan** sebagai metadata yang berguna untuk audit di kemudian hari.

Selain input awal, sistem juga harus dirancang agar bisa menerima **penambahan kasus baru secara dinamis**. Ini memungkinkan sistem terus belajar dari waktu ke waktu. Agar proses ini berjalan lancar, sistem sebaiknya dilengkapi dengan validasi input dan notifikasi jika ada data yang kurang atau tidak sesuai format.

Salah satu fitur penting dalam pengolahan data kasus adalah **kemampuan pencarian dan penyaringan**. Pengguna sebaiknya dapat menelusuri kasus yang ada berdasarkan filter tertentu, seperti rentang nilai parameter atau kata kunci. Ini tidak hanya membantu dalam retrieve, tetapi juga berguna bagi pengguna yang ingin belajar dari kasus-kasus sebelumnya.

Dengan proses pengumpulan dan pengolahan data yang terstruktur dan berkualitas, sistem CBR akan memiliki fondasi yang kuat untuk menjalankan tugasnya. Data yang baik bukan hanya membuat sistem "bekerja", tetapi membuatnya benar-benar cerdas—mampu mengingat dengan akurat, membandingkan dengan adil, dan menyarankan solusi yang relevan dan terpercaya.

Membangun Logika Retrieve dan Reuse

Setelah data kasus berhasil dikumpulkan dan disusun dengan rapi dalam basis data, langkah selanjutnya dalam membangun sistem CBR berbasis web adalah mengembangkan logika *retrieve* dan *reuse*. Kedua proses ini merupakan inti dari sistem, karena di sinilah sistem benar-benar menunjukkan kecerdasannya: memilih kasus yang paling relevan dan menggunakan kembali pengalaman tersebut untuk menangani masalah baru.

Tahap **retrieve** bertugas mencari satu atau beberapa kasus yang paling mirip dengan kondisi yang sedang dihadapi. Sistem akan membandingkan nilai-nilai parameter dari input pengguna dengan semua kasus yang tersimpan, lalu menghitung tingkat kemiripannya menggunakan algoritma

similarity. Semakin kecil perbedaan antara parameter kasus baru dengan kasus lama, semakin tinggi nilai kemiripannya.

Perhitungan kemiripan biasanya menggunakan rumus seperti *Euclidean Distance*, *Manhattan Distance*, atau metode *Nearest Neighbor*. Untuk sistem yang memiliki parameter berbobot, metode *Weighted Similarity* lebih umum digunakan. Sistem akan mengalikan selisih nilai antar parameter dengan bobot masing-masing, lalu menjumlahkannya untuk mendapatkan skor akhir kemiripan.

Dalam praktiknya, sistem tidak hanya mencari satu kasus paling mirip, tetapi juga beberapa kasus teratas—misalnya 3 hingga 5 kasus terbaik. Hasil retrieve ini akan ditampilkan kepada pengguna sebagai referensi awal. Terkadang, sistem juga menampilkan informasi lengkap seperti nilai parameter, solusi yang pernah diberikan, dan hasilnya, agar pengguna bisa menilai relevansi solusi secara lebih menyeluruh.

Setelah kasus mirip ditemukan, tahap berikutnya adalah **reuse**, yaitu menggunakan solusi dari kasus yang dipilih untuk menyelesaikan masalah baru. Reuse bisa dilakukan dengan berbagai cara, tergantung dari desain sistem dan tingkat kepercayaan terhadap solusi sebelumnya. Pada sistem yang bersifat otomatis, solusi bisa langsung diberikan. Pada sistem semi-otomatis, pengguna masih diberi kesempatan untuk

menyesuaikan atau memilih solusi yang dianggap paling sesuai.

Dalam sistem berbasis web, implementasi logika retrieve dan reuse dilakukan di bagian backend—biasanya dengan bahasa pemrograman seperti PHP, Python (Flask atau Django), atau JavaScript (Node.js). Fungsi retrieve akan membaca input dari pengguna, memprosesnya melalui algoritma similarity, lalu mengambil data dari database untuk ditampilkan di antarmuka. Sementara fungsi reuse akan menyimpan keputusan yang dipilih pengguna, yang nantinya dapat digunakan dalam proses revise dan retain.

Agar performa sistem tetap tinggi, terutama jika basis kasus sangat besar, pengembang dapat mengoptimalkan logika retrieve dengan teknik pencarian cepat, seperti indexing atau pre-filtering berdasarkan kategori tertentu. Hal ini membuat proses pencocokan lebih efisien dan hasil dapat muncul dalam hitungan detik, bahkan untuk ribuan kasus.

Pada tahap reuse, penting juga untuk memberi ruang bagi pengguna untuk memberikan masukan. Misalnya, apakah solusi yang ditampilkan sudah cukup memuaskan? Apakah ada hal yang perlu diperbaiki? Masukan ini akan menjadi bahan evaluasi pada tahap revise, dan sangat berharga untuk meningkatkan kualitas sistem dalam jangka panjang.

Kelebihan pendekatan ini adalah fleksibilitasnya. Sistem tidak memaksakan satu jawaban benar, melainkan memberikan alternatif berdasarkan pengalaman yang pernah terbukti berhasil. Ini sangat cocok untuk situasi yang kompleks, ambigu, atau tidak memiliki aturan baku. CBR tidak meniru logika matematis kaku, melainkan mengikuti cara manusia berpikir—mengandalkan ingatan, perbandingan, dan pertimbangan.

Dengan logika retrieve dan reuse yang dibangun secara tepat, sistem CBR berbasis web akan mampu memberikan rekomendasi yang cepat, akurat, dan adaptif. Ini adalah fondasi dari sistem yang tidak hanya responsif, tetapi juga "berpengalaman"—karena setiap rekomendasi yang muncul adalah hasil dari pelajaran yang telah dilalui sebelumnya.

Proses Revisi dan Penyempurnaan Solusi (Revise)

Setelah sistem berhasil merekomendasikan solusi berdasarkan kasus sebelumnya melalui proses *retrieve* dan *reuse*, langkah selanjutnya dalam siklus Case-Based Reasoning adalah **revise**. Tahap ini bertujuan untuk memastikan bahwa solusi yang diusulkan benar-benar cocok dan efektif dalam konteks kasus baru. Di sinilah peran manusia atau sistem evaluasi

otomatis menjadi sangat penting, karena tidak semua solusi dari masa lalu bisa langsung diterapkan begitu saja.

Proses revise mencerminkan sikap kritis terhadap pengalaman. Dalam praktik sehari-hari, seseorang yang menghadapi masalah tidak serta-merta menyalin solusi yang dulu pernah digunakan. Biasanya, ada pertimbangan ulang—apakah konteksnya masih sama? Apakah ada faktor baru yang muncul? Hal inilah yang ditiru oleh sistem CBR pada tahap revisi, menjadikannya lebih cerdas dan adaptif.

Dalam sistem CBR berbasis web, proses revise bisa dilakukan secara **manual, semi-otomatis, atau otomatis**. Pada sistem manual, pengguna (misalnya pakar atau operator) meninjau kembali solusi yang diusulkan dan melakukan penyesuaian jika perlu. Di sistem semi-otomatis, sistem memberikan rekomendasi, lalu pengguna memvalidasi atau mengubahnya sebelum diterapkan. Sedangkan dalam sistem otomatis, algoritma khusus akan memverifikasi kecocokan solusi berdasarkan data tambahan atau logika tertentu.

Tahap ini juga menjadi momen evaluasi terhadap solusi. Jika solusi tidak berhasil atau menghasilkan hasil yang kurang baik, sistem harus mencatatnya sebagai bagian dari pembelajaran. Bahkan kegagalan pun berharga dalam CBR, karena dapat digunakan untuk memperbaiki sistem di masa

depan. Dengan mencatat bahwa solusi A tidak cocok untuk kondisi B, sistem akan lebih hati-hati jika menemukan pola yang mirip di lain waktu.

Proses revise juga bisa melibatkan **perbandingan hasil aktual dengan hasil yang diharapkan**. Misalnya, jika sistem digunakan dalam pemilihan metode budidaya ikan, maka hasil panen, pertumbuhan ikan, atau kondisi air bisa menjadi indikator keberhasilan. Jika hasilnya tidak sesuai, maka sistem perlu mencatat bahwa solusi tersebut kurang efektif, dan mungkin perlu penyesuaian sebelum disimpan.

Dalam sistem berbasis web, revisi juga bisa didukung dengan **fitur feedback pengguna**. Pengguna diberi kesempatan untuk menilai solusi, memberi komentar, atau menyarankan perbaikan. Fitur ini sangat bermanfaat untuk pengembangan sistem jangka panjang karena memperkaya basis pengetahuan dengan sudut pandang pengguna langsung.

Untuk menjaga konsistensi, setiap proses revisi harus disertai dokumentasi yang jelas. Sistem sebaiknya menyimpan versi awal solusi, hasil revisi, serta alasan revisi tersebut. Hal ini berguna tidak hanya untuk pelacakan, tetapi juga untuk analisis kualitas sistem secara berkala. Riwayat revisi juga bisa menjadi dasar pembobotan baru dalam retrieve di masa depan.

Beberapa sistem CBR modern bahkan mengintegrasikan machine learning untuk membantu dalam proses revise. Misalnya, sistem dapat mendeteksi pola kegagalan solusi dan secara otomatis menyarankan penyesuaian berdasarkan statistik historis. Integrasi seperti ini menjadikan proses revise lebih dinamis dan semakin menyerupai cara manusia belajar dari kesalahan.

Tahap revise juga merupakan titik kontrol kualitas dalam sistem CBR. Ia memastikan bahwa sistem tidak sekadar mengulang solusi lama, tetapi benar-benar memahami konteks baru. Proses ini menghindarkan sistem dari kesalahan berulang dan menjadikannya semakin cerdas seiring waktu.

Dengan adanya proses revise yang matang, sistem CBR berbasis web bukan hanya menjadi alat bantu keputusan, tetapi juga menjadi mitra belajar yang terus berkembang. Ia tidak hanya mengingat dan meniru, tapi juga mengevaluasi dan memperbaiki—karakteristik yang sangat penting dalam menciptakan sistem cerdas yang andal dan dapat dipercaya.

Retain: Menyimpan Pengalaman Baru

Tahap terakhir dalam siklus Case-Based Reasoning adalah **retain**, yaitu proses menyimpan kasus baru beserta solusinya ke dalam basis kasus sistem. Tahap ini sangat penting karena

di sinilah sistem benar-benar belajar dan berkembang dari waktu ke waktu. Tanpa tahap retain, sistem akan berhenti belajar dan hanya mengandalkan kasus-kasus lama, yang pada akhirnya membuatnya tidak lagi relevan dengan kondisi terbaru.

Dalam dunia nyata, retain bisa diibaratkan seperti seseorang mencatat pengalaman baru ke dalam jurnal pribadinya. Ketika suatu masalah berhasil diselesaikan—baik dengan solusi yang pernah digunakan atau setelah melalui proses revisi—pengalaman tersebut dicatat agar bisa digunakan lagi jika situasi serupa muncul di masa depan. Semakin sering seseorang mencatat dan merefleksikan pengalamannya, semakin bijak ia dalam mengambil keputusan. Hal yang sama berlaku dalam sistem CBR.

Retain tidak dilakukan sembarangan. Tidak semua kasus harus disimpan. Biasanya, sistem memiliki **kriteria atau filter** untuk menentukan kasus mana yang layak masuk ke dalam basis data. Misalnya, hanya kasus yang berhasil diselesaikan, memiliki parameter lengkap, dan mendapat nilai kepuasan tertentu dari pengguna yang akan disimpan. Ini mencegah terjadinya akumulasi data yang tidak berguna atau justru membingungkan sistem di kemudian hari.

Dalam sistem berbasis web, retain biasanya terjadi setelah pengguna menyelesaikan satu siklus interaksi dengan sistem. Pengguna akan diminta untuk memberi konfirmasi apakah solusi yang diberikan sudah berhasil atau perlu revisi. Jika berhasil, sistem akan menawarkan untuk menyimpan kasus tersebut. Jika tidak, sistem bisa menunda penyimpanan hingga proses revisi tuntas.

Proses retain melibatkan pencatatan **parameter input, solusi yang diberikan, hasil atau feedback pengguna, serta catatan revisi jika ada**. Semua informasi ini kemudian disusun menjadi satu paket kasus baru, lengkap dengan metadata seperti waktu, lokasi, atau identitas pengguna (jika diperlukan). Informasi ini akan digunakan dalam retrieve pada kasus-kasus berikutnya.

Salah satu tantangan dalam proses retain adalah **menghindari duplikasi kasus**. Jika dua kasus baru yang masuk ternyata sangat mirip dengan yang sudah ada, sistem harus mampu mengenali dan menolak penyimpanan ganda. Untuk itu, sistem perlu melakukan pengecekan kemiripan terlebih dahulu sebelum menambahkan kasus ke database.

Dalam beberapa sistem, retain bisa dilakukan secara otomatis. Namun dalam praktiknya, retain yang **diperkuat dengan penilaian manusia** sering kali memberikan hasil yang lebih

baik. Misalnya, pakar atau pengguna dapat meninjau apakah solusi memang layak dicatat sebagai pengetahuan baru, atau masih perlu dievaluasi lebih lanjut.

Retain juga memberi peluang untuk menambahkan **informasi pembelajaran tambahan**, seperti insight atau catatan khusus dari pengguna. Hal ini sangat berguna jika sistem digunakan dalam konteks pelatihan, pendidikan, atau riset, di mana bukan hanya solusi yang penting, tapi juga proses berpikir dan pertimbangan yang digunakan.

Seiring waktu, basis kasus akan bertambah banyak dan kompleks. Karena itu, proses retain juga perlu diikuti dengan **manajemen basis kasus**, seperti pengelompokan, penghapusan kasus usang, atau pemeliharaan performa database. Hal ini penting agar sistem tetap efisien dan tidak terbebani oleh data yang terlalu besar atau tidak relevan.

Dengan proses retain yang dijalankan secara sistematis dan selektif, sistem CBR berbasis web akan menjadi semakin kuat dan kaya pengalaman. Ia tidak hanya menjadi alat bantu keputusan, tetapi juga berkembang menjadi sebuah sistem pembelajaran yang hidup, terus tumbuh dan menyesuaikan diri seiring dengan bertambahnya pengetahuan yang dimilikinya.

Bab 4 – Dari Data ke Keputusan

Cara Sistem Mengambil Keputusan

Setelah melalui seluruh tahapan siklus Case-Based Reasoning, kini saatnya sistem mengambil peran sebagai pengambil keputusan. Proses ini tidak hanya soal menyajikan informasi, tetapi bagaimana sistem mengubah data menjadi rekomendasi konkret yang bisa digunakan oleh manusia. Keputusan yang dihasilkan merupakan hasil akhir dari proses retrieve, reuse, revise, dan retain yang telah berjalan secara terpadu.

Cara sistem mengambil keputusan dalam CBR sangat berbeda dengan sistem aturan (*rule-based system*) yang biasanya bergantung pada logika "jika–maka". Dalam CBR, keputusan diambil berdasarkan tingkat kemiripan antara kasus baru dengan kasus-kasus lama, serta efektivitas solusi yang pernah diterapkan. Pendekatan ini jauh lebih fleksibel karena tidak membutuhkan aturan baku yang kaku dan bisa menyesuaikan dengan berbagai variasi kasus nyata.

Langkah awal dalam proses pengambilan keputusan adalah menerima input dari pengguna. Input ini biasanya berupa nilai-nilai parameter yang menggambarkan kondisi saat ini. Misalnya, dalam sistem untuk analisis kualitas air, input bisa berupa pH, suhu, kadar oksigen, dan sebagainya. Sistem akan

mencocokkan input ini dengan database kasus untuk menemukan kasus-kasus yang paling mirip.

Setelah retrieve dilakukan, sistem menyusun daftar kasus yang relevan, biasanya disertai skor kemiripan. Kasus dengan skor tertinggi dianggap paling relevan. Namun, sistem tidak serta-merta hanya mengambil satu solusi dari kasus tersebut. Dalam banyak sistem, pengguna diberikan beberapa opsi kasus yang mirip, sehingga keputusan akhir bisa disesuaikan dengan pertimbangan tambahan yang mungkin tidak tersedia di data.

Keputusan yang diberikan oleh sistem bersifat rekomendatif. Artinya, sistem menyarankan solusi berdasarkan pengalaman sebelumnya, tetapi tetap memberi ruang bagi pengguna untuk mengevaluasi atau memodifikasi saran tersebut. Inilah kelebihan CBR dibanding sistem otomatis murni—ia melibatkan unsur manusia dalam pengambilan keputusan, menjadikannya lebih realistis dan dapat dipercaya.

Beberapa sistem CBR dilengkapi dengan modul penguat keputusan, seperti indikator kepercayaan (*confidence level*) atau riwayat keberhasilan solusi. Misalnya, sistem bisa menyatakan bahwa solusi A telah berhasil diterapkan pada 12 kasus serupa dengan tingkat keberhasilan 90%. Informasi

semacam ini sangat membantu pengguna dalam mempertimbangkan pilihan terbaik.

Sistem yang baik juga mampu mempertimbangkan **faktor non-teknis** dalam pengambilan keputusan. Sebagai contoh, solusi terbaik secara teknis mungkin membutuhkan biaya tinggi atau waktu lama. Jika sistem menyimpan informasi semacam ini dari kasus sebelumnya, maka ia bisa merekomendasikan alternatif solusi yang lebih praktis sesuai kondisi pengguna saat ini.

Selain itu, sistem juga dapat memperhitungkan **pengaruh kontekstual**, seperti waktu, lokasi, atau musim. Misalnya, suatu metode berhasil di wilayah dataran tinggi, tetapi belum tentu cocok di dataran rendah. Dengan kemampuan mengingat konteks dari kasus sebelumnya, sistem menjadi lebih selektif dalam menyarankan solusi yang benar-benar relevan.

Keputusan yang diberikan oleh sistem bisa bersifat tunggal atau bertahap. Dalam kasus kompleks, sistem dapat memberikan langkah-langkah bertingkat, misalnya: langkah pertama, lakukan A; jika berhasil, lanjutkan ke B; jika tidak, pertimbangkan C. Pendekatan ini meniru proses berpikir manusia dan sangat berguna dalam sistem yang membutuhkan strategi bertahap.

Dengan segala proses dan perhitungan di baliknya, sistem CBR sejatinya tidak hanya menyampaikan jawaban, tapi membimbing pengguna dari data mentah menuju keputusan yang berdasar. Ia tidak menggantikan peran manusia, tetapi memperkuat kemampuan manusia dalam mengambil keputusan yang lebih cepat, cerdas, dan berpengalaman.

Simulasi Penggunaan Sistem Cerdas

Untuk memahami bagaimana sistem CBR bekerja secara nyata, mari kita simulasikan penggunaan sistem cerdas berbasis web yang dirancang untuk memberikan rekomendasi berdasarkan parameter tertentu. Simulasi ini akan membantu menggambarkan bagaimana seluruh tahapan CBR berlangsung mulai dari input hingga keluaran keputusan, sehingga pembaca dapat melihat alur sistem dengan lebih konkret.

Misalnya, sistem yang digunakan adalah untuk membantu pengguna memilih metode budidaya ikan air tawar berdasarkan kondisi kualitas air. Seorang pengguna mengakses aplikasi web dan mengisi formulir yang telah disediakan: suhu air 28°C, pH 6.8, DO (oksigen terlarut) 5 mg/L, dan kekeruhan sedang. Parameter-parameter ini akan menjadi *input kasus baru* yang akan diproses oleh sistem.

Setelah pengguna menekan tombol “Cari Solusi”, sistem mulai bekerja pada tahap *retrieve*. Ia akan membandingkan parameter yang diinputkan dengan seluruh data kasus yang ada dalam basis data. Sistem menghitung nilai kemiripan menggunakan algoritma tertentu, misalnya *weighted similarity*, lalu menghasilkan daftar kasus-kasus yang memiliki tingkat kemiripan tertinggi.

Hasil retrieve menampilkan tiga kasus teratas dengan tingkat kemiripan masing-masing 93%, 88%, dan 85%. Sistem menampilkan masing-masing kasus tersebut secara ringkas: kondisi lingkungan, metode budidaya yang digunakan, dan hasilnya. Misalnya, pada kasus dengan 93% kemiripan, metode yang digunakan adalah sistem kolam terpal dengan aerasi tambahan, yang menghasilkan pertumbuhan ikan optimal selama 45 hari.

Tahap berikutnya adalah *reuse*, yaitu penerapan solusi dari kasus dengan kemiripan tertinggi ke kasus baru. Sistem menyarankan pengguna untuk menggunakan metode budidaya kolam terpal dengan tambahan aerasi dan filter sederhana. Solusi ini ditampilkan bersama deskripsi langkah-langkah, estimasi biaya, dan tips tambahan yang sebelumnya digunakan dalam kasus aslinya.

Namun sistem tidak berhenti di sana. Ia memberi opsi kepada pengguna untuk melakukan evaluasi awal atau *revise*. Pengguna dapat menyatakan apakah kondisi lingkungan di tempat mereka benar-benar sama atau ada faktor tambahan yang mungkin berbeda, seperti curah hujan atau kualitas pakan. Jika ada perbedaan signifikan, pengguna bisa menyesuaikan solusi sebelum diterapkan.

Setelah revisi selesai dan solusi diterapkan, pengguna kembali ke sistem dan mengisi umpan balik: hasil panen baik, waktu panen tepat, dan pertumbuhan ikan sesuai harapan. Dengan hasil seperti itu, sistem melanjutkan ke tahap *retain*—kasus baru ini disimpan ke dalam basis data, lengkap dengan parameter input, solusi, hasil, dan evaluasi pengguna.

Simulasi ini menunjukkan bahwa sistem tidak hanya menyajikan data, tapi juga berperan aktif dalam membantu proses pengambilan keputusan secara bertahap. Pengguna tidak dibiarkan menebak-nebak, tetapi diarahkan secara logis dan berdasarkan pengalaman nyata yang telah tersimpan dalam sistem.

Dalam jangka panjang, semakin banyak pengguna yang menggunakan dan memberikan umpan balik terhadap sistem, semakin kaya pula basis kasus yang dimiliki. Sistem menjadi semakin cerdas, semakin akurat, dan semakin bisa diandalkan

dalam berbagai situasi karena telah belajar dari banyak pengalaman.

Melalui simulasi seperti ini, kita dapat melihat betapa efektifnya pendekatan CBR dalam sistem cerdas berbasis web. Ia tidak hanya menyimpan dan menampilkan data, tetapi juga memfasilitasi proses berpikir manusia: mengingat, membandingkan, menyesuaikan, dan akhirnya—mengambil keputusan terbaik.

Menyempurnakan Solusi Lewat Validasi

Setiap solusi yang diberikan oleh sistem CBR, sebaik dan serasional apapun, tetap memerlukan tahap **validasi**. Validasi bertujuan untuk memastikan bahwa solusi tersebut benar-benar bekerja di dunia nyata dan tidak hanya tampak baik secara teori atau perhitungan. Tanpa proses validasi, sistem hanya akan menjadi mesin saran tanpa jaminan keefektifan, padahal salah satu kekuatan utama CBR adalah pembelajaran dari pengalaman nyata.

Validasi bisa dilakukan dengan berbagai cara, tergantung dari konteks penggunaannya. Pada sistem berbasis web, validasi umumnya melibatkan **feedback dari pengguna** setelah solusi dijalankan. Misalnya, setelah pengguna menerapkan metode yang direkomendasikan, sistem akan meminta umpan balik:

apakah hasilnya memuaskan, apa yang berhasil, dan apa yang tidak berjalan sesuai harapan.

Feedback tersebut menjadi bagian penting dari *loop pembelajaran sistem*. Jika hasil validasi menunjukkan bahwa solusi berhasil dengan baik, maka kasus tersebut bisa langsung masuk ke tahap *retain*. Namun jika hasilnya kurang baik atau tidak sesuai, sistem akan mencatatnya sebagai pengalaman yang perlu dievaluasi lebih lanjut—baik oleh pengguna maupun pengembang.

Beberapa sistem bahkan menggabungkan validasi dengan **pengukuran hasil otomatis**. Misalnya, pada sistem berbasis sensor atau IoT, hasil penerapan solusi bisa diukur melalui indikator digital—seperti suhu, kualitas air, atau tingkat pertumbuhan. Data ini langsung dikirim ke sistem sebagai bentuk validasi tanpa perlu intervensi manual.

Validasi juga berfungsi sebagai bentuk *refleksi*. Bagi pengguna, tahap ini menjadi kesempatan untuk mempertimbangkan kembali keputusan yang diambil, mengevaluasi efektivitas tindakan, dan mencatat pembelajaran. Bagi sistem, validasi berfungsi sebagai jaminan kualitas. Ia membantu menjaga agar basis kasus hanya terisi oleh solusi yang sudah terbukti.

Dalam sistem yang dirancang untuk digunakan dalam jangka panjang, hasil validasi dapat digunakan sebagai **penentu skor kepercayaan (confidence score)** pada kasus yang tersimpan. Semakin banyak kasus yang divalidasi positif, semakin tinggi skor kepercayaan yang diberikan, dan semakin sering solusi tersebut akan direkomendasikan kembali kepada pengguna lain.

Selain membantu pengambilan keputusan, validasi juga berguna dalam **audit sistem**. Pengelola dapat melacak solusi apa saja yang telah diuji di lapangan, mana yang paling sering digunakan, dan mana yang perlu diperbaiki atau dihapus dari basis kasus. Hal ini penting untuk menjaga kualitas sistem, terutama jika digunakan di bidang-bidang yang sensitif seperti kesehatan atau lingkungan.

Validasi juga dapat melibatkan **pihak ketiga atau pakar**. Dalam beberapa sistem, terutama yang digunakan untuk tujuan profesional atau akademik, setiap kasus yang baru akan diperiksa oleh pakar sebelum disimpan permanen. Ini menambah tingkat kepercayaan terhadap sistem dan memperkuat posisi sistem sebagai alat bantu yang kredibel.

Tahap validasi tidak hanya menyempurnakan solusi, tetapi juga menyempurnakan sistem secara keseluruhan. Dengan rutin menerima dan mengelola umpan balik dari pengguna,

sistem CBR menjadi lebih tangguh, adaptif, dan terpercaya. Ia bukan hanya sistem yang mengingat dan menyarankan, tetapi juga sistem yang belajar dari realitas.

Dengan validasi sebagai penutup dari satu siklus lengkap, sistem CBR berbasis web menjadi ekosistem pembelajaran yang terus tumbuh. Ia tidak hanya menjawab pertanyaan saat ini, tapi juga mempersiapkan jawaban yang lebih baik untuk pertanyaan di masa depan.

Bab 5 – Studi Kasus: Sistem dalam Aksi

Untuk benar-benar memahami cara kerja dan manfaat dari sistem Case-Based Reasoning (CBR) berbasis web, tak ada cara yang lebih baik daripada melihat langsung bagaimana sistem ini digunakan dalam dunia nyata. Studi kasus ini memberikan gambaran konkret tentang bagaimana seluruh konsep yang telah dijelaskan sebelumnya diimplementasikan dalam bentuk sistem yang berjalan dan memberikan dampak nyata bagi penggunanya.

Misalnya, kita ambil contoh sistem yang dikembangkan untuk membantu petani ikan air tawar memilih metode budidaya yang optimal berdasarkan parameter kualitas air. Sistem ini dibangun menggunakan pendekatan CBR, di mana pengguna dapat memasukkan nilai-nilai seperti suhu air, pH, kadar oksigen terlarut, dan kekeruhan. Sistem kemudian akan membandingkan input tersebut dengan basis kasus berisi pengalaman budidaya sebelumnya yang telah berhasil.

Seorang petani pemula dari daerah dataran rendah ingin memulai budidaya ikan lele. Ia mengakses aplikasi web yang telah disiapkan, lalu memasukkan data: suhu 30°C, pH 7, DO 4 mg/L, dan kekeruhan sedang. Setelah itu, sistem memproses data dengan mencari kasus serupa dalam basis kasus yang telah diisi dari pengalaman puluhan pengguna sebelumnya.

Dalam hitungan detik, sistem menampilkan tiga kasus dengan kemiripan tinggi, lengkap dengan rekomendasi metode budidaya yang digunakan, seperti jenis kolam (terpal atau beton), sistem aerasi yang dipakai, frekuensi pemberian pakan, hingga jenis pakan yang terbukti efektif dalam kondisi serupa. Petani tersebut memilih salah satu rekomendasi, lalu sistem menyarankan langkah-langkah pelaksanaan secara rinci.

Setelah satu siklus budidaya selesai, petani kembali ke sistem dan memberikan feedback: pertumbuhan ikan cepat, angka kematian rendah, dan hasil panen sesuai target. Sistem mencatat informasi ini, dan dengan validasi tersebut, kasus baru disimpan dalam basis data sebagai pengalaman yang berhasil dan bisa digunakan untuk rekomendasi selanjutnya.

Dari sisi teknis, sistem ini menggunakan PHP dan MySQL untuk backend, serta HTML, CSS, dan JavaScript untuk antarmuka pengguna. Algoritma similarity yang diterapkan adalah weighted Euclidean distance, dengan bobot berbeda untuk setiap parameter—misalnya pH dan DO diberi bobot lebih besar karena lebih menentukan keberhasilan budidaya.

Studi kasus ini menunjukkan bagaimana teknologi yang awalnya terkesan kompleks seperti CBR dapat diterjemahkan menjadi aplikasi praktis yang mudah digunakan oleh

masyarakat luas, bahkan oleh pengguna yang tidak memiliki latar belakang teknologi. Yang dibutuhkan hanyalah input data sederhana, dan sistem akan "mengambil alih" proses pencarian solusi berdasarkan pengalaman-pengalaman terbaik sebelumnya.

Tak hanya memberikan solusi instan, sistem ini juga mendidik penggunanya. Petani yang sebelumnya tidak tahu tentang pentingnya kadar oksigen atau kekeruhan, kini menjadi lebih sadar dan terlibat dalam pengambilan keputusan teknis. Sistem secara tidak langsung menjadi media pembelajaran berbasis pengalaman kolektif.

Studi kasus lain juga bisa ditemukan dalam bidang pendidikan, di mana guru menggunakan sistem serupa untuk memilih metode pengajaran yang paling efektif berdasarkan karakteristik siswa. Di bidang medis, sistem membantu mendiagnosis penyakit langka dengan mencocokkan gejala yang tidak umum terhadap database kasus dari rumah sakit-rumah sakit lain. Di semua kasus ini, prinsip yang sama berlaku: belajar dari pengalaman nyata untuk membuat keputusan lebih cerdas.

Melalui studi kasus ini, terlihat jelas bahwa sistem CBR tidak hanya teoritis, tetapi sangat praktis dan bermanfaat. Dengan sistem yang dirancang baik, pengalaman individu berubah

menjadi pengetahuan kolektif, dan keputusan-keputusan menjadi lebih terarah, cepat, dan efektif.

Perbandingan Hasil dan Evaluasi Sistem

Setelah sistem Case-Based Reasoning (CBR) dijalankan dalam dunia nyata melalui studi kasus, langkah berikutnya adalah melakukan **evaluasi menyeluruh terhadap performanya**. Evaluasi ini tidak hanya penting untuk menilai apakah sistem bekerja, tetapi juga untuk mengetahui seberapa besar manfaat yang diberikannya dibandingkan metode konvensional atau sistem lain yang serupa.

Dalam studi kasus budidaya ikan air tawar, evaluasi dilakukan dengan membandingkan hasil panen petani yang menggunakan sistem CBR dengan mereka yang tidak menggunakan sistem (kontrol). Indikator yang digunakan antara lain: pertumbuhan ikan per hari, tingkat kelangsungan hidup (survival rate), waktu panen, dan biaya produksi. Hasilnya menunjukkan bahwa petani yang menggunakan sistem mendapatkan efisiensi waktu dan biaya yang lebih baik, dengan kualitas panen yang stabil.

Dari sisi waktu, keputusan yang sebelumnya membutuhkan diskusi panjang atau coba-coba kini bisa diambil dalam hitungan menit. Sistem membantu memangkas proses

pencarian informasi dan membuat pengguna lebih percaya diri dalam mengambil langkah. Dalam kasus ini, waktu panen bisa dipercepat sekitar 10–15% karena solusi yang diberikan langsung relevan dan telah terbukti.

Dalam hal keakuratan, tingkat keberhasilan solusi yang disarankan sistem berada di atas 80%, berdasarkan validasi pengguna. Artinya, sebagian besar rekomendasi sistem terbukti efektif dan bisa diterapkan langsung dengan sedikit atau tanpa modifikasi. Ini menunjukkan bahwa basis kasus yang digunakan sistem cukup kuat dan representatif terhadap kondisi di lapangan.

Selain evaluasi kuantitatif, dilakukan juga **evaluasi kualitatif** melalui survei kepuasan pengguna. Sebagian besar pengguna menyatakan puas karena sistem mudah digunakan, memberikan informasi yang lengkap, dan membantu mereka memahami hubungan antarparameter. Beberapa bahkan mengungkapkan bahwa sistem membuat mereka lebih "melek data" dan terbuka untuk belajar dari pengalaman sebelumnya.

Namun, evaluasi juga menunjukkan beberapa **keterbatasan sistem**. Misalnya, ketika pengguna menghadapi kondisi ekstrem atau di luar jangkauan data kasus yang ada, sistem menjadi kurang akurat atau tidak bisa memberikan solusi yang memadai. Hal ini mengindikasikan perlunya perluasan basis

kasus dan fleksibilitas sistem dalam menyesuaikan dengan kasus langka.

Untuk mengatasi hal tersebut, dilakukan uji coba integrasi sistem dengan fitur upload data dan kolaborasi antarpengguna. Dengan begitu, kasus-kasus unik dapat disimpan dan digunakan sebagai referensi di masa depan. Ini menunjukkan bahwa sistem CBR tidak bersifat statis, melainkan bisa terus berkembang sesuai dengan kontribusi pengguna.

Sebagai bagian dari evaluasi menyeluruh, pengembang sistem juga menganalisis **kinerja teknis** sistem: waktu respon, kecepatan retrieve, dan efisiensi penyimpanan data. Hasilnya cukup baik—rata-rata waktu pencarian solusi berada di bawah 2 detik meskipun basis kasus telah berisi ratusan data. Optimasi pada struktur database dan logika retrieve berkontribusi besar dalam hal ini.

Secara keseluruhan, evaluasi menunjukkan bahwa sistem CBR berbasis web memberikan dampak nyata bagi pengguna dan layak untuk dikembangkan lebih lanjut. Perbandingan hasil menunjukkan peningkatan efisiensi, akurasi keputusan, serta peningkatan pemahaman pengguna terhadap masalah yang mereka hadapi.

Melalui evaluasi sistem ini, kita tidak hanya mengetahui keberhasilan sistem dari sisi teknologi, tetapi juga dari segi

kebermanfaatannya dalam kehidupan nyata. CBR terbukti bukan hanya pendekatan cerdas secara teori, tetapi juga alat bantu yang konkret, fleksibel, dan bisa diandalkan untuk mendukung keputusan di berbagai bidang.

Proses Pembelajaran Sistem Secara Berkala

Salah satu keunggulan utama dari sistem Case-Based Reasoning (CBR) adalah kemampuannya untuk **belajar secara berkelanjutan**. Tidak seperti sistem berbasis aturan yang cenderung statis, CBR dirancang untuk terus berkembang melalui pengalaman baru. Proses pembelajaran ini terjadi secara bertahap dan berulang, menciptakan sistem yang semakin cerdas seiring dengan waktu dan banyaknya kasus yang dialami.

Pembelajaran dalam CBR terjadi setiap kali siklus lengkap—dari retrieve hingga retain—berhasil dijalankan. Ketika sistem menerima kasus baru, menyarankan solusi, kemudian mendapatkan validasi dari pengguna, ia akan menyimpan pengalaman itu sebagai bagian dari basis pengetahuan yang terus bertambah. Ini menjadikan sistem sebagai entitas yang hidup, bukan sekadar kumpulan kode dan data.

Agar proses pembelajaran ini berlangsung efektif, sistem perlu dirancang dengan mekanisme pemantauan dan

penyegaran data secara berkala. Misalnya, dilakukan pembersihan terhadap kasus-kasus usang yang sudah tidak relevan, atau penyusunan ulang indeks pencarian untuk mempercepat retrieve. Sistem juga bisa dianalisis untuk mengetahui parameter apa yang sering muncul dan kasus mana yang paling banyak digunakan.

Pengembang juga dapat menambahkan fitur **analisis tren** berdasarkan akumulasi data dari waktu ke waktu. Dari situ, pengguna bisa melihat pola—seperti metode yang paling sering berhasil dalam kondisi tertentu, waktu panen terbaik, atau kombinasi parameter yang paling aman. Dengan memvisualisasikan data ini, sistem tidak hanya memberi solusi, tetapi juga memberikan wawasan strategis.

Penting juga bagi sistem untuk memiliki kemampuan **adaptasi terhadap perubahan lingkungan**. Misalnya, jika ada perubahan iklim atau kebijakan yang memengaruhi pola budidaya, sistem perlu memperbarui pendekatannya berdasarkan kasus terbaru yang masuk. Dengan begitu, sistem tetap relevan dan responsif terhadap dunia nyata yang terus berubah.

Proses pembelajaran berkala ini juga membutuhkan peran aktif dari pengguna. Semakin banyak pengguna yang terlibat, memberikan input berkualitas, dan membagikan hasilnya,

semakin kaya basis kasus yang dimiliki. Inilah mengapa sistem CBR yang sukses sering kali berbentuk platform terbuka atau komunitas digital, bukan hanya sistem tertutup.

Selain pembelajaran dari kasus sukses, sistem juga bisa belajar dari **kegagalan**. Kasus-kasus yang menghasilkan solusi kurang tepat akan dianalisis untuk memperbaiki logika similarity atau bobot parameter. Bahkan dalam beberapa implementasi, sistem bisa menandai kasus bermasalah agar tidak terlalu diandalkan dalam retrieve di masa mendatang.

Dalam sistem yang lebih canggih, pembelajaran berkala juga dapat diotomatisasi dengan bantuan **machine learning**. Misalnya, sistem dapat menyesuaikan bobot parameter secara otomatis berdasarkan efektivitas solusi sebelumnya. Atau sistem bisa mengelompokkan kasus dengan teknik clustering agar lebih mudah dikelola dan dianalisis.

Dengan adanya proses pembelajaran yang terus-menerus, sistem CBR akan menjadi lebih tajam dalam pengambilan keputusan, lebih cepat dalam pencarian, dan lebih bijak dalam menyarankan solusi. Inilah kekuatan sejati dari sistem cerdas: kemampuannya untuk tumbuh dari pengalaman.

Pada akhirnya, proses pembelajaran sistem bukan hanya soal menambah data, tetapi soal **meningkatkan kualitas keputusan** dari waktu ke waktu. CBR bukan sekadar

teknologi, tetapi filosofi pemecahan masalah yang manusiawi—belajar dari masa lalu, memahami masa kini, dan mempersiapkan jawaban untuk masa depan.

Bab 6 – Refleksi dan Pengembangan Lanjutan

Apa yang Sudah Dicapai

Setelah menelusuri berbagai konsep, proses, dan contoh nyata dari sistem Case-Based Reasoning (CBR) berbasis web, kini saatnya melakukan refleksi: sejauh mana sistem ini telah memberikan manfaat? Apa yang telah berhasil dicapai? Dan bagaimana nilai-nilai dari pendekatan ini bisa dirasakan dalam penerapan sehari-hari?

Pertama dan yang paling jelas, sistem CBR berhasil **mengubah pengalaman menjadi pengetahuan terstruktur**. Pengalaman yang sebelumnya hanya tersimpan di kepala individu atau laporan-laporan manual kini dikumpulkan, dikelola, dan dibagikan dalam bentuk sistem digital. Ini adalah pencapaian besar dalam dunia informasi—karena pengalaman yang terdistribusi kini bisa dimanfaatkan oleh siapa saja yang membutuhkannya.

Penerapan CBR juga telah terbukti mampu **mempercepat pengambilan keputusan** dalam berbagai bidang. Dari budidaya perikanan, pendidikan, hingga layanan kesehatan, sistem memberikan rekomendasi berbasis kasus nyata dalam waktu yang jauh lebih singkat dibandingkan pendekatan tradisional. Ini berdampak langsung pada efisiensi waktu, biaya, dan energi.

Lebih dari itu, sistem ini turut **memberdayakan pengguna**. Petani, guru, teknisi, atau bahkan pengguna awam dapat menjadi bagian dari sistem pengetahuan, bukan hanya sebagai penerima informasi, tapi juga sebagai kontributor pengalaman. Dengan memberikan feedback dan berbagi hasil penggunaan sistem, mereka ikut membentuk kecerdasan kolektif.

Dalam sisi teknis, pencapaian lain yang signifikan adalah **kemampuan sistem untuk terus belajar**. Retain bukan hanya fitur, tapi juga cerminan filosofi bahwa sistem cerdas harus berkembang bersama waktu. Sistem CBR yang dibahas dalam buku ini berhasil menunjukkan bahwa mesin bisa "mengingat" dan "mengadaptasi" dengan cara yang menyerupai manusia.

Pengalaman implementasi juga memperlihatkan bahwa pendekatan CBR relatif **mudah diadopsi dan diintegrasikan** ke dalam sistem berbasis web. Dengan teknologi yang terbuka dan tersedia secara luas, pengembang dapat membangun sistem CBR dari skala kecil hingga besar, dengan fleksibilitas tinggi dan tanpa kebutuhan komputasi berat seperti pada sistem AI berbasis deep learning.

Tidak kalah penting, sistem ini mampu menyajikan **keputusan yang transparan dan bisa ditelusuri kembali**.

Setiap solusi yang diberikan memiliki riwayat: dari mana asal kasusnya, seperti apa kondisi sebelumnya, dan bagaimana hasilnya. Ini membangun kepercayaan pengguna terhadap sistem dan memudahkan proses audit atau pelacakan jika terjadi kesalahan.

Secara tidak langsung, CBR juga berperan dalam **transfer pengetahuan antar individu dan generasi**. Dalam dunia kerja, misalnya, ketika seorang ahli pensiun atau berpindah tugas, pengalaman dan keahliannya tidak serta-merta hilang. Sistem CBR memungkinkan pengalamannya tetap hidup dalam bentuk basis kasus yang bisa digunakan oleh penerusnya.

Dalam hal pendidikan, buku ini sendiri telah menjadi bagian dari pencapaian. Ia berfungsi sebagai pengantar yang memperkenalkan pembaca pada konsep sistem cerdas berbasis pengalaman, membimbing mereka langkah demi langkah, dan memberikan inspirasi bagaimana teknologi bisa disusun secara sederhana namun berdampak besar.

Apa yang telah dicapai sejauh ini adalah fondasi yang kuat untuk melangkah lebih jauh. Dengan sistem yang sudah berjalan, pengguna yang aktif, dan data yang terus bertambah, langkah selanjutnya adalah membayangkan—dan

mewujudkan—masa depan dari sistem cerdas yang benar-benar hidup dan berkembang.

Ide Pengembangan di Masa Depan

Dengan fondasi sistem yang sudah terbentuk dan pengalaman implementasi yang telah menunjukkan hasil, kini saatnya melihat ke depan: bagaimana sistem Case-Based Reasoning (CBR) bisa dikembangkan lebih jauh? Teknologi terus bergerak cepat, dan sistem cerdas harus mampu menyesuaikan diri dengan perubahan kebutuhan, tantangan baru, serta potensi kolaborasi yang lebih luas.

Salah satu ide pengembangan utama adalah **integrasi CBR dengan teknologi Internet of Things (IoT)**. Dalam banyak bidang seperti pertanian, perikanan, dan lingkungan, data dari sensor bisa dikirim secara langsung ke sistem CBR tanpa perlu input manual. Sistem bisa menerima pembacaan suhu, pH, kelembapan, atau kadar oksigen secara real-time, lalu memberikan rekomendasi otomatis berdasarkan basis kasus yang ada.

Selain itu, sistem CBR dapat ditingkatkan dengan **kemampuan prediksi menggunakan machine learning**. Misalnya, berdasarkan kecenderungan data yang masuk, sistem bisa memprediksi potensi masalah sebelum terjadi,

atau menyarankan tindakan preventif. Kombinasi antara pengalaman masa lalu (CBR) dan pembelajaran dari pola (ML) akan menciptakan sistem yang tidak hanya reaktif, tetapi juga proaktif.

Dari sisi pengalaman pengguna, pengembangan ke arah **antarmuka yang lebih interaktif dan visual** juga sangat potensial. Visualisasi data kasus, tren parameter, atau peta kemiripan antar kasus bisa membantu pengguna memahami konteks dan alasan di balik rekomendasi. Ini memperkuat kepercayaan terhadap sistem dan meningkatkan kenyamanan dalam penggunaannya.

Selanjutnya, sistem dapat dikembangkan dalam bentuk **aplikasi mobile**. Akses melalui perangkat seluler akan memungkinkan pengguna lapangan—seperti petani, teknisi, atau tenaga medis—untuk menggunakan sistem kapan saja dan di mana saja. Bahkan dalam kondisi tanpa koneksi internet stabil, aplikasi bisa diatur untuk bekerja secara offline dan melakukan sinkronisasi saat online.

Aspek lain yang penting untuk pengembangan adalah **kerja sama antar pengguna atau organisasi**. Sistem CBR bisa dirancang sebagai platform kolaboratif di mana pengguna dari berbagai wilayah dapat saling berbagi kasus dan solusi.

Dengan demikian, sistem tidak hanya menjadi alat individual, tetapi juga wadah pengetahuan kolektif yang terus diperluas.

Keamanan dan privasi data juga perlu menjadi fokus dalam pengembangan ke depan. Sistem perlu memiliki mekanisme untuk melindungi data sensitif, menetapkan hak akses, serta menyediakan audit trail agar setiap aktivitas dalam sistem bisa ditelusuri. Ini sangat penting jika sistem digunakan dalam sektor yang bersifat strategis atau mengandung data pribadi.

Dari sisi konten, pengembangan sistem juga mencakup **penambahan metadata dan dimensi kontekstual**. Informasi seperti musim, wilayah geografis, jenis pengguna, atau tujuan akhir bisa menjadi lapisan tambahan yang membantu sistem menyaring kasus dengan lebih relevan. Konteks menjadi penentu penting dalam efektivitas rekomendasi.

Sistem juga dapat dibuka untuk **kontribusi komunitas berbasis reputasi**, di mana pengguna yang sering memberikan feedback berkualitas atau membagikan kasus sukses mendapat peringkat atau lencana. Pendekatan ini telah terbukti efektif di berbagai platform digital dan bisa diterapkan untuk mendorong partisipasi aktif dalam ekosistem CBR.

Akhirnya, pengembangan sistem bukan hanya tentang menambahkan fitur, tetapi menjaga agar sistem tetap **mudah**

digunakan, dapat dipelajari, dan bermanfaat nyata. Dengan terus mendengarkan pengguna, mengadopsi teknologi baru secara bijak, dan menjaga prinsip pembelajaran dari pengalaman, sistem CBR akan tetap relevan dan bahkan semakin penting dalam menghadapi kompleksitas masa depan.

Tantangan dan Peluang Penerapan Lain

Meski sistem Case-Based Reasoning (CBR) menawarkan banyak keunggulan, penerapannya di dunia nyata tidak lepas dari tantangan. Tantangan ini bisa datang dari sisi teknis, non-teknis, hingga sosial. Namun di balik setiap tantangan, tersimpan pula peluang besar untuk pengembangan lebih lanjut dan penerapan di berbagai bidang baru yang belum tergarap optimal.

Salah satu tantangan utama adalah **ketersediaan data berkualitas**. Sistem CBR sangat bergantung pada kasus-kasus terdahulu. Jika data awal minim, tidak lengkap, atau tidak representatif, maka solusi yang ditawarkan pun cenderung tidak akurat. Karena itu, pada tahap awal implementasi, pengumpulan data sering menjadi pekerjaan yang berat dan memakan waktu.

Selain itu, sistem CBR menuntut **standarisasi format data**. Dalam banyak organisasi atau komunitas, data sering dicatat dengan cara berbeda-beda, baik dari segi istilah, satuan, maupun tingkat detail. Tanpa format yang seragam, sistem akan kesulitan membaca dan membandingkan antar kasus. Diperlukan pedoman input yang jelas agar data yang masuk konsisten dan dapat dibandingkan.

Tantangan lainnya adalah **resistensi pengguna**, terutama jika sistem menggantikan metode lama yang sudah terbiasa digunakan. Beberapa pengguna mungkin tidak percaya pada sistem digital, atau merasa kurang yakin terhadap rekomendasi berbasis mesin. Oleh karena itu, edukasi dan pendampingan sangat penting untuk membangun kepercayaan dan pemahaman terhadap manfaat sistem.

Namun di balik itu semua, peluang penerapan sistem CBR justru semakin luas. Dalam **dunia pendidikan**, CBR bisa digunakan untuk menyesuaikan gaya belajar dengan profil siswa. Misalnya, sistem dapat merekomendasikan metode pembelajaran atau materi tertentu berdasarkan pengalaman siswa sebelumnya yang memiliki kesulitan dan gaya belajar yang sama.

Di **bidang pelayanan publik**, CBR bisa membantu petugas dalam menangani keluhan masyarakat berdasarkan pola kasus

yang sering terjadi. Sistem dapat menyarankan langkah penanganan yang efisien tanpa harus menunggu analisis manual dari pihak terkait. Ini bisa mempercepat respons dan meningkatkan kepuasan masyarakat.

Dalam **manajemen organisasi dan bisnis**, sistem CBR bisa digunakan untuk pengambilan keputusan strategis, seperti pemilihan vendor, manajemen risiko, atau resolusi konflik internal. Dengan melihat kasus sukses dan kegagalan masa lalu, organisasi dapat menghindari kesalahan yang sama dan mempercepat proses pengambilan keputusan.

Bidang **hukum** dan **medis** juga memiliki potensi besar dalam menerapkan CBR. Dalam hukum, sistem dapat membantu menyusun argumen atau rekomendasi berdasarkan preseden kasus. Sementara di dunia medis, sistem bisa digunakan untuk mendiagnosis penyakit langka berdasarkan kombinasi gejala yang tidak umum namun pernah terjadi sebelumnya.

Tak kalah penting, CBR juga dapat diterapkan dalam **pengelolaan pengetahuan organisasi**. Setiap organisasi memiliki siklus masalah yang berulang. Jika pengalaman masa lalu tidak terdokumentasi dengan baik, maka solusi yang sudah pernah ditemukan bisa hilang begitu saja. CBR membantu mendokumentasikan dan menyajikan kembali solusi tersebut secara cepat dan terstruktur.

Pada akhirnya, tantangan dan peluang dalam penerapan sistem CBR berjalan beriringan. Di setiap kendala terdapat pelajaran, dan di setiap peluang terdapat tanggung jawab. Dengan pendekatan yang tepat, sistem CBR bukan hanya menjadi alat bantu, tetapi juga jembatan antara masa lalu dan masa depan—yang menghubungkan pengalaman dengan inovasi, dan masalah dengan pembelajaran yang tak pernah berhenti.