**BAB II**

**KAJIAN PUSTAKA**

**2.1 Kajian Penelitian terdahulu**

Pada penelitian ini mengacu kepada beberapa penelitian yang sudah dilakukan sebelumnya, yang pertama yaitu penelitian tentang pemilihan menu makanan bagi anak(Kumaladewi, Rizky Ria Kusumadewi, 2015). Pada penelitian ini menggunakan metode CBR yaitu mengadopsi pengetahuan dari manusia dimana pada penelitian ini dengan pakar gizi, sehingga komputer dapat menyelesaikan masalah seperti yang biasa dilakukan oleh pakar gizi. Metode penalaran berbasis kasus (*Case Based* *Reasoning*) digunakan karena telah ada sejumlah kasus dan solusi-solusi yang pernah dicapai sebelumnya yang disimpan dalam basis pengetahuan, dari basis pengetahuan yang ada kemudian akan diturunkan solusi untuk keadaan yang terjadi sekarang. Pada pemilihan menu makanan ini membantu orang tua untuk memenuhi kebutuhan gizi anaknya secara optimal dengan bahan makanan yang dimiliki, dimana data yang dimasukkan kedalam sistem adalah data anak, kemudian bahan makanan yang dimiliki, selanjutnya sistem akan menyeleksi dengan basis pengetahuan yang telah tersimpan di database, dalam basis pengetahuan ini terdapat beberapa aturan yang disesuaikan dengan bahan yang tersedia dengan kebutuhan gizi anak. Setelah orang tua selesai memasukkan data maka akan ditampilkan menu makanan apa yang cocok untuk dipilih. Pada penelitian ini mampu menentukan menu makanan bagi anak berdasarkan bahan makanan yang dimiliki dan sesuai kebutuhan gizi.

Pada penelitian lainnya yaitu penelitian tentang sistem pendukung keputusan untuk penanganan dini pada kecelakaan mobil dengan metode herbal (Wijaya & Sedana, 2015). Pada penelitian menggunakan metode CBR sama seperti penelitian yang dilakukan sebelumnya hanya saja pada penelitian ini diterapkan pada kasus kecelakaan dalam rumah tangga, pada penelitian ini data yang dimasukkan adalah nama kecelakaan kemudian user memilih gejala yang dialami lalu kemudian sistem akan memprosesnya dengan menggunakan metode CBR yaitu mencari solusi kasus yang sebleumnya pernah terjadi di database, sehingga user akan memperoleh solusinya berupa solusi herbal.

Selanjutnya yaitu penelitian tentang diagnosa penyakit kejiwaan(Ibrahim & Winiarti, 2014), pada penelitian ini menggunakan metode CBR dengan kedekatan *nearest neighbour* yaitu menghitung tingkat kemiripan tertinggi dengan kemiripan solusi kasus yang tersimpan dalam basis pengetahuan. Data yang dibutuhkan dalam penelitian ini adalah data penyakit, data gejala, dan data solusi yang bersumber dari data rekam medis di Klinik Praktek Dokter Mitra Keluarga yang terkait mengenai penyakit kejiwaan khususnya penyakit kejiwaan *skizofrenia*. Dalam penelitian ini mampu memberikan diagnosa awal terhadap penyakit *skizofrenia* sebelum ditangani oleh dokter, dalam penelitian ini masih perlu ditambahkan beberapa kriteria yang dibutuhkan untuk keakuratan sistem dan memperbanyak jenis penyakit kejiwaan serta pencarian basis kasus untuk meningkatkan kemampuan sistem.

Penelitian yanng lainnya yaitu tentang penentuan kesesuaian lahan pertanian untuk budidaya tanaman buah-buahan (Habibullah & Winiarti, 2014), Penelitian ini menggunakan metode *similarity* yaitu sebuah metode yang akan mencari solusi kesimpulan berdasarkan kemiripan objek yang ada kemudian dipilih solusi dengan nilai *similarity* terbesar. Penentuan kesesuaian lahan ini dapat membantu petani untuk menentukan lahan yang sesuai untuk budidaya tanaman buahnya, Data yang diperlukan dalam penelitian ini yaitu data tanaman buah, fakta lahan, dan cara penanaman. Data tanaman buah diperlukan karena merupakan inti dari pengetahuan yang akan digunakan sebagai tujuan diagnosis. Data fakta lahan merupakan data yang dimasukan oleh user. Data cara penanaman merupakan data yang berisi cara penanaman tanaman buah. Dalam sistem ini petani dapat memasukkan kriteria-kriteria lahan yang sesuai kemudian sistem akan memberikan solusi jenis lahan yang sesuai dengan fakta lahan.

Penelitian berikutnya yaitu tentang penentuan dosen pembimbing skripsi

(Cahyono & Thamrin, 2013) dengan menggunakan metode AHP dikombinasikan dengan menggunakan metode *similarity*. Dalam penelitian ini akan dilakukan kecocokan mengenai judul yang diambil oleh mahasiswa dengan data salah satu dosen dengan mengacu pada penelitian yang pernah dilakukan oleh dosen, berikutnya menghitung kemiripan judul dengan jabatan dosen. Selanjutnya yaitu menghitung bobot kriteria mata kuliah berdasarkan konsentrasi. Tahap ini merupakan tahap untuk menentukan bobot kriteria mata kuliah. Dalam menghitung bobot ini yang diperlukan yaitu data mata kuliah setiap dosen dan mata kuliah tersebut sudah dikelompokan dalam konsentrasi masing-masing matakuliah. selanjutnya yaitu menghitung jumlah mata kuliah yang diajar oleh setiap dosen sesuai konsentrasi mahasiswa, memasukan nilai kemiripan judul dan bobot mata kuliah kedalam AHP. Tahap ini merupakan tahap terakhir dalam penentuan pembimbing skripsi. Pada tahap ini nilai dari kemiripan judul mahassiwa dengan paper, penelitian dan bimbingan skripsi, serta nilai bobot mata kuliah akan dimasukan dalam rumus AHP. Penelitian ini memudahkan biro skripsi dan prodi informatika UNY untuk untuk pemilihan pembimbing skripsi.

Dari beberapa penelitian diatas, perbedaan antar penelitian yang telah dilakukan dapat dilihat pada tabel 2.1

Tabel 2.1. Perbedaan beberapa penelitian terdahulu

* 1. **Landasan Teori**

1. Sistem Pendukung Keputusan

Bonczek, dkk., (1980) mendefinisikan sistem pengambilan keputusan sebagai sistem berbasis komputer yang terdiri dari 3 komponen yang saling berinteraksi: sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen sistem pendukung keputusab lain), sisten pengetahuan (repositori pengetahuan domain masalah uang ada pada DSSentah sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan)(Turban, Aronson, & Liang, 2005).

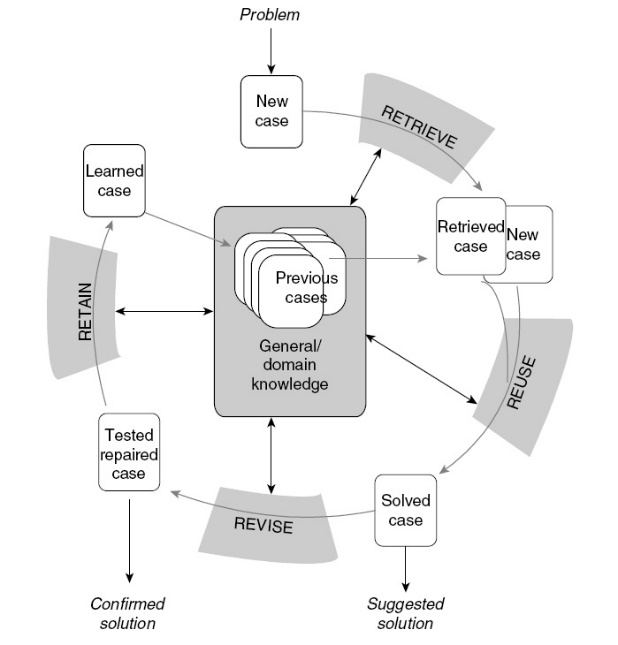
Komponen-komponen sistem pendukung keputussan sebagai berikut :

1. Subsistem manajemen data : Memasukkkan satu database yang berisi data yang relevan untuk situasi dan dikelola oleh perangkat lunak.
2. Subsistem manajemen model : Merupakan paket perangkat lunak yang memasukkan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat.
3. Subsistem antarmuka pengguna : Pengguna berkomunikasi dengan dan memerintahkan DSSmelalui subsistem ini. Pengguna adalah bagian yang dipertimbangkan dari sistem.
4. Subsistem manajemen berbasis-pengetahuan : Subsistem ini dapat mendukung semua subsistem lain atau bertindak sebagai suatu komponen independen(Turban et al., 2005).
5. *Case Based Reasoning* (CBR)
6. Pengertian *Case Based Reasoning*

Sistem berbasis kasus atau Case Based Reasoning (CBR) merupakan model penalaran untuk menyelesaikan masalah dengan menggunakan konsep analogi. Pada dasarnya, CBR akan membnadingkan suatu kasus baru dengan kasus-kasus lain yang sudah tersimpan sebelumnya. Selain itu juga akan dilakukan penandaan terhadap kasus-kasus klinis, serta menemukan kembali kasus-kasus yang mirip. Sehingga dapat dikatakan bahwa pada CBR ada dua fungsi yang dilakukan, yaitu modul penandaan kasus baru, dan aktivitas temu kembali kasus yang telah ada (Kusumadewi, 2009).

1. Siklus Hidup *Case Based Reasoning*

Siklus hidup pemecahan masalah dalam sistem CBR dasarnya terdiri dari empat seperti yang tergambar pada gambar 2.1.



Gambar 2.1. Siklus hidup CBR ((Kolodner, 1993)

1. *Retrieve* (memperoleh kembali) kasus atau kasus-kasus yang paling prinsip.

Proses ini dimulai dengan mendeskripsikan satu atau sebagian masalah dan berakhir apabila telah ditemukan kasus sebelumnya yang paling cocok. Sub tasknya mengacu pada identifikasi fitur, pencocokan awal, pencarian, dan pemeliharaan.

1. *Reuse* (menggunakan) informasi dan pengetahuan dari kasus tersebut untuk memecahkan permasalahan.

Proses *reuse* dari solusi kasus yang telah diperoleh dalam konteks kasus baru difokuskan pada dua aspek yaitu :

1) Perbedaan antara kasus yang sebelumnya dan yang sekarang.

2) Bagian apa dari kasus yang telah diperoleh yang dapat di transfer menjadi kasus baru.

c. *Revise* (meninjau kembali atau memperbaiki) usulan solusi.

Fase ini terdiri dari dua tugas, yaitu :

1) Mengevaluasi solusi kasus yang dihasilkan oleh *reuse*. Jika berhasil, maka dilanjutkan dengan proses retain.

2) Jika tidak maka memperbaiki solusi kasus menggunakan domain spesifikasi pengetahuan.

d. *Retain* (menyimpan) bagian-bagian dari pengalaman tersebut yang mungkin berguna untuk memecahkan masalah dimasa yang akan datang.

Proses ini terdiri dari memilih informasi apa dari kasus yang akan disimpan, disimpan dalam bentuk apa, cara penyusunan kasus untuk agar mudah untuk menentukan maslah yang mirip, dan bagaimana mengintegrasikan kasus baru pada struktur memori.

3. Kemiripan (*Similarity*)

Fungsi kemiripan (*Similarity*) adalah langkah yang digunakan untuk mengenali kesamaan atau kemiripan antar kasus-kasus yang tersimpan dalam basis kasus dengan kasus yang baru. Pengambilan data, pada langkah ini kasus-kasus yang telah tersimpan dalam basis kasus diambil atau dipilih sebagai sebuah solusi, dimana data ditampilkan dengan urutan tingkat nilai kemiripan (*similarity*) yang paling tinggi dengan range antara 0 sampai 1. Kasus dengan nilai *similarity* yang paling besar di anggap sebagai kasus yang “paling mirip”. Asumsi dasar yang digunakan adalah kasus yang mirip.

Untuk mencari nilai kemiripan *(similarity)* digunakan persamaan *2.1*

Ti = nx1 + nx2 + nx3 + . . . nxn

N

Keterangan :

Ti = Nilai kesamaan dengan kasus ke i

nx1, nx2, nx3, nxn = Nilai kesamaan objek x1, x2, x3, sampai xn

N = Banyak elemen pada basis kasus

1. Sistem Informasi Manajemen

Sistem informasi manajemen didefinisikan sebagai suatu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa. Para pemakai biasanya membentuk suatu entitas organisasi formal perusahaan atau sub unit dibawahnya. Informasi menjelaskan perusahaan atau salah satu sistem utamanya mengenai apa yang telah terjadi di masa lalu, apa yang sedang terjadi sekarang dan apa yang mungkin terjadi di masa depan. Output informasi digunakan oleh manajer maupun non manajer dalam perusahaan saat mereka membuat keputusan untuk memecahkan masalah(McLeod, 1998).

SIM dan susbsistem-subsistem organisasinya berkontribusi pada pemecahan masalah dalam dua cara dasar:

1. Sumber daya informasi seorganisasi

SIM adalah suatu usaha seorganisasi untuk menyediakan informasi pemecahan masalah. Sistem tersebut merupakan suatu komitmen formal dari para eksekutif untuk menyediakan komputer bagi semua manajer.

1. Identifikasi dan pemahaman masalah

Ide utama di balik SIM adalah menjaga agar pasokan informasi terus mengalir ke manajer. Manajer menggunakan SIM terutama untuk menandai masalah atau mendekati masalah, kemudian memahaminya dengan menentukan lokasi dan penyebabnya.

1. Rekayasa Perangkat Lunak

Menurut Fritz Bauer (NAU69) rekayasa perangkat lunak adalah pengembangan dan penggunaan prinsip pengembangan suara untuk memperoleh perangkat lunak secara ekonomis yang realibel dan bekerja secara efisien pada mesin nyata(Pressman, 2002).

Perangkat lunak lebih merupakan elemen logika dan bukan mmerupakan elemen sistem fisik. Dengan demikian, perangkat lunak memiliki ciri yang berbeda dari perangkat keras :

1. Perangkat lunak dibangun dan dikembangkan, tidak dibuat dalam bentuk yang klasik.
2. Perangkat lunak tidak pernah usang.
3. Sebagian besar perangkat lunak dibuat secara *custom-built*, serta tidak dapat dirakit dari komponen yang sudah ada.

Metode pengembangan perangkat lunak salah satunya yaitu Model Sekuensial Linier atau disebut juga dengan “siklus kehidupan klasik” atau “model air terjun”. *Sekuensial linier* mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivutas sebagai berikut:

1. Rekayasa dan pemodelan sistem/informasi, karena perangkat lunak selalu merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan perangkat lunak tersebut. Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen lain seperti perangkat lunak, manusia dan database. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.
2. Analisis Kebutuhan Perangkat, proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya perangkat lunak.
3. Desain, merupakan proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan kedalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode.
4. Generasi kode, desain harus diterjemahkan kedalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara otomatis.
5. Pengujian, sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.
6. Pemeliharaan, perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah perangkat lunak yang dilekatkan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya, atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.
7. *Unified Modelling Language* (UML)

*Unified Modelling Language* (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun (Dharwiyanti & Wahono, 2003).

UML mendefinisikan diagram-diagram antara lain :

1. Diagram *Use Case*

Diagram *use case* atau *use case* diagram menyajikan informasi antara *use case* dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai(Sholiq, 2006).

1. Diagram Aktivitas

Diagram aktivitas atau *actvity* diagram menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran bisnis (*business workflow*). Dapat juga digunakan untuk menggambarkan aliran kejadian *(flow of events*) dalam *use case*(Sholiq, 2006).

1. Basis Data Relasional SQL
2. Pembuatan Tabel

Digunakan untuk membuat tabel baru dengan spesifikasi atribut dan konstrin. Tiap atribut diberikan spesifikasi yang meliputi nama atribut, tipe data, dan konstrin. Penyataan *Primary Key* sebagai tanda persyaratn bahwa atribut yang disebutkan sebagai identitas utama dan tidak diperbolehkan adanya nilai kosong. Sedangkan *Foreign key* sebagai indikasi atibut yang dijadikan kunci atau identitas tamu(Waljiyanto, 2003).

1. Pemasukkan Data

Apabila tabel-tabel basis data sudah terbentuk, maka instan data dapat dimasukkan dengan menggunakan perintah insert sebagai berikut :

INSERT INTO Dosen VALUES (1300,’ Suryo’,’IV’,1005000.50);

Perintah tersebut akan memasukkan instan data pada tabel Dosen dengan nilai data pada baris pertama, demikian juga nilai data yang lain dimasukkan kedalam tabel dengan menggunakan perintah yang sama.

1. Manipulasi Data

Untuk memanipulasi data dalam basis data digunakan pertanyaan (“query”). Bentuk perintahnya secara umum dituliskan sebagai berikut :

SELECT <daftar atribut>

FROM <daftar tabel>

WHERE <kondisi>

Dalam hal ini,

1). <daftar atribut> adalah nama-nama atribut yang nilainya akan dipanggil dalam basis data.

2). <daftar tabel. Merupakan nama-nama tabel yang akan diproses oleh pertanyaan yang diajukan.

3). <kondisi> adalah persyaratan kondisi (Boolean) sebagai ekspresi yang harus dipenuhi untuk memanggil data dalam basis data.

Beberapa pertanyaan untuk mengambil informasi dari basis data :

1). Seleksi baris

Digunakan untuk menampilkan baris dari suatu tabel yang memenuhi kondisi sesuai dengan yang ditentukan. Contoh :

SELECT \* FROM Mahasiswa WHERE Jurusan = “Geodesi”;

Pernyataan diatas digunakan untuk memperoleh informasi dari tabel Mahasiswa tentang daftar mahasiswa yang ada jurusan Geodesi. Tanda (\*) merupakan notasi agar semua atribut yang ada pada tabel mahasiswa ditampilkan.

2) Seleksi Kolom

Digunakan untuk menampilkan isi kolom atribut pada tabel dengan kondisi yang ditentukan. Contoh :

SELECT NIP, Nama\_Dos FROM Dosen WHERE NIP <130555;

Pertanyaan ini digunakan untuk memilih daftar dosen dengan menampilkan NIP dan Nama\_Dos dari tabel Dosen dengan NIP lebih kecil 130555.

3) Operasi Penggabungan

Digunakan untuk menggabungkan beberapa tabel. Contoh :

SELECT \* FROM Mata\_Kuliah, Dosen WHERE Mata\_Kuliah.NIP = Dosen.NIP;

Fungsi diatas untuk menggabungkan tabel Mata\_Kuliah dan tabel Dosen dengan kondisi gabungan dalah atribut NIP.

4) Fungsi Agregat

Pada kasus tertentu perintah SQL dapat digunakan fungsi agregat, yaitu operasi dengan menggunakan sekelompok baris (“group”) dengan hasil pertanyaan hanya satu baris. Fungsi yang digunakan adalah :

1. AVG : operasi dengan sekelompok baris untuk memperoleh harga rerata dari kelompok yang ditentukan.a
2. COUNT(\*) : menghitung jumlajh baris dengan tidak memperhatikan nilai atribut duplikasi.
3. COUNT (DISTINCT <nama kolom>) : menghitung jumlah baris dengan memperhatikan nilai duplikasi atribut.
4. MAX : Memilih nilai terbesar dari sekelompok data dalam kolom atribut.
5. MIN : Memilih nilai terkecil dari sekelompok data dalam kolom atribut.
6. SUM : menjumlahkan sekelompok nilai data dalam kolom atribut.
7. Membuat “VIEW”

Dalam SQL terdapat fasilitas untuk membuat tabel sementara yang disebut dengan “VIEW”, yaitu tabel sementara yang dapat diakses tetapi secara fisik tidak tersimpan dalam basis data. Namun demikian definis “ VIEW” terdapat dalam sistem katalog. Contoh :

CREATE VIEW Gajidos AS SELECT Nama\_Dos, Golongan FROM Dosen WHERE Gaji > 1000000;

Perintah diatas digunakan untuk mememperoleh informsi nama dan golongan dosen yang gajinya lebih besar dari Rp 1.000.000,- dari tabel Dosen yang telah dibuat.