

**MODUL Manajemen Pengetahuan**

**(KODE CSM410)**

**MODUL SESI 11**

**TOPIK TEKNOLOGI KM**

**DISUSUN OLEH**

**Riya Widayanti**

**UNIVERSITAS ESA UNGGUL**

**TAHUN PENULISAN**

**SUBTOPIK 1 TOPIK SESI INI**

1. **Kemampuan Akhir Yang Diharapkan**

Setelah mempelajari modul ini, diharapkan mahasiswa mampu :

* 1. Aplikasi manajemen Pengetahuan
  2. Teknologi yang mendukung Manajemen PEngetahuan
  3. Penerapan Knowledge Base (rule atau case)

1. **Uraian dan Contoh**

Dalam bab terdahulu, pernah membahas dampak organisasi dari manajemen pengetahuan. Dalam bab ini, akan dijelaskan sistem aplikasi pengetahuan, bagaimanas dikembangkan, dan menghubungkan pengalaman tentang bagaimana organisasi telah menerapkan sistem seperti itu. Sistem aplikasi pengetahuan mendukung proses di mana individu memanfaatkan pengetahuan yang dimiliki oleh orang lain tanpa secara aktif memperoleh, atau belajar tentang pengetahuan itu. Baik mekanisme dan teknologi dapat mendukung sistem aplikasi pengetahuan dengan memfasilitasi proses manajemen pengetahuan tentang rutinitas dan arahan. Sistem aplikasi pengetahuan biasanya didukung oleh teknologi cerda. Dalam bab ini, kami memperkenalkan mengenai kecerdasan buatan (AI), perspektif historisnya, hubungannya dengan pengetahuan, dan mengapa itu merupakan aspek penting dari manajemen pengetahuan. Dalam teknologi cerdas yang paling mendasari sebagian besar sistem KM, dari sistem pakar rule base, hingga case base resoning (CBR), hingga sistem informasi manajemen tradisional. Selain itu, akan dibahas berbagai jenis sistem aplikasi pengetahuan: sistem pakar, sistem help desk, dan sistem diagnosis kesalahan. Studi kasus dalam bab ini menceritakan penerapan sistem aplikasi pengetahuan. Masing-masing didasarkan pada teknologi cerdas yang berbeda dan dirancang untuk mencapai tujuan yang berbeda: memberikan saran, meningkatkan deteksi kesalahan, dan memfasilitasi penalaran kreatif.

Aplikasi pengetahuan tergantung pada arahan dan rutinitas. Mekanisme yang memfasilitasi arahan meliputi hubungan hierarkis, help desk, dan support center; sedangkan mekanisme yang memfasilitasi rutinitas meliputi kebijakan organisasi, praktik kerja, dan standar.

Teknologi yang mendukung arahan dan rutinitas meliputi sistem pakar/expert system, decision support sustem, advise system, sistem diagnosis kesalahan (atau pemecahan masalah), dan sistem help desk. Teknologi ini dapat mendukung arah, seperti dalam kasus service support lapangan yang ingin memecahkan masalah produk tertentu; atau dapat mendukung rutinitas, seperti dalam kasus perwakilan layanan pelanggan yang mungkin perlu mengidentifikasi mekanisme pengiriman produk alternatif saat mempersiapkan pengiriman pesanan. Selain itu, mekanisme dan teknologi dapat memfasilitasi aplikasi pengetahuan melalui arahan dan rutinitas baik di dalam atau di seluruh organisasi.

Untuk tinjauan singkat tentang apa itu sistem aplikasi pengetahuan dan bagaimana mereka digunakan, mari kita lihat studi kasus singkat tentang bagaimana NEC Corporation mendefinisikan kembali cara organisasi dapat menerapkan pengalaman kolektif mereka untuk menghasilkan lebih baik perangkat lunak berkualitas tingg.

Menerapkan Pengalaman Organisasi untuk Menghasilkan Perangkat Lunak Berkualitas

NEC adalah perusahaan global terkemuka yang memproduksi produk-produk canggih untuk jaringan broadband dan pasar Internet seluler. Pada tahun 1981, NEC mengakui perlunya memperluas kegiatan kontrol kualitas (QC) mereka ke domain pengembangan perangkat lunak. Untuk mencapai tujuan ini, perusahaan membentuk struktur perusahaan di seluruh perusahaan untuk membantu karyawan dalam menerapkan prinsip-prinsip kontrol kualitas perangkat lunak (SWQC). Kegiatan QC biasanya menghasilkan laporan kasus yang menguraikan analisis masalah, kemungkinan akar masalahnya, tindakan korektif yang diambil, dan hasil dari tindakan korektif.

Pada 1991, perusahaan telah mengumpulkan lebih dari 25.000 kasus seperti itu dalam upaya untuk menerapkan peningkatan produktivitas di seluruh organisasi. Awalnya laporan kasus disimpan dalam sebuah buku dan kemudian dalam database yang dapat dicari, tetapi orang-orang merasa sulit untuk mencari dan menerapkan kasus QC. NEC kemudian memutuskan untuk mengimplementasikan penasihat software quality control advisor (SQUAD), 1 berdasarkan metodologi penalaran berbasis kasus, untuk meningkatkan akses pengguna dan aplikasi kasus QC yang dilaporkan. Kasus-kasus dalam SQUAD dinominasikan melalui komite peninjau yang meninjau setiap kasus dan memilih kasus-kasus terbaik. Kasus dipilih berdasarkan kualitas analisis, signifikansi hasil, dan seberapa digeneralisasikan masalahnya.

**Insentif** yang memadai dibentuk untuk mendorong partisipasi karyawan. Awalnya sekitar 3.000 kasus diajukan setiap tahun, dan kemudian pengiriman baru menurun menjadi sekitar 1.000 kasus per tahun. Penurunan signifikan dalam tingkat pengajuan kasus baru terjadi karena sebagian besar kasus khas sudah dilaporkan dalam sistem. Pada 1994, sistem tersebut mewakili sekitar 24.000 kasus dan melayani lebih dari 150.000 pengguna.

Beberapa faktor keberhasilan yang menandai pengembangan SQUAD termasuk biaya pengembangan yang rendah, karena pengembangannya hanya membutuhkan empat orang-bulan. Selain itu, pengembangan SQUAD mendukung modifikasi tambahan, karena memungkinkan kasus untuk secara bertahap dimasukkan dalam basis data kasus. Pada 1991, diperkirakan bahwa SQUAD telah membayar kepada organisasi lebih dari 100 juta dolar per tahun.

KECERDASAN BUATAN

Pada bagian ini kita mulai dengan menggambarkan perspektif historis AI, bidang ilmu komputer yang berkaitan dengan desain dan pengembangan sistem komputer yang menunjukkan kemampuan kognitif mirip manusia. Artificial Intelligence (AI) mengacu pada memungkinkan komputer untuk melakukan tugas yang menyerupai kemampuan berpikir manusia. Sama seperti KM dan kecerdasan manusia, AI dikaitkan dengan pengetahuan. Definisi untuk AI berkisar dari: sistem yang bertindak seperti manusia, sistem yang berpikir seperti manusia, sistem yang berpikir secara rasional, hingga sistem yang bertindak secara rasional (Russell dan Norvig 2002). Sistem yang bertindak seperti manusia merujuk pada mereka yang lulus Turing Test, yang mengacu pada komputer yang lulus tes oleh seorang interogator manusia, yang tidak dapat mengetahui apakah respons datang dari seseorang atau tidak. Sistem yang berpikir seperti manusia merujuk pada program komputer yang inputnya ke perilaku keluaran cocok dengan manusia, misalnya ketika memecahkan masalah, seperti bermain catur atau melakukan diagnosis medis. Sistem yang berpikir rasional merujuk pada mereka yang mengikuti logika tertentu untuk menyelesaikan masalah karakteristik yang memungkinkan mereka untuk beroperasi secara mandiri dalam lingkungan mereka, dan bahkan beradaptasi dengan perubahan dalam menghadapi ketidakpastian.

Biasanya komputer melakukan tugas berulang yang logis dengan sangat baik, seperti perhitungan aritmatika yang kompleks dan pengambilan dan penyimpanan basis data. Satu karakteristik umum dari tugas-tugas komputer konvensional ini adalah sifat algoritmiknya, yang berarti bahwa mereka menggunakan instruksi yang dirancang secara tepat dan logis untuk menghasilkan output yang benar dan berbeda. Manusia, sebaliknya, unggul dalam memecahkan masalah menggunakan simbol-simbol yang makna tertentu dapat dilampirkan, seperti memahami makna sebuah puisi. Inteligensi buatan berurusan dengan manipulasi simbol-simbol ini. Oleh karena itu, definisi kami sendiri tentang AI (Becerra-Fernandez et al. 2004) dalam istilah yang lebih spesifik adalah:

Ilmu yang menyediakan komputer dengan kemampuan untuk mewakili dan memanipulasi simbol sehingga mereka dapat digunakan untuk memecahkan masalah yang tidak mudah dipecahkan melalui model algoritmik.

Sistem AI modern didasarkan pada pemahaman bahwa kecerdasan dan pengetahuan saling terkait erat. Pengetahuan dikaitkan dengan simbol-simbol kognitif yang kita manipulasi, sedangkan kecerdasan manusia mengacu pada kemampuan kita untuk belajar dan berkomunikasi untuk menyelesaikan masalah.

Namun, ketika kita menilai kinerja siswa di kelas atau memutuskan siapa yang akan dipekerjakan, kita biasanya fokus pada seberapa banyak yang mereka ketahui, bukan kecerdasan mereka. Orang dilahirkan dengan tingkat kecerdasan tertentu, yang mereka gunakan untuk belajar dan dengan demikian memperoleh pengetahuan baru. Beberapa sistem AI (juga dikenal sebagai sistem berbasis pengetahuan atau sistem aplikasi pengetahuan) mencoba untuk meniru kemampuan pemecahan masalah dari pemecah masalah yang terampil dalam domain tertentu. Sistem cerdas menawarkan kepada kita teknologi untuk mengelola pengetahuan — yaitu, untuk menerapkan, menangkap, berbagi, dan menemukannya.

Gagasan menciptakan komputer yang menyerupai kemampuan kecerdasan manusia dapat ditelusuri hingga 1950-an, ketika para ilmuwan meramalkan pengembangan mesin tersebut dalam satu dekade. Meskipun para ilmuwan yang sama mungkin telah meremehkan kerumitan pikiran manusia, penelitian AI membuat langkah signifikan. Istilah "kecerdasan artifisial" diciptakan oleh John McCarthy selama lokakarya yang ia selenggarakan di Dartmouth College pada tahun 1956, di mana ia mengumpulkan empat perintis lapangan: John McCarthy, Marvin Minsky, Allan Newell, dan Herbert Simon.

RULE BASE SYSTM

Secara tradisional, pengembangan sistem berbasis pengetahuan telah didasarkan pada penggunaan aturan atau model untuk mewakili pengetahuan domain. Pengembangan sistem tersebut membutuhkan kolaborasi ahli materi pelajaran dengan insinyur pengetahuan, yang terakhir bertanggung jawab untuk elisitasi dan representasi pengetahuan pakar.

Proses pengembangan sistem aplikasi pengetahuan membutuhkan kemunculan pengetahuan dari pakar dan mewakilinya bentuk yang dapat digunakan oleh komputer. Proses ini disebut rekayasa pengetahuan. Insinyur pengetahuan biasanya membangun sistem aplikasi pengetahuan dengan terlebih dahulu mewawancarai pakar domain dan mewakili pengetahuan tersebut lebih umum dalam serangkaian heuristik, atau aturan praktis. Para ahli mengembangkan aturan praktis ini selama bertahun-tahun pengalaman praktis dalam memecahkan masalah. Agar komputer memahami aturan-aturan ini, kami mewakilinya sebagai aturan produksi atau pernyataan IF-THEN. Sebagai contoh: JIKA jumlah karyawan kurang dari 500, KEMUDIAN perusahaan adalah bisnis kecil adalah salah satu aturan yang diperiksa oleh SOS Advisor untuk memastikan perusahaan tersebut memenuhi syarat untuk program peminjamang(contoh). Aturan adalah paradigma representasi pengetahuan yang paling umum digunakan, mungkin karena penerapannya yang intuitif. Bagian IF adalah kondisi (juga premis atau anteseden), yang menguji nilai kebenaran dari serangkaian asersi. Jika pernyataan itu benar, maka THEN bagian dari aturan (juga tindakan, kesimpulan, atau konsekuensi) juga disimpulkan sebagai fakta.

CASE BASE REASONING SYSTEM

Meskipun pendekatan aturan untuk representasi pengetahuan telah menghasilkan banyak contoh sistem aplikasi pengetahuan yang sukses, banyak dari sistem ini semakin didasarkan pada implementasi metodologi penalaran berbasis kasus.

Case-based reasoning (CBR) adalah teknik kecerdasan buatan yang dirancang untuk meniru pemecahan masalah manusia. CBR didasarkan pada model memori dinamis Schank (1982). Tujuannya adalah untuk meniru cara manusia memecahkan masalah. Ketika dihadapkan dengan masalah baru, manusia mencari ingatan mereka untuk masalah masa lalu yang menyerupai masalah saat ini dan mengadaptasi solusi sebelumnya untuk "menyesuaikan" masalah saat ini. CBR adalah metode penalaran analogis yang memanfaatkan kasus atau pengalaman lama dalam upaya menyelesaikan masalah, mengkritik solusi, menjelaskan situasi anomali, atau menafsirkan situasi (Aamodt dan Plaza 1994; Kolodner 1991, 1993; Leake 1996; Watson 2003). Sistem aplikasi pengetahuan berbasis kasus yang khas akan terdiri dari proses berikut:

1. Cari perpustakaan perpustakaan untuk kasus serupa. Ini hanya menggunakan mesin pencari yang memeriksa hanya kasus yang sesuai dan bukan seluruh perpustakaan kasus, karena mungkin cukup besar.
2. Pilih dan ambil kasing yang paling mirip. Masalah baru diselesaikan dengan mengambil kembali kasus yang sebelumnya dialami. Ini menyiratkan memiliki sarana untuk membandingkan setiap kasus yang diperiksa dengan masalah saat ini, mengukur kesamaan mereka, dan entah bagaimana memeringkat mereka dalam mengurangi urutan kesamaan.
3. Adaptasi solusi untuk kasus yang paling mirip. Jika masalah saat ini dan kasus yang paling mirip tidak cukup mirip, maka solusinya mungkin harus disesuaikan agar sesuai dengan kebutuhan masalah saat ini. Masalah baru akan diselesaikan dengan bantuan solusi lama yang telah disesuaikan dengan masalah baru.
4. Terapkan solusi yang dihasilkan dan dapatkan umpan balik. Setelah solusi atau klasifikasi dihasilkan oleh sistem, itu harus diterapkan pada masalah. Efeknya pada masalah dimasukkan kembali ke sistem CBR untuk klasifikasi solusinya (sebagai keberhasilan atau kegagalan).
5. Tambahkan masalah yang baru diselesaikan ke pustaka kasus. Pengalaman baru kemungkinan akan berguna dalam pemecahan masalah di masa depan. Langkah ini membutuhkan pengidentifikasian jika case baru layak ditambahkan ke perpustakaan dan menempatkannya di lokasi yang sesuai di perpustakaan case.

**Developing Knowledge Management System**

Di sini kami menjelaskan bagaimana membangun sistem aplikasi pengetahuan. Kami menggunakan banyak contoh dan kotak untuk meningkatkan pengalaman belajar. Bagian selanjutnya membahas berbagai jenis sistem aplikasi pengetahuan, dan contoh spesifik disajikan pada bagian selanjutnya.

Implementasi yang efektif dari sistem aplikasi pengetahuan membutuhkan metodologi yang dipikirkan secara matang. Case-Method Cycle (Kitano 1993; Kitano dan Shimazu 1996) adalah metodologi yang menggambarkan pendekatan berulang untuk secara efektif untuk mengembangkan CBR dan sistem aplikasi pengetahuan secara umum. Siklus-Metode Kasus menjelaskan enam proses berikut:

1. Proses pengembangan sistem — Proses ini didasarkan pada pendekatan rekayasa perangkat lunak standar, dan tujuannya adalah untuk mengembangkan sistem aplikasi pengetahuan yang akan menyimpan kasus baru dan mengambil kasus yang relevan.
2. Proses pengembangan perpustakaan - Tujuan pengembangan proses ini dan mengelola perpustakaan kasus skala besar yang akan cukup mendukung domain yang dimaksud.
3. Proses operasi sistem — Proses ini didasarkan pada rekayasa perangkat lunak standar dan prosedur manajemen basis data relasional. Tujuannya adalah untuk menentukan instalasi, penyebaran, dan dukungan pengguna dari sistem aplikasi pengetahuan.
4. Proses penambangan basis data — Proses ini menggunakan teknik aturan-pendugaan dan analisis statistik untuk menganalisis perpustakaan kasus. Langkah ini dapat membantu menyimpulkan hubungan baru antara data, yang dapat diartikulasikan untuk meningkatkan sistem aplikasi pengetahuan.
5. Proses manajemen — proses ini menjelaskan bagaimana proyek harus dibentuk dan dukungan organisasi apa yang akan diberikan kepada proyek.
6. Proses transfer pengetahuan — Proses ini menjelaskan sistem insentif yang akan diterapkan untuk mendorong penerimaan dan dukungan pengguna sistem aplikasi pengetahuan. Langkah ini akan memastikan bahwa pengguna akan merasa perlu menambah pustaka kasus dengan kasus baru.

Dalam hal mengembangkan perpustakaan kasus (langkah 2 di atas), prosesnya juga dapat dijelaskan dalam istilah subproses berikut (Kitano dan Shimazu 1996):

1. Pengumpulan Kasus — Proses ini mencakup pengumpulan kasus benih, yang memberikan tampilan awal aplikasi. Misalnya, untuk sistem SQUAD yang dijelaskan dalam Kotak 6.1, pengembang memulai dengan 100 kasing kasus. Kaset benih ini digunakan untuk menentukan format untuk pengumpulan kasing masa depan dan untuk desain struktur basis data. Kasing kasus biasanya tidak mengikuti struktur yang telah ditentukan, sedangkan koleksi kasing berikutnya akan mengikuti format yang ditentukan. Jumlah kasus benih dapat bervariasi sesuai dengan aplikasi, seperti yang akan kita lihat dalam bab ini, dan bahkan dapat dihasilkan secara artifisial dengan membuat permutasi dari kasus yang tersedia, seperti yang dibahas dalam studi kasus nanti dalam bab ini.
2. Ekstraksi Nilai-Nilai dan Formasi Hierarki — Langkah ini sangat penting untuk mengindeks dan mengatur perpustakaan kasus. Tujuan dari fase ini adalah untuk menggali atribut yang mendefinisikan representasi kasus dan pengindeksan. Fase ini akan berupaya membuat daftar atribut yang mendefinisikan setiap kasus, daftar nilai untuk setiap atribut, dan kemungkinan pengelompokan atribut tersebut. Selain itu, hubungan antar atribut juga harus didefinisikan. Setelah hierarki ditentukan, kepentingan relatif dari setiap atribut ditentukan. Keputusan ini biasanya mencerminkan domain implementasi.
3. Umpan Balik — Fase ini akan memberikan umpan balik yang diperlukan bagi mereka yang memasok kasus ke sistem CBR, sehingga kualitas kasus dapat ditingkatkan.

Ada beberapa batasan praktis untuk pengembangan sistem aplikasi pengetahuan. Ini terkait dengan fakta bahwa sebagian besar sistem ini dikembangkan untuk melayani masalah domain khusus-tugas dan biasanya tidak terintegrasi dengan sistem perusahaan organisasi. Batasan lain juga ada — misalnya, untuk sistem aplikasi pengetahuan yang didasarkan pada teknologi CBR, batasan berikut ini berlaku (Kitano dan Shimazu 1996):

1. Keamanan: Kasus dapat mencakup informasi sensitif. Sistem aplikasi pengetahuan harus mempertimbangkan penggabungan langkah-langkah keamanan, termasuk kontrol akses sesuai dengan peran pengguna. Jika sistem aplikasi pengetahuan tidak memasukkan langkah-langkah keamanan, sistem mungkin tidak menyadari nilai maksimalnya.
2. Skalabilitas: Sistem aplikasi pengetahuan harus mewakili jumlah kasus yang cukup besar sehingga sebagian besar pengalaman baru terwakili dalam sistem berbasis kasus. Ini berarti sistem aplikasi pengetahuan harus mencapai saturasi sebelum penempatannya. Mencapai kejenuhan sistem berarti bahwa kebanyakan kasus tipikal sudah pernah dilaporkan dalam sistem. Jumlah kasus yang diperlukan untuk mencapai perubahan titik jenuh menurut domain. Untuk SQUAD, yang dibahas sebelumnya, untuk mencapai titik ini diperlukan pencantuman sekitar 3.000 kasus setiap tahun, jumlah yang kemudian dikurangi menjadi 1.000 per tahun. Semakin kompleks domain, semakin tinggi pentingnya menjaga pertumbuhan basis kasus. Jelas, pertumbuhan berkelanjutan dari perpustakaan kasus juga akan memerlukan penggunaan skema pengindeksan kompleks, yang dapat mengakibatkan penurunan stabilitas sistem.
3. Kecepatan: Ketika ukuran kasus perpustakaan tumbuh menjadi representasi yang lebih komprehensif dari lingkungan nyata, biaya komputasi dan pencarian juga akan meningkat. Oleh karena itu, pengembang sistem aplikasi pengetahuan harus mempertimbangkan penggunaan skema pengindeksan kompleks yang akan menjamin waktu pengambilan kasus dan tingkat kinerja yang dapat diterima.

Selain itu, sistem aplikasi pengetahuan mungkin tidak dapat menyelesaikan semua masalah yang mereka hadapi. Secara khusus, mendiagnosis masalah mungkin semakin sulit di lingkungan yang kompleks, sebagaimana dibuktikan oleh tragedi Space Shuttle Columbia. Mendiagnosis segala sesuatu yang salah atau mungkin salah dalam lingkungan seperti itu mungkin tidak dimungkinkan dengan sistem seperti yang sudah dijelaskan. Teknologi baru perlu dikembangkan untuk mencegah insiden di lingkungan teknik yang kompleks.

Beberapa sistem berbasis aturan dapat menderita dari keterbatasan lain, yaitu kurangnya skalabilitas. Teknologi lain menawarkan serangkaian batasan yang berbeda. Tetapi pada dasarnya, manfaat yang dibawa oleh penerapan sistem aplikasi pengetahuan ke organisasi melebihi keterbatasan mereka, dan mereka akan terus memberikan keunggulan kompetitif bagi organisasi yang terus menerapkannya.

Penelitian menunjukkan bahwa help desk TI bergerak cepat dari sejumlah besar pengikat tiga cincin ke beberapa jenis sistem berbasis pengetahuan. Gerakan ini mengkonfirmasi sebagian tesis studi eksplorasi ini bahwa basis pengetahuan menjadi bagian penting dari help desk TI. Pertanyaan penting yang masih harus diselidiki adalah manfaat relatif CBR dan RBR untuk tugas ini.

Sistem Pakar Didefinisikan

Jackson (1999) menyatakan bahwa sistem pakar adalah program komputer yang diturunkan dari Artificial Intelligence (AI). Tujuan AI adalah untuk memahami kecerdasan dengan membangun program komputer, atau cangkang, yang menunjukkan perilaku cerdas. AI lebih lanjut memusatkan perhatian pada konsep dan metode inferensi simbolik, atau penalaran, dan bagaimana pengetahuan itu akan digunakan untuk membuat kesimpulan itu akan terwakili di dalam mesin.

Jackson (1999) lebih lanjut menyatakan bahwa sistem pakar dapat sepenuhnya memenuhi kesenjangan pengetahuan dalam suatu domain atau dapat bertindak sebagai asisten untuk membantu ahli domain dalam memecahkan masalah yang kompleks. Biasanya, sistem pakar digunakan dalam beragam bidang pengetahuan, seperti kedokteran internal, kimia organik, dan aplikasi bisnis seperti help desk TI. Tugas sistem pakar mencakup interpretasi data, diagnostik seperti kerusakan mesin, analisis senyawa kimia kompleks, konfigurasi sistem komputer, dan untuk digunakan dalam robotika.

Program komputer konvensional dapat ditulis untuk memasok beberapa pengetahuan domain yang diperlukan yang diperlukan oleh seorang individu. Namun, ada perbedaan besar dalam cara sistem pakar dapat dibedakan dari program aplikasi konvensional ini.

Pertama, sistem pakar "mensimulasikan penalaran manusia" tentang domain masalah yang diberikan tetapi tidak mensimulasikan domain itu sendiri. Kedua, sistem pakar melakukan "penalaran atas representasi" pengetahuan manusia sambil mempertahankan kemampuan untuk melakukan perhitungan numerik dan / atau pengambilan data. Ketiga, program konvensional memecahkan masalah menggunakan metodologi algoritme yang ketat sementara sistem pakar menggunakan "metode heuristik atau perkiraan" yang tidak menjamin bahwa jawabannya akan benar-benar benar dan akan berhasil. Ini "heuristik atau metode perkiraan" lebih dikenal sebagai aturan praktis (Jackson, 1999).

Jackson (1999) melanjutkan bahwa ada perbedaan antara jenis lain dari program kecerdasan buatan dan sistem pakar. Pertama, sistem pakar menangani masalah yang membutuhkan banyak keahlian manusia. Program AI lainnya terutama berurusan dengan masalah matematika abstrak dan, sebagian besar, dianggap kendaraan penelitian. Kedua, sistem pakar, karena berusaha memecahkan masalah yang berhubungan dengan keahlian manusia, harus menunjukkan tingkat kinerja yang tinggi dalam hal kecepatan dan keandalan untuk mempertahankan kegunaannya. Ini karena masalah-masalah yang berhubungan dengan keahlian manusia, misalnya menyelesaikan masalah yang berkaitan dengan kinerja komputer, membutuhkan solusi dalam jangka waktu singkat karena volume masalah yang dihadapi dalam satu hari. Program AI hanya itu, program, yang digunakan sebagai kendaraan penelitian. Akhirnya, Jackson menyarankan bahwa sistem pakar harus "mampu menjelaskan dan membenarkan solusi atau rekomendasi" untuk memberikan kepercayaan pengguna bahwa jawaban yang diambil dari sistem pakar, pada kenyataannya, benar.

Englemore dan Feigenbaum (1993) menyatakan bahwa AI berkaitan dengan konsep dan metode inferensi simbolik, atau penalaran, oleh komputer, dan bagaimana pengetahuan ini akan digunakan untuk mewakili kesimpulan ini dalam sistem komputer itu sendiri. Englemore dan Feigenbaum melanjutkan bahwa setiap sistem pakar terdiri dari dua bagian utama: basis pengetahuan; dan mesin penalaran, atau inferensi.

Luger dan Stubblefield (2002) menyatakan bahwa basis pengetahuan sistem pakar berisi pengetahuan faktual dan heuristik. Pengetahuan faktual adalah pengetahuan tentang domain tugas yang dibagikan secara luas dan, biasanya ditemukan dalam buku dan jurnal. Selanjutnya, pengetahuan faktual ini umumnya disepakati oleh para ahli dalam domain tertentu.

Pengetahuan heuristik adalah pengetahuan kinerja yang kurang ketat, lebih pengalaman, dan lebih menghakimi. Pengetahuan yang dimiliki manusia itulah yang mendasari kemampuan kita untuk merumuskan “tebakan yang baik”. Formalisasi dan organisasi pengetahuan dikenal sebagai representasi pengetahuan.

Studi eksplorasi dalam disertasi ini membahas dua paradigma terpenting yang digunakan dalam sistem pakar. Paradigma ini adalah alasan berdasarkan hukum dan kasus.

**Penggunaan saat ini dan Kemajuan dalam Sistem Pakar dan Akuisisi Pengetahuan menggunakan Rule-based Reasoning (RBR**)

Representasi pengetahuan berbasis aturan adalah salah satu jenis yang paling banyak digunakan (Englemore dan Feigenbaum, 1993). Tapi apa sebenarnya Penalaran berbasis Rule? Crossman et al. (1995) menyatakan dengan memecah ini menjadi bagian-bagian individualnya, aturan adalah representasi pengetahuan tentang mana "pola informasi yang digunakan para ahli untuk membuat keputusan dan apa keputusan yang mengikuti." Penalaran berbasis aturan menawarkan seperangkat aturan yang berantai ke diberikan kesimpulan. Cara paling populer untuk mewakili pengetahuan ini adalah dengan menggunakan aturan "jika-maka". Crossman et al. nyatakan bahwa aturan ini dapat direpresentasikan dalam hubungan logis “p → q”. Dalam hubungan ini, "p" mewakili suatu kondisi atau serangkaian kondisi dan "q" mewakili suatu kesimpulan atau serangkaian kesimpulan. Himpunan relasi / kesimpulan ini, menurut Ignizio (1991), himpunan paling umum digunakan oleh sistem pakar, dan terutama dalam sistem pakar berbasis aturan. Strategi inferensi ini lebih dikenal sebagai "modus ponens" yang berarti bahwa jika A menyimpulkan B dan A benar, maka B benar.

Crossman et al. (1995) lebih lanjut menyatakan bahwa banyak algoritma yang berbeda telah dikembangkan untuk mendukung premis dasar penalaran berbasis aturan. Ada berbagai perbedaan dalam pendekatan ini dan semuanya berada dalam domain rekayasa pengetahuan.

Salah satu contoh yang diberikan oleh Crossman et al. adalah bahwa "aturan rantai maju memfasilitasi sintesis pemrograman, sedangkan aturan rantai belakang lebih cocok untuk analisis atau pencarian."

Jika jenis penalaran yang terlibat dalam domain yang menarik melibatkan penggunaan diagram alir atau pohon, maka penggunaan aturan adalah cara terbaik untuk melanjutkan. Namun, peraturan ini tidak; mewakili fakta atau data itu sendiri; mereka lebih mewakili alasan tentang fakta atau data. Sistem berbasis aturan menggunakan mesin inferensi yang merupakan algoritme yang mengatur apa yang dapat dilakukan aturan, kapan aturan itu diaktifkan atau dipicu, dan prioritas apa yang diberikan kepada masing-masing untuk dieksekusi. Aturan dalam sistem berbasis aturan juga dapat menghibur bentuk tertentu dari penalaran tidak pasti, misalnya, menambah atau mengurangi tingkat kepercayaan saat mengevaluasi hipotesis atau menyediakan mekanisme alternatif untuk menangani jalur penalaran lain (Crossman et al., 1995).

Perkakas berbasis aturan dan sistem pakar lainnya lebih dikenal sebagai cangkang. Keuntungan terbesar yang ditawarkan oleh sistem berbasis aturan adalah mereka memungkinkan pengguna untuk melihat aturan dalam format bahasa yang hampir alami dan memberikan penjelasan mengapa penjelasan dibuat.

Luger dan Stubblefield (2002) menyatakan bahwa upaya pertama untuk membangun sistem pakar menggunakan paradigma berbasis aturan, tidak mungkin sangat berhasil. Alasan utama untuk ini adalah karena ahli domain merasa sangat sulit untuk mengekspresikan pengetahuan diam-diam dalam istilah yang dapat digunakan untuk memecahkan masalah spesifik.

Akhirnya, Luger dan Stubblefield (2002) menyimpulkan bahwa skema representasi pengetahuan yang paling banyak digunakan untuk sistem pakar adalah berbasis aturan. Biasanya, aturan itu sendiri tidak akan memiliki kesimpulan tertentu tetapi akan ada beberapa tingkat kepastian bahwa kesimpulan itu akan berlaku jika kondisinya berlaku. Ada teknik statistik yang digunakan untuk menentukan kepastian ini. Sistem berbasis aturan, apakah mereka memiliki kepastian atau tidak, biasanya mudah dimodifikasi. Kepastian ini membuatnya relatif sederhana untuk memberikan jejak bermanfaat dari penalaran sistem. Jejak-jejak ini dapat digunakan dalam memberikan penjelasan tentang apa yang dilakukannya.

Perlu dicatat bahwa alasan berbasis aturan digunakan dalam lingkungan help desk terutama sebagai peningkatan teknologi berbasis kasus (Luger dan Stubblefield, 2002).

**Penggunaan saat ini dan Kemajuan dalam Sistem Pakar dan Akuisisi Pengetahuan menggunakan Case-based Reasoning (CBR)**

Case-based reasoning (CBR) adalah paradigma Artificial Intelligence (AI) untuk pemecahan masalah dan penggunaan kembali pengetahuan yang menggunakan contoh serupa sebelumnya untuk memecahkan masalah saat ini. Selanjutnya, CBR menarik kemampuannya untuk mencari ingatannya untuk solusi dan memperoleh yang baru tanpa harus memahami prinsip-prinsip yang mendasari domainnya (Kolodner, 1993).

Namun, untuk menjelaskan cara kerja CBR, kita harus terlebih dahulu memahami makna kasus. Watson (2002, hlm. 27) mendefinisikan case sebagai:

Kasus adalah catatan pengalaman yang mengandung pengetahuan, yang bisa eksplisit dan diam-diam. Misalnya, mereka dapat menjadi kasus dalam pengertian hukum; mereka dapat berupa riwayat kasus pasien dalam pengertian medis, rincian pinjaman bank, atau deskripsi situasi pemecahan masalah peralatan.

Watson melanjutkan dengan menggambarkan apa yang terdiri dari kasus hukum, riwayat kasus medis, pinjaman bank dan catatan pemecahan masalah. Pertama, ia menyatakan bahwa deskripsi terdiri dari "masalah hukum, gejala pasien, detail pinjaman, dan masalah peralatan". Dia menyimpulkan dengan menyatakan bahwa hasil atau solusi dari masing-masing uraian ini terdiri dari "putusan atau putusan, perlakuan, hasil pinjaman, dan perbaikan teknis".

Kolodner (1993) menyatakan bahwa suatu kasus lebih lanjut dapat digambarkan sebagai laporan dari suatu peristiwa, cerita, atau beberapa catatan yang biasanya terdiri dari masalah yang menggambarkan keadaan dunia ketika kasus itu terjadi dan solusi yang menyatakan solusi yang berasal dari masalah itu. Ini berarti bahwa CBR mendapatkan solusi dari kasus-kasus sebelumnya saja dan memperoleh kasus-kasus baru untuk meningkatkan dan mengembangkan kemampuan pengambilan keputusannya.

Lebih jauh, representasi suatu kasus memiliki berbagai bentuk, seperti contoh atau bahkan cerita, selama itu dapat dikenali oleh seorang pemikir dalam domain tertentu. Secara semantik, sebuah kasus mewakili sepotong pengetahuan tertentu dan konteksnya, di mana kasus tersebut akan diambil untuk membangun solusi untuk masalah baru. Ini berarti bahwa kita dapat melihat penalaran berbasis kasus sebagai proses mengingat serangkaian kasus sebelumnya dan membuat keputusan berdasarkan perbandingan antara mereka dan situasi baru.

Kolodner (1993) melanjutkan dengan membahas kelebihan dan kekurangan menggunakan CBR dalam pengembangan pengetahuan. Kolodner menegaskan bahwa CBR memungkinkan pemikir untuk mengajukan solusi atas masalah dengan cepat, menghindari waktu yang diperlukan untuk memperoleh jawaban-jawaban itu dari awal. Sebagai contoh, seorang dokter yang mengingat diagnosis lama atau perawatan mengalami manfaat ini. Pemikir berbasis kasus, seperti halnya pemikir lainnya, harus mengevaluasi solusi yang diusulkan; memulai upaya memecahkan masalah karena dapat menghasilkan proposal dengan mudah. Hal ini tentu terungkap selama evaluasi aplikasi CBR yang disebut CASEY (aplikasi CBR pertama Kolodner), yang menunjukkan percepatan dua urutan besarnya ketika masalah telah terlihat di masa lalu.

CBR memungkinkan pemikir untuk mengajukan solusi dalam domain yang tidak sepenuhnya dipahami oleh pemikir. Banyak domain yang mustahil untuk dipahami sepenuhnya, sering kali karena banyak yang bergantung pada perilaku manusia yang tidak terduga. CBR memungkinkan asumsi dan prediksi dibuat berdasarkan apa yang berhasil di masa lalu tanpa memiliki pemahaman yang lengkap tentang masalah atau masalah.

CBR memberi pemikir sarana untuk mengevaluasi solusi ketika tidak ada metode algoritmik yang tersedia untuk evaluasi. Menggunakan kasus, dalam hal ini, sangat membantu ketika ada banyak hal yang tidak diketahui, membuat jenis evaluasi lain menjadi tidak mungkin atau paling tidak sulit. Solusi dievaluasi dalam konteks situasi serupa sebelumnya. Sekali lagi, pemikir melakukan evaluasinya berdasarkan apa yang berhasil di masa lalu (Kolodner, 1993).

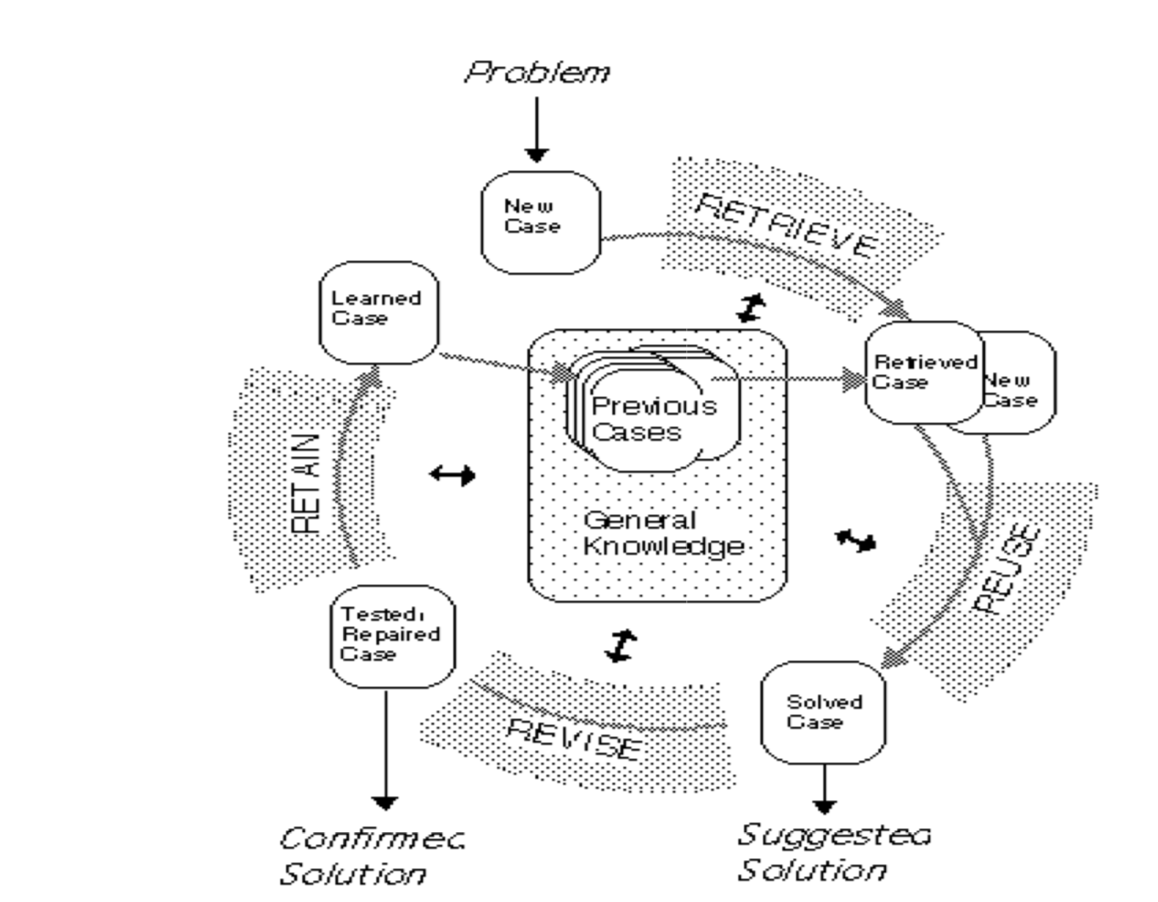
Metodologi penyelesaian masalah CBR sama dengan cara manusia menyelesaikan masalah ketika ada situasi ketika seseorang menghadapi situasi atau masalah baru, orang itu akan sering merujuk pada pengalaman masa lalu dari masalah yang sama.

(Pal dan Shiu, 2004). Pal dan Shiu melanjutkan bahwa konsep CBR sangat menarik karena fakta bahwa itu sangat mirip dengan perilaku pemecahan masalah manusia dan dengan demikian akan menghilangkan tugas analisis mendalam dari domain masalah di mana sejarah tersedia. Akhirnya, Pal dan Shiu menyimpulkan bahwa penggunaan metode ini mengarah pada keuntungan bahwa CBR dapat didasarkan pada "pengetahuan dangkal" dan tidak memerlukan upaya rekayasa pengetahuan yang diperlukan oleh sistem berbasis aturan.

Bergmann et al. (2003, p. 16) menyatakan bahwa "Dibandingkan dengan sistem pakar, sistem pendukung keputusan berbasis kasus tidak bergantung pada aturan yang dipasok oleh spesialis". Bergmann et al. percaya bahwa CBR lebih merupakan "pendekatan alami" di mana teknisi help-desk (atau spesialis lainnya) tidak pernah harus menyediakan aturan diagnostik atau untuk menentukan spesifikasi formal dari setiap proses keputusan yang digunakan untuk menentukan solusi untuk suatu masalah. Sistem pendukung keputusan CBR memiliki kemampuan untuk memperoleh dan mempertahankan pengetahuan karena sistem memiliki kemampuan untuk mempelajari kasus-kasus baru.

Pal dan Shiu (2004) menyatakan bahwa proses CBR dapat diabstraksi sebagai siklus yang terdiri dari empat langkah dasar (Gambar 1, Siklus CBR): (1) Pengambilan Kasus untuk menemukan kasus yang paling mirip yang akan mengatasi masalah baru , (2) Case Reuse untuk memanfaatkan case yang diambil untuk menyelesaikan masalah, (3) Case Revisi atau mengadaptasi untuk memodifikasi case yang diambil dengan harapan akan sesuai dengan masalah baru, dan (4) Retensi Kasus untuk mempertahankan kasus yang direvisi sebagai kasus baru dalam basis kasus setelah dikonfirmasi atau divalidasi.

Penelitian lain telah menyimpulkan bahwa empat langkah dasar dalam siklus CBR sebenarnya harus menjadi proses enam langkah dengan menambahkan fase Restore and Review.



Gambar Siklus CBR Cycle

Göker dan Roth-Berghofer (1999) percaya bahwa langkah-langkah dalam siklus CBR terkandung dalam dua siklus umum; Siklus Aplikasi dan Siklus Pemeliharaan. Siklus Aplikasi, yang berisi langkah Ambil, Penggunaan Kembali, dan Revisi, dilakukan setiap kali pengguna atau teknisi help-desk menyelesaikan masalah dengan sistem dukungan help-desk berbasis kasus. Jika solusi yang dihasilkan selama siklus Reuse tidak benar dan tidak dapat diperbaiki, teknisi help-desk harus menghasilkan solusi baru. Solusi baru ini digunakan selama fase ReCyle. Semua solusi baru yang dihasilkan dengan cara ini disimpan dalam buffer dan tersedia untuk teknisi help desk sebagai kasus yang belum dikonfirmasi. Kasing yang belum dikonfirmasi ini kemudian dikirim ke Siklus Pemeliharaan untuk diproses dan dimasukkan ke pangkalan kasus. Siklus Perawatan, yang berisi langkah Retain dan Perbaiki, jarang dilakukan. Biasanya,

fase pemeliharaan dilakukan pada interval tertentu untuk memperbarui basis kasus dengan case yang belum dikonfirmasi yang terkandung dalam buffer fase ReCycle. Ada dua langkah tambahan yang telah ditambahkan Göker dan Roth-Berghofer ke siklus CBR empat fase yang diterima secara umum. Ini adalah langkah-langkah ReCycle dan Perbaiki. Langkah ReCycle digunakan sebagai persimpangan antara Aplikasi dan Pemeliharaan Siklus dan berisi kasus yang tidak dikonfirmasi dikirim oleh Siklus Aplikasi. Kasing yang belum dikonfirmasi yang diambil dari buffer ReCycle ditempatkan pada langkah Perbaiki di mana kasus tersebut diperbaiki dan ditulis berbasis kasus.

Misi utama dari langkah Perbaiki adalah untuk memastikan bahwa kasus dasar akurat. Ada lima pemeriksaan yang dinyatakan oleh Göker dan Roth-Berghofer (1999, p. 144) yang harus dilakukan sebelum kasing dapat ditambahkan ke pangkalan kasing. Ini adalah:

1. “Apakah itu merupakan alternatif yang layak yang belum ada dalam basis kasus
2. Apakah itu termasuk atau dapat dimasukkan oleh kasus yang ada,
3. Apakah dapat dikombinasikan dengan kasus lain untuk membentuk yang baru,
4. Apakah kasus baru akan menyebabkan ketidakkonsistenan, dan
5. Apakah ada kasing yang lebih baru sudah tersedia di kasing. ”

Roth-Berghofer dan Iglezakis (2001) juga percaya bahwa siklus CBR enam langkah adalah metode yang benar dan bahwa dua fase, Pemeliharaan dan Aplikasi, paling baik menggambarkan proses CBR yang benar untuk mengambil solusi dan memastikan bahwa mereka akurat dan kemudian menyimpannya. dalam basis kasus. Gambar 4 menunjukkan bagaimana fase Perawatan dan Aplikasi berinteraksi.

1. **Daftar Pustaka**
2. https://core.ac.uk/download/pdf/51097894.pdf