

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Menurut Abdul Kadir bahwa “Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan”, sedangkan menurut Sutabri bahwa “Sistem adalah suatu kumpulan atau himpunan dari suatu unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu” (Gobai et al., 2020). Menurut Sutarman bahwa “Sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama” (Doni & Lubis, 2019).

Dan menurut (Fatansyah, 2019) bahwa “Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu”. Dari pengertian yang dijabarkan di atas dapat diambil kesimpulan bahwa sistem merupakan kumpulan elemen, himpunan dari suatu unsur, komponen fungsional yang saling berhubungan dan berinteraksi satu sama lain untuk mencapai tujuan yang diharapkan.

2.1.1 Ciri-ciri Sistem

Sistem dapat dikatakan menjadi sebuah sistem yang bermanfaat bagi kehidupan manusia apabila dapat mendatangkan atau memberikan sebuah keberuntungan. Untuk mempermudah dalam mempelajari sebuah sistem terdapat beberapa ciri-ciri sistem yang dapat diketahui untuk dapat dijadikan sebagai informasi pengetahuan. Dimana, berikut merupakan beberapa ciri-ciri sistem secara umum:

1. **Memiliki Tujuan**

Tujuan dapat dikatakan sebagai goals yang harus tercapai. Dalam konteks apapun, tujuan selalu ada. Begitu juga dalam sistem, sistem tentu akan memiliki sebuah tujuan tergantung dari dalam aspek apa

sistem tersebut digunakan. Perbedaan penggunaan sistem dalam suatu bidang dapat mempengaruhi tujuan dari sistem tersebut.

2. Memiliki Batas

Batas merupakan salah satu ciri yang dimiliki oleh sistem. Dimana batas merupakan suatu batasan sebuah sistem untuk menunjukkan adanya pemisah antara sistem dengan lingkungan eksternalnya. Dengan adanya batasan inilah sistem dapat ditentukan konfigurasi, ruang lingkupnya, dan kemampuan dari sistem itu sendiri.

3. Subsistem

Subsistem ini menggambarkan adanya sebuah sistem yang saling terkait dan berhubungan satu sama lain. Sehingga dalam proses pencapaian tujuannya akan lebih mudah.

4. Keterikatan

Sama halnya dengan subsistem, dalam sebuah sistem pasti akan memiliki keterkaitan antar subsistem atau komponen lainnya yang mendukung. Hal ini agar tujuan dari sistem tersebut lebih mudah dicapai.

2.1.2 Karakteristik Sistem

Sistem juga memiliki karakteristik yang menunjukkan bahwa hal tersebut adalah sistem. Karakteristik dari sistem tersebut yang dapat membuat kita mencarapai tujuan yang diinginkan. Berikut adalah beberapa karakteristik dari sistem :

1. Komponen

Komponen menjadi salah satu hal penting dalam sistem. Karena, meskipun sebuah sistem dapat memberikan banyak manfaat, tetapi tidak memiliki komponen yang mendukungnya, sistem tersebut tidak akan bernilai. Komponen dalam sistem akan beragam jenisnya dan bisa disebut dengan sub-sistem.

2. *Boundary*

Batasan dalam sebuah sistem dapat menggambarkan ruang lingkup dari sistem tersebut. Batasan ini berfungsi untuk membatasi kinerja dari sistem agar tujuan utama dari sistem tersebut dapat terpenuhi.

3. *Environment*

Lingkungan sistem merupakan salah satu bagian dari sistem yang tuugasnya memberikan ruang bagi sistem dalam bekerja, atau ruang lingkup dari sebuah sistem.

4. *Interface*

Interface merupakan penghubung antar subsistem yang terdapat dalam sistem, yang tujuan utamanya adalah sebagai alat bantu komunikasi antar subsistem untuk mendukung kinerja sistem sesuai dengan tujuan yang ingin dicapai

5. *Input*

Input adalah komponen sistem yang berfungsi sebagai alat memasukkan data atau inputan yang diberikan oleh developer maupun pengguna demi menghasilkan output yang diinginkan nantinya.

6. *Output*

Output adalah keluaran atau hasil yang telah diproses dari inputan sebelumnya.

7. *Process*

Process adalah tahap pengolahan inputan menjadi output dalam sistem

8. *Objective and Goals*

Objective and goals merupakan gambaran dari tujuan yang ingin dicapai oleh sebuah sistem tertentu. Dimana tujuan ini dapat berpengaruh ke pengguna sistemnya karena bisa mendapatkan informasi yang berguna dan bermanfaat.

2.1.3 Komponen Sistem

Sistem juga memiliki komponen yang mendukung berjalannya sistem itu sendiri. Komponen-komponen berfungsi untuk menjalankan tugasnya sesuai dengan fungsi yang telah di programkan kedalamnya. Sehingga pekerjaan yang diberikan kepada sistem dapat dengan mudah diselesaikan. Berikut ini adalah komponen sistem :

1. *Input*, merupakan masukan yang bisa berupa fakta atau data apapun yang bisa mendukung tercapainya tujuan suatu sistem.
2. Proses, merupakan tahapan pengelolaan input sistem sehingga dapat menjadi suatu informasi yang bermanfaat sehingga tujuan sistem dapat tercapai.
3. *Output*, merupakan hasil yang didapatkan dari sebuah sistem. Dimana output ini dapat berupa informasi penting ataupun output lainnya yang dapat menggambarkan kesuksesan pencapaian tujuan dari suatu sistem.
4. Kendala, merupakan masalah yang dapat terjadi saat sistem dijalankan, sehingga perlu adanya solusi untuk menyelesaikan masalah ini.
5. *Control*, merupakan pemantauan sistem yang dilakukan agar tidak ada masalah yang terjadi.
6. *Feedback*, merupakan respon yang diberikan oleh sistem ketika sedang aktif.

2.2 Sistem Pakar

Sistem pakar, atau yang sering disebut sebagai (*Expert system*), merupakan suatu sistem yang memiliki kemampuan untuk mengakumulasi pengetahuan seorang pakar ke dalam komputer. Hal ini memungkinkan komputer untuk menangani masalah dengan cara yang serupa dengan seorang ahli atau pakar. Pada Sub bab ini akan membahas berbagai aspek terkait sistem pakar, mulai dari definisi, karakteristik, kelebihan dan kekurangan, konsep dasar, struktur, basis pengetahuan,

hingga mesin inferensi dalam sistem pakar (Maulina & Wulanningsih, 2020) Penjelasan rinci mengenai hal-hal tersebut adalah sebagai berikut:

2.2.1 Pengertian Sistem Pakar

Pada awalnya, Sistem pakar dikembangkan oleh General Purpose Problem Solver (GPPS) pada tahun 1960 oleh Newel Simon. Secara keseluruhan, sistem pakar merupakan suatu sistem yang bertujuan menggantikan peran pakar dalam mendiagnosa hal-hal tertentu. Penggantian yang dimaksud di sini tidak bersifat mutlak, melainkan melibatkan integrasi kemampuan pakar ke dalam sistem. Dengan menyatukan kemampuan pakar ke dalam sistem, maka sistem tersebut dapat menangani situasi yang sesuai dengan keahlian yang dimiliki oleh sistem. Sistem Pakar adalah bidang khusus dalam ilmu komputer yang memungkinkan komputer berpikir dan berperilaku mirip manusia (Aldo, 2020). Sebagai suatu program, sistem pakar dapat melakukan aktivitas yang biasanya dilakukan oleh seorang pakar, menghasilkan output yang hampir sebanding dengan kemampuan manusia dalam berpikir.

2.2.2 Karakteristik Sistem Pakar

Sistem pakar sendiri memiliki beberapa karakteristik yang meliputinya. Berdasarkan buku yang ditulis oleh (Hayadi, 2018) berikut ini adalah karakteristik dari sistem pakar :

1. Memiliki keterbatasan pada domain keahlian tertentu;
2. Mampu melakukan penalaran terhadap data yang tidak lengkap atau data yang tidak valid;
3. Memberikan penjelasan atas alasan-alasan dengan cara yang mudah dimengerti;
4. Beroperasi dengan rule dan kaidah-kaidah tertentu;
5. Fleksibel untuk dimodifikasi;
6. Terdiri dari mekanisme dan basis pengetahuan yang terpisah;
7. Menghasilkan output yang bersifat anjuran;
8. Mampu memberikan kaidah yang sesuai dan sejalan.

2.2.3 Kelebihan dan Kekurangan Sistem Pakar

Dalam mempelajari sistem pakar ini, sangat penting untuk mengetahui kelebihan dan kekurangan yang melekat di dalamnya. Dengan memahami dengan baik kelebihan dan kekurangan tersebut, kita dapat memperoleh pandangan yang lebih komprehensif terkait dengan kinerja dan potensi perbaikan sistem tersebut. Berikut ini adalah kelebihan yang dirangkum dalam buku yang ditulis oleh (Hayadi, 2018) :

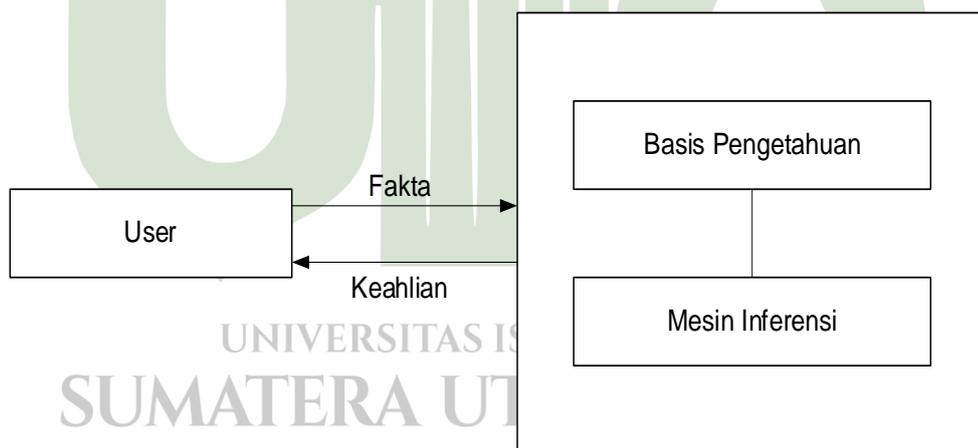
1. Mampu mengumpulkan data dalam jumlah yang besar;
2. Dapat menyimpan data dalam format tertentu dan dalam jangka waktu yang sangat panjang;
3. Mampu mencari data yang tersimpan dengan kecepatan tinggi serta melakukan perhitungan dengan cepat dan akurat;
4. Meningkatkan produktivitas;
5. Memungkinkan individu yang awam untuk bekerja sebagaimana seorang pakar yang ahli;
6. Memberikan nasihat yang konsisten dan mengurangi kesalahan untuk meningkatkan kualitas;
7. Mampu menangkap pengetahuan dan keahlian seseorang;
8. Tetap dapat beroperasi di lingkungan yang berbahaya;
9. Memfasilitasi akses ke pengetahuan seorang pakar;
10. Meningkatkan kapabilitas komputer;
11. Dapat bekerja meskipun dengan informasi data yang tidak valid atau tidak pasti; dengan berkonsultasi pada sistem pakar, jawaban tetap dapat diberikan;
12. Pengguna awam yang bekerja dengan sistem pakar akan menemui kemudahan karena adanya fasilitas penjelasan yang berfungsi sebagai pemandu dan dapat digunakan sebagai media pelatihan atau pelengkap;
13. Mampu meningkatkan kemampuan dalam menyelesaikan masalah dengan mengambil data penyelesaian masalah dari seorang pakar.

Sedangkan kekurangan yang terdapat pada sistem pakar, dan dirangkum dalam buku yang ditulis oleh (Hayadi, 2018), adalah sebagai berikut :

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar butuh pengujian ulang hingga beberapa kali, karena sistem pakar tidak 100% bernilai benar.

2.2.4 Konsep Dasar Sistem Pakar

Sistem pakar sendiri memiliki konsep dasar yang terbentuk dalam penyusunan kaidah atau pola kerja dalam melakukan inferensi menggunakan basis pengetahuan yang diperoleh dari pakar yang memiliki keahlian dalam bidang tertentu. Kombinasi informasi ini kemudian diintegrasikan ke dalam sistem komputer, yang selanjutnya digunakan untuk proses pengambilan keputusan dalam menyelesaikan masalah. Pada dasarnya, sistem pakar dibangun berdasarkan pengetahuan yang dimiliki oleh seorang pakar, yang kemudian diubah menjadi format yang dapat diproses oleh komputer untuk menangani persoalan sejenis. Keberhasilan penyelesaian masalah oleh sistem pakar sangat bergantung pada pemahaman mendalam terhadap suatu bidang khusus (Hayadi, 2018).

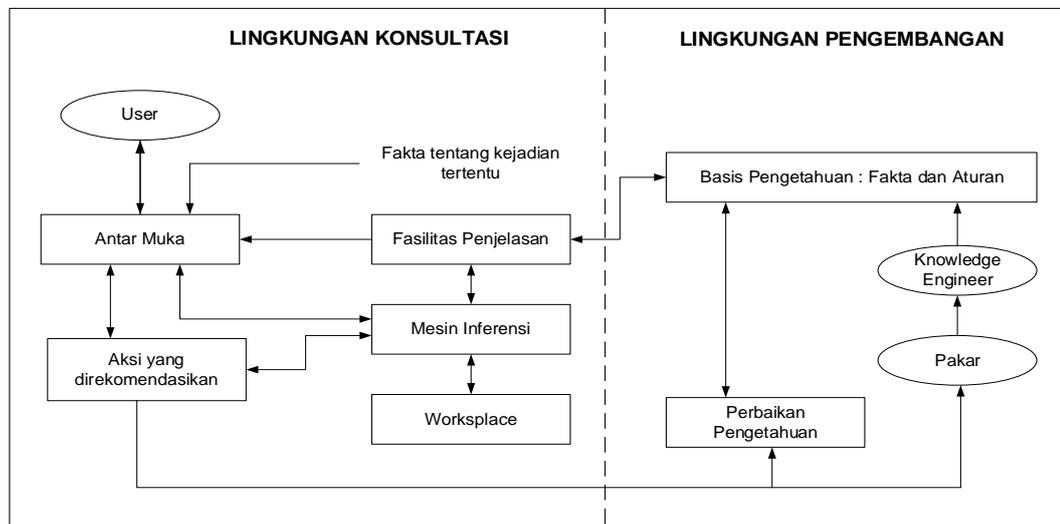


Gambar 2.1 Konsep Dasar Sistem Pakar

2.2.5 Arsitektur Sistem Pakar

Sistem pakar terdiri dari dua komponen utama, yaitu lingkungan pengembangan (*development environment*) dan lingkungan konsultasi (*consultation environment*). Lingkungan pengembangan mencakup berbagai elemen yang digunakan untuk mengintegrasikan kemampuan pakar ke dalam

struktur sistem pakar, sementara lingkungan konsultasi mencakup elemen-elemen yang akan dimanfaatkan oleh pengguna untuk memperoleh pengetahuan dari pakar (Hayadi, 2018).



Gambar 2.2 Arsitektur Sistem Pakar

Berikut ini adalah penjelasan tentang sistem pakar, yaitu user interface (antar muka pengguna), basis pengetahuan, akuisisi pengetahuan, mesin interface, workplace, fasilitas penjelasan dan perbaikan pengetahuan.

1. Fasilitas Akuisisi Pengetahuan

Fasilitas akuisisi pengetahuan adalah suatu proses akumulasi, transfer, dan penerapan keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer (H.W.Putra,2019). Fasilitas ini melibatkan pengumpulan data pengetahuan tentang suatu masalah dari pakar. Pengetahuan dapat diperoleh melalui studi pustaka, observasi, serta wawancara langsung dengan pakar. Pengetahuan dan data yang terkumpul ini dikenal sebagai *knowledge base* atau basis pengetahuan.

2. Basis Pengetahuan

Basis pengetahuan berisi pengetahuan yang diperlukan untuk memahami, merumuskan, dan menyelesaikan suatu masalah. Komponen ini dalam

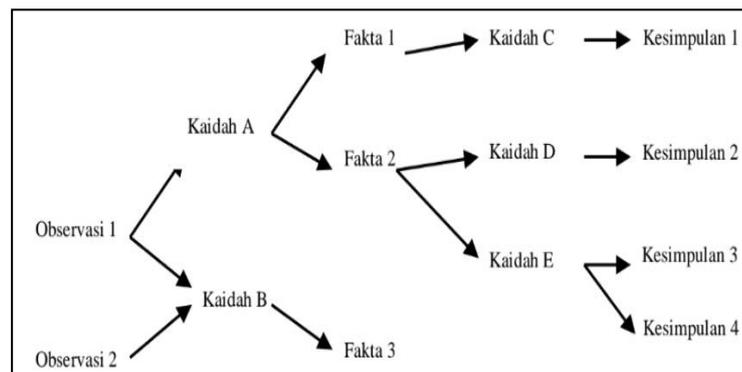
sistem pakar terdiri dari dua elemen dasar, yaitu fakta dan aturan. Fakta mencakup informasi tentang objek dalam suatu area permasalahan tertentu, sementara aturan mencakup informasi tentang cara memperoleh fakta yang sudah diketahui (H.W.Putra, 2019).

3. Mesin Inferensi

Mesin Inferensi merupakan sebuah program komputer yang menyediakan metodologi untuk melakukan penalaran terhadap informasi yang terdapat dalam basis pengetahuan dan lingkungan kerja, serta untuk menarik kesimpulan. Selama proses konsultasi antara sistem dan pengguna, mekanisme antarmuka menguji aturan satu per satu hingga kondisi aturan tersebut terpenuhi (H.W.Putra,2019). Secara keseluruhan, terdapat dua teknik utama yang digunakan dalam mekanisme inferensi untuk menguji aturan, yaitu penalaran maju (*forward chaining*) dan penalaran mundur (*backward chaining*).

a. *Forward Chaining*

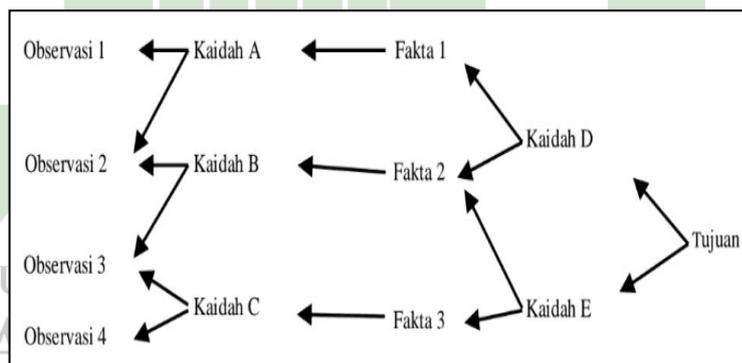
Forward chaining merupakan proses penelusuran yang dimulai dengan menampilkan kumpulan data atau fakta yang meyakinkan menuju suatu simpulan akhir. *Forward chaining*, atau pelacakan atau penalaran ke depan, adalah metode penarikan simpulan yang berdasarkan pada data atau fakta yang ada untuk mencapai suatu kesimpulan. Proses penelusuran ini dimulai dari fakta yang ada dan bergerak maju melalui premis-premis untuk mencapai kesimpulan, yang sering disebut sebagai *bottom-up reasoning*. *Forward chaining* melakukan pencarian dari suatu masalah menuju solusinya (Hayadi, 2018).



Gambar 2.3 Mesin Inferensi *Forward Chaining*

b. *Backward Chaining*

Pelacakan atau penalaran ke belakang (*backward chaining*) merupakan pendekatan yang didorong oleh tujuan (*goal-driven*). Penalaran ini sering disebut sebagai penalaran dari atas ke bawah, di mana proses dimulai dari tingkat tertinggi untuk membangun suatu hipotesis, kemudian turun ke tingkat paling rendah yang dapat mendukung hipotesis tersebut. Dengan kata lain, dalam *backward chaining*, fakta yang sudah ada digunakan untuk mendukung hipotesis yang sudah dibuat (Hayadi, 2018)..



Gambar 2.4 Mesin Inferensi *Backward Chaining*

4. *Workplace*

Workplace merupakan area dari sekumpulan memori kerja (*working memory*). *Working memory* berfungsi untuk menyimpan fakta yang dihasilkan oleh mesin antarmuka dengan penambahan informasi berupa tingkat kepercayaan atau dikenal juga sebagai basis data global dari fakta yang digunakan oleh aturan yang ada (H. W. Putra, 2019).

5. *User Interface*

User Interface adalah cara yang digunakan oleh pengguna dan sistem pakar untuk berkomunikasi (H. W. Putra, 2019). Antarmuka pengguna memberikan fasilitas komunikasi antara pengguna dan sistem, menyediakan berbagai informasi dan keterangan untuk membantu memandu alur penelusuran masalah hingga menemukan solusi. Beberapa syarat utama dalam mendesain antarmuka pengguna termasuk kemudahan pengoperasian sistem, tampilan interaktif, komunikatif, dan kemudahan penggunaan.

2.3 *Bullying*

Menurut Olweus pada bukunya mengatakan bahwasannya *bullying* adalah tindakan atau perilaku agresif yang disengaja, yang dilakukan oleh sekelompok orang atau seseorang berulang kali dan dari waktu ke waktu kepada seorang korban yang tidak dapat mempertahankan dirinya dengan mudah atau sebagai penyalahgunaan kekuasaan / kekuatan sistematis (Samsiyah et al., 2023).

Sedangkan menurut Wicaksana selaku pakar dibidangnya mengatakan *Bullying* adalah kekerasan fisik dan psikologis jangka panjang yang dilakukan oleh seseorang atau kelompok, terhadap seseorang yang tidak dapat membela diri dalam situasi di mana ada keinginan untuk menyakiti atau menakut-nakuti orang tersebut atau membuatnya murung (Kurniawan & Pranowo, 2018).

Dan menurut Black and Jackson pada bukunya mengatakan bahwa, *Bullying* adalah tipe perilaku agresif proaktif di mana ada aspek yang disengaja untuk mendominasi, menyakiti, atau menyingkirkan, ada ketidak seimbangan kekuatan baik secara fisik, usia, kemampuan kognitif, keterampilan, dan status sosial, dan dilakukan berulang kali oleh satu atau beberapa anak terhadap anak lain (Wanita Marwan et al., 2023)..

Secara umum *bullying* dapat diartikan sebagai perilaku agresif atau menindas yang dilakukan secara berulang-ulang oleh satu individu atau kelompok terhadap individu lain yang memiliki ketidakseimbangan kekuatan atau kekuasaan. Perilaku ini biasanya terjadi di berbagai lingkungan, termasuk di sekolah, tempat kerja, dan dalam dunia maya.

2.3.1 Faktor Penyebab *Bullying*

Terdapat banyak faktor yang menjadi alasan atau awal dari suatu kasus *bullying* terjadi, namun secara umum terdapat 5 faktor yang menjadi faktor utama pemicu kasus *bullying* (Sapitri, 2020), sebagai berikut :

1. Keluarga

Pelaku *bullying* seringkali berasal dari keluarga yang bermasalah : orang tua yang sering menghukum anaknya secara berlebihan, atau situasi rumah yang penuh stress, agresi, dan permusuhan. Anak akan mempelajari perilaku *bullying* ketika mengamati konflik-konflik yang terjadi pada orang tua mereka, dan kemudian menirunya terhadap teman-temannya. Jika tidak ada konsekuensi yang tegas dari lingkungan terhadap perilaku cobacobanya itu, ia akan belajar bahwa “mereka yang memiliki kekuatan diperbolehkan untuk berperilaku agresif, dan perilaku agresif itu dapat meningkatkan status dan kekuasaan seseorang”. Dari sini anak mengembangkan perilaku *bullying*

2. Sekolah

Pihak sekolah sering mengabaikan keberadaan *bullying* ini. Akibatnya, anak-anak sebagai pelaku *bullying* akan mendapatkan penguatan terhadap perilaku mereka untuk melakukan intimidasi terhadap anak lain. *Bullying* berkembang dengan pesat dalam lingkungan sekolah sering memberikan masukan negatif pada siswanya, misalnya berupa hukuman yang tidak membangun sehingga tidak mengembangkan rasa menghargai dan menghormati antar sesama anggota sekolah

3. Faktor Kelompok Sebaya

Anak-anak ketika berinteraksi dalam sekolah dan dengan teman di sekitar rumah, kadang kala terdorong untuk melakukan *bullying*. Beberapa anak melakukan *bullying* dalam usaha untuk membuktikan bahwa mereka bisa masuk dalam kelompok tertentu, meskipun mereka sendiri merasa tidak nyaman dengan perilaku tersebut

4. Kondisi lingkungan sosial

Kondisi lingkungan sosial dapat pula menjadi penyebab timbulnya perilaku *bullying*. Salah satu faktor lingkungan social yang menyebabkan tindakan *bullying* adalah kemiskinan. Mereka yang hidup dalam kemiskinan akan berbuat apa saja demi memenuhi kebutuhan hidupnya, sehingga tidak heran jika di lingkungan sekolah sering terjadi pemalakan antar siswanya.

5. Tayangan televisi dan media cetak

Televisi dan media cetak membentuk pola perilaku *bullying* dari segi tayangan yang mereka tampilkan. Survey yang dilakukan Kompas memperlihatkan bahwa 56,9% anak meniru adegan-adegan film yang ditontonnya, umumnya mereka meniru gerakannya (64%) dan kata-katanya (43%).

2.3.2 Jenis *Bullying*

Terdapat beberapa jenis *bullying* yang sering terjadi (Sapitri, 2020), sebagai berikut

1. *Bullying* Verbal

Kekerasan verbal adalah bentuk penindasan yang paling umum digunakan, baik oleh anak perempuan maupun anak laki-laki. Kekerasan verbal mudah dilakukan dan dapat dibisikkan dihadapan orang dewasa serta teman sebaya, tanpa terdeteksi. Penindasan verbal dapat diteriakkan di taman bermain bercampur dengan hingar binger yang terdengar oleh pengawas, diabaikan karena hanya dianggap sebagai dialog yang bodoh dan tidak simpatik di antara teman sebaya. Penindasan verbal dapat berupa julukan nama, celaan, fitnah, kritik kejam, penghinaan, dan pernyataan-pernyataan bernuansa ajakan seksual atau pelecehan seksual. Selain itu, penindasan verbal dapat berupa perampasan uang jajan atau barang-barang, telepon yang kasar, e-mail yang mengintimidasi, surat-surat kaleng yang berisi ancaman kekerasan, tuduhantuduhan yang tidak benar, kasak-kusuk yang keji, serta gosip.

2. *Bullying Physical*

Penindasan fisik merupakan jenis *bullying* yang paling tampak dan paling dapat diidentifikasi diantara bentuk-bentuk penindasan lainnya, namun kejadian penindasan fisik terhitung kurang dari sepertiga insiden penindasan yang dilaporkan oleh siswa. Jenis penindasan secara fisik di antaranya adalah memukul, mencekik, menyikut, meninju, menendang, menggigit, memiting, mencakar, serta meludahi anak yang ditindas hingga ke posisi yang menyakitkan, serta merusak dan menghancurkan pakaian serta barang-barang milik anak yang tertindas. Semakin kuat dan semakin dewasa sang penindas, semakin berbahaya jenis serangan ini, bahkan walaupun tidak dimaksudkan untuk mencederai secara serius.

3. *Bullying Social*

Jenis ini paling sulit dideteksi dari luar. Penindasan *social* adalah pelemahan harga diri si korban penindasan secara sistematis melalui pengabaian, pengucilan, pengecualian, atau penghindaran. Penghindaran, suatu tindakan penyingkiran, adalah alat penindasan yang terkuat. Anak yang digunjingkan mungkin akan tidak mendengar gosip itu, namun tetap akan mengalami efeknya. Penindasan relasional dapat digunakan untuk mengasingkan atau menolak seorang teman atau secara sengaja ditujukan untuk merusak persahabatan. Perilaku ini dapat mencakup sikap-sikap tersembunyi seperti pandangan yang agresif, lirik mata, helaan napas, bahu yang bergidik, cibiran, tawa mengejek, dan bahasa tubuh yang kasar.

4. *Bullying Cyber*

Penindasan *cyber* adalah bentuk intimidasi yang terjadi melalui platform digital seperti media sosial, pesan teks, email, dan forum online. Pelaku menggunakan teknologi untuk melecehkan, mengancam, atau mempermalukan korban di depan audiens yang luas dan sering anonim. Tindakan ini bisa mencakup menyebarkan rumor, mengirim pesan kebencian, atau memposting gambar yang memalukan. Dampaknya bisa sangat merusak karena dapat menyebar dengan cepat dan sulit dihapus, menyebabkan korban merasa tertekan, cemas, dan terisolasi.

5. *Bullying Emotional*

Penindasan *emotional* adalah bentuk intimidasi yang menargetkan perasaan dan harga diri korban. Ini bisa melibatkan penghinaan, ancaman, pelecehan verbal, atau perlakuan merendahkan yang terus-menerus. Tujuan dari bullying emosional adalah untuk membuat korban merasa tidak berharga, takut, atau cemas. Dampak jangka panjangnya bisa sangat serius, termasuk depresi, kecemasan, dan gangguan kesehatan mental lainnya. Bullying emosional seringkali dilakukan di balik layar, membuatnya sulit untuk diidentifikasi dan ditangani secara efektif.

2.4 Metode *Certainty Factor*

Certainty Factor adalah sebuah metode yang digunakan untuk mengatasi ketidakpastian dalam menganalisa suatu informasi. Metode ini mendefinisikan keyakinan terhadap suatu fakta atau aturan berdasarkan tingkat keyakinan suatu atau seorang pakar. Metode ini membantu sistem pakar untuk mengatasi ketidakpastian yang mungkin terjadi dalam proses penalaran dengan pola pakar yang sebelumnya sudah ditetapkan (Rachman & Mukminin, 2018).

Certainty Factor (CF) bekerja dengan mengukur tingkat keyakinan atau ketidakpastian dalam suatu keputusan berdasarkan informasi yang ada. Cara kerjanya adalah dengan menginterpertasikan sejauh mana informasi baru, mempengaruhi keyakinan awal yang ada. Kemudian hasil atau output yang didapatkan adalah CF positif, dan CF negatif, CF positif sendiri berarti informasi baru mendukung kebenaran dari kejadian yang terjadi, sedangkan CF negatif menunjukkan informasi baru yang dimasukkan mendukung ketidak benar dari kejadian yang terjadi (R. S. Putra & Yuhandri, 2021). Berikut ini adalah komponen utama dari metode *Certainty Factor* :

1. Positif *Certainty Factor* (CF⁺): Merupakan nilai yang menunjukkan tingkat keyakinan bahwa suatu kejadian atau pernyataan benar. Nilai ini berkisar antara 0 hingga 1, di mana 1 menunjukkan keyakinan penuh.

2. Negatif *Certainty Factor* (CF^-): Merupakan nilai yang menunjukkan tingkat keyakinan bahwa suatu kejadian atau pernyataan salah. Nilai ini juga berkisar antara 0 hingga 1, dengan 1 menunjukkan keyakinan penuh.
3. Nilai Awal *Certainty Factor* (CF_0): Nilai awal CF sebelum adanya informasi tambahan. Nilai ini dapat diberikan oleh pakar manusia atau dihitung berdasarkan distribusi probabilitas awal.
4. Informasi Tambahan atau Data Baru (D): Merupakan informasi tambahan yang diperoleh atau diobservasi oleh sistem pakar yang dapat mempengaruhi tingkat keyakinan terhadap suatu kejadian atau pernyataan.

Sedangkan rumus umum yang biasa digunakan dalam melakukan perhitungan dengan menggunakan metode *Certainty Factor*, adalah sebagai berikut :

$$CF(A, B) = \frac{P(A|B) - P(-A|B)}{1 - \min [P(A|B), P(-A|B)]} \quad (1)$$

Dimana :

$CF(A, B)$: adalah nilai dari certainty factor untuk kejadian A berdasarkan informasi B

$P(A|B)$: adalah probabilitas bahwa A benar berdasarkan informasi B

$P(-A|B)$: adalah probabilitas bahwa A salah berdasarkan informasi B

2.5 Metode Bayes

Metode *Bayes* atau yang biasa disebut dengan *Naïve Bayes*, dikemukakan oleh seorang ilmuwan asal Inggris bernama Thomas *Bayes*. Metode ini sendiri dikemukakan olehnya dengan tujuan sebagai alat pengklasifikasian data dengan memanfaatkan probabilitas dan statistik, untuk memprediksi suatu kejadian berdasarkan pengalaman-pengalaman dimasa sebelumnya. Banyak sumber maupun pakar yang mengatakan bahwasannya metode *Bayes* ini tergolong metode yang cocok dan efisien dalam hal pengklasifikasian dibanding metode lainnya (Dwiramadhan et al., 2022). Berikut ini adalah tahapan dari proses perhitungan metode *Bayes* :

1. Menentukan probabilitas awal (*Prior Probability*) dari suatu peristiwa sebelum mendapatkan informasi tambahan.
2. Melakukan pencarian informasi baru yang relevan dengan peristiwa yang menjadi objek penelitian.
3. Menghitung probabilitas bersyarat (*Likelihood*) dengan cara mengamati seberapa besar kemungkinan informasi baru yang sebelumnya sudah didapatkan, dapat mempengaruhi peristiwa yang terjadi.
4. Menghitung probabilitas marginal dari informasi baru dengan menentukan probabilitas margin dari informasi baru. Dimana sebuah informasi baru terjadi independen dari apakah peristiwa yang diamati adalah benar atau tidak
5. Menerapkan Teorema *Bayes* untuk menghitung probabilitas posterior, yaitu probabilitas peristiwa setelah mempertimbangkan informasi baru.
6. Terakhir adalah memperbarui keyakinan atau probabilitas tentang peristiwa tersebut. Probabilitas posterior ini dapat menjadi probabilitas awal untuk iterasi berikutnya jika ada informasi tambahan.

Sedangkan rumus umum yang biasa digunakan dalam melakukan perhitungan dengan menggunakan metode *Bayes* atau biasa disebut dengan Teroema *Bayes*, adalah sebagai berikut :

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2)$$

Dimana :

$P(A|B)$: probabilitas posterior dari peristiwa A setelah memperoleh informasi B.

$P(B|A)$: probabilitas dari peristiwa B jika peristiwa A terjadi.

$P(A)$: probabilitas dari peristiwa A.

2.6 Aplikasi Berbasis Web

Aplikasi *web* berbeda dengan situs web, dimana situs web bersifat statis atau tetap karena hanya menampilkan isi dari berkas statis dan sedikit mengalami perubahan tampilan dan informasi yang ditampilkan, sedangkan aplikasi web lebih bersifat dinamis karena menampilkan isi yang dinamis tergantung pada permintaan klien dan berdasarkan data yang dimasukkan. Web aplikasi dibangun dengan

menggunakan bahasa pemrograman yang sudah terintegrasi dengan basis data, sehingga dapat memungkinkan terjadinya aktifitas masukan (*input*), proses (*process*) dan keluaran atau *output* (Putri et al., 2020).

2.7 HTML

HTML adalah singkatan dari *HyperText Markup Language*. HTML didefinisikan sebagai bahasa *markup* berbasis teks untuk membuat halaman web dinamis. Bahasa *markup* yang digunakan merupakan untuk menerjemahkan dan mendeskripsikan teks halaman web, gambar, dan materi lainnya secara visual dan suara. Menurut Tim EMS dalam (Maryani et al., 2018), “*Hypertext Markup Language* atau *HTML* adalah bahasa pemformatan teks untuk dokumen di jaringan komputer dikenal dengan *World Wide Web (WWW)*”. Yang dapat dilakukan dengan *HTML* yaitu: (Fitri Ayu, 2018)

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halaman *web*.
3. Mempublikasikan halaman *web* secara *online*.
4. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.

2.8 PHP

PHP adalah singkatan dari *Personal Home Page* yang digunakan untuk membuat *website*. *PHP* adalah *script* yang terintegrasi dengan *HTML* dan berada pada *server (server side HTML embedded scripting)* (M. Arfa, 2021). Ketika terkoneksi jaringan internet dan telah mendapatkan URL maka *browser* akan mencari hingga menemukan sebuah alamat dari *web server*, kemudian dilakukan identifikasi dan *web server* mulai menampilkan informasi yang dibutuhkan. *Browser* akan menerjemahkan kode *HTML* dan menampilkan pada halaman *web* sesuai dengan permintaan *user*, tapi apabila terdapat *script PHP* di dalam file, maka proses diteruskan dengan menerjemahkan *script-script PHP* dan diolah pada beberapa modul *PHP*, setelah diterjemahkan menjadi bahasa *HTML* maka akan langsung ditampilkan pada *browser user* (Kristanti, 2020).

PHP berasal dari kata “*Hypertext Preprocessor*” bahasa pemrograman yang banyak digunakan pengembang *web*. *PHP* dapat disematkan dalam *HTML* dan bersifat umum atau open *source*. Pada saat mengakses halaman *web*, kode *PHP* dibaca oleh *server* tempat halaman itu berada. Bahasa ini memiliki tujuan agar pengembang *web* dapat menulis halaman yang dihasilkan secara dinamis dengan cepat dan mudah (Dwihatami, 2021). Komponen dasar *PHP*, yaitu: (Fitri Ayu, 2018)

1. Sintaks dasar *PHP*

Ada beberapa aturan sintaks yang harus dipenuhi ketika membuat file program *PHP*.

- a. *PHP opening* dan *closing tag*.
- b. *PHP* mendukung komentar seperti bahasa ‘C’, ‘C++’ dan *Unix shell-style (perl style)*.

2. Konstanta dalam *PHP*

Suatu konstanta dapat didefinisikan dengan menggunakan fungsi *define ()* yang merupakan salah satu fitur *function* dari *PHP*.

3. Aritmatika dalam *PHP*

Untuk mempermudah menggunakan *operand* dan operator pada *PHP*.

2.9 *MySQL*

MySQL merupakan *software database open source* yang sering digunakan untuk mengolah basis data yang menggunakan bahasa SQL. Menurut (Daniel Dodi, 2020), “*MySQL* adalah aplikasi yang digunakan untuk mengolah basis data yang banyak digunakan untuk membangun aplikasi yang menggunakan *database*”. Sistem database *MySQL* menggunakan arsitektur client-server. *Client-server* memiliki kendala pusat pada *server*, dan *server* bertindak sebagai program yang dapat memanipulasi *database*.

Program client menjelaskan maksud pengguna ke server dengan menulis query dalam bahasa SQL (Dwihatami, 2021). *MySQL* adalah singkatan dari (*My Structure Query System Language*). *MySQL* adalah software RDBMS (*Relational*

Database Management System) atau *server database* yang dapat mengolah *database* dengan cepat, bisa menampung data pada jumlah yang sangat besar, dan bisa diakses oleh banyak *user (multi-user)* (Maryani et al., 2018).

2.10 XAMPP

Menurut Wicaksono dalam (Maryani et al., 2018), “XAMPP adalah sebuah software yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL dikomputer lokal. XAMPP merupakan paket dari perangkat lunak atau *software* yang berisi *Apache HTTP server* dan MySQL. Dengan menginstal XAMPP dapat menjalankan sebuah *web server* pada komputer karena XAMPP mendukung program PHP, dan *Perl* dan juga berjalan di beberapa sistem operasi lain (Dwihatami, 2021).

2.11 Unified Model Language (UML)

Pada perkembangan teknik pemrograman berorientasi objek, muncul sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek (Putra, 2018).

Menurut Rosa A.S dan M. Shalahudin dalam (Daniel Dodi, 2020), “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram atau teks-teks pendukung. UML hanya sebagai pemodelan”. UML merupakan kumpulan diagram-diagram yang sudah memiliki standar untuk membangun perangkat lunak berbasis objek (Fitri Ayu, 2018).

UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut adalah penjelasan dari pembagian kategori tersebut : (Daniel Dodi, 2020)

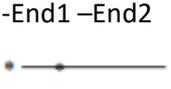
1. *Structure diagrams*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagrams* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.

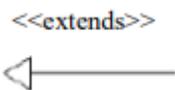
2. *Behavior diagrams*, yaitu sekumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sistem. Behavior diagrams terdiri dari, use case diagram, *activity diagram*, dan *state machine system*.
3. *Interaction diagrams*, yaitu sekumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Interaction diagrams terdiri dari, *sequence diagram*.

2.11.1 Usecase Diagram

Usecase diagram merupakan diagram yang harus dibuat pertama kali saat pemodelan perangkat lunak berorientasi objek dilakukan (Fitri Ayu, 2018). *Use case diagram* menjelaskan manfaat dari sistem jika dilihat dari sudut pandang orang yang berada diluar sistem (aktor) (Putra, 2018). *Use case diagram* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut (Julianto & Setiawan, 2019).

Tabel 2.1 Simbol Use Case Diagram

Simbol	Nama	Keterangan
	Aktor	Merupakan penggunaan dari sistem. Penamaan aktor menggunakan kata benda.
	<i>Use Case</i>	Merupakan pekerjaan yang dilakukan oleh aktor. Penamaan <i>use case</i> dengan kata kerja.
	Asosiasi	Hubungan antara aktor dengan <i>use case</i> .
	<i>Include</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>include</i> menyatakan bahwa sebelum pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu.

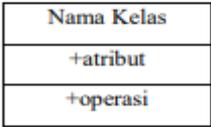
	<p><i>Extends</i></p>	<p>Hubungan antara <i>use case</i> dengan <i>use case</i>, <i>extends</i> menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, maka lakukan pekerjaan itu.</p>
---	-----------------------	---

2.11.2 Class Diagram

Class diagram dibuat setelah diagram *use case* dibuat terlebih dahulu. *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek (Raka et al., 2018).

Class diagram ini harus menjelaskan hubungan apa saja yang terjadi antara suatu objek dengan objek lainnya sehingga terbentuklah suatu sistem aplikasi (Fitri Ayu, 2018). *Class diagram* memberikan gambaran hubungan antara label-label yang ada dalam *database*. Masing-masing *class* memiliki atribut dan metode atau fungsi sesuai dengan proses yang terjadi (Putra, 2018).

Tabel 2.2 Simbol Class Diagram

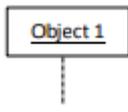
Simbol	Keterangan
<p>Kelas</p> 	<p>Kelas pada struktur sistem.</p>
<p>Antar muka/Interface</p> 	<p>Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.</p>
<p>Asosiasi / association</p> 	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i>.</p>

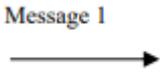
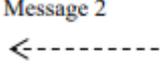
<p>Asosiasi berarah/directed association</p> 	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan <i>multiplicity</i>.</p>
<p>Generalisasi</p> 	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).</p>
<p>Kebergantungan/dependency</p> 	<p>Relasi antar kelas dengan makna ketergantungan antar kelas.</p>
<p>Agresiasi/aggregation</p> 	<p>Relasi antar kelas dengan makna semua bagian.</p>

2.11.3 Sequence Diagram

Sequence diagram menjelaskan interaksi antar objek di dalam dan di sekitar sistem berupa pesan (*message*) yang disusun dalam suatu urutan waktu yaitu urutan kejadian yang dilakukan oleh seorang aktor dalam menjalankan sistem. *Sequence diagram* biasa digunakan untuk menggambarkan scenario atau rangkaian langkah-langkah yang dilakukan sebagai response dari sebuah kegiatan untuk menghasilkan *output* tertentu (Putra, 2018).

Tabel 2.3 Simbol Sequence Diagram

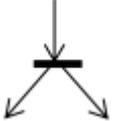
Simbol	Nama	Keterangan
	Objek/aktor	Sebuah objek yang berasal dari kelas atau dapat dinamai dengan kelasnya saja. Aktor termasuk objek. Garis putus-putus menunjukkan garis hidup suatu objek.
	Aktivasi	Menunjukkan masa hidup dari objek

	Pesan	Interaksi antara satu objek dengan objek lainnya. Objek dapat mengirimkan pesan ke objek lain. Interaksi antar objek ditunjukkan pada bagian operasi pada diagram kelas.
	return	Pesan kembalian dari komunikasi antar objek.

2.11.4 Activity Diagram

Activity diagram menggambarkan aktivitas yang dapat dilakukan oleh sistem. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas (Putra, 2018). *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem proses atau proses bisnis (Muhammad & Dame, 2018).

Tabel 2.4 Simbol *Activity Diagram*

Simbol	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awalan aktivitas.
	<i>End point</i> , merupakan akhir aktivitas.
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.

2.12 Penelitian Terdahulu

Berikut ini adalah beberapa penelitian terdahulu yang penulis gunakan sebagai acuan dalam membuat, dan menyusun penelitian ini.

No	Penulis	Judul	Hasil Penelitian
1.	Eka Yuni Rachmawati, Budi Prasetyo, Riza Arifudin (2018)	Perbandingan antara Metode <i>Bayes</i> dan Faktor Keyakinan dalam Sistem Pakar untuk Diagnosis Dini Infeksi Demam Berdarah	Memabndingkan 2 metode system pakar dengan menghasilkan performa dari metode <i>Bayes</i> sebesar 90%, dan certainty factor sebesar 93.75%
2.	Adi Sucipto, Yusra Fernando, Rohmat Indra Borman, Nisa Mahmuda (2018)	Penerapan Metode Certainty Factor Pada Diagnosa Penyakit Saraf Tulang Belakang	Penelitian tersebut berhasil mengatasi ketidakpastian dalam diagnose tulang belakang, dan menghasilkan system pakar yang hasil outputnya sesuai dengan perhitungan manual
3.	Finanta Okmayura, Noverta Effendi (2019)	Rancang Bangun Sistem Pakar untuk Identifikasi Dini Pelaku Bullying pada Remaja Menggunakan Teori Dempster-Shafer	Sistem Pakar Identifikasi Dini Pelaku Bullying pada Remaja telah berhasil dirancang dan diimplementasikan dalam bentuk sistem pakar berbasis web untuk mengidentifikasi dini pelaku bullying anak berdasarkan gejala yang dialami serta dapat memberikan solusi atas gangguan yang dialami.
4.	Muhammad Ridho Syahwana, R.Mahdalena	Sistem Pakar Untuk Mendiagnosis Penyakit Tuberculosis Menggunakan Metode	Fokus dari penelitian ini adalah mendiagnosis penyakit tuberculosis, dan berhasil mendiagnosis penyakit tuberculosis dengan <i>Bayes</i>

	Simanjourang (2020)	<i>Bayes</i> Pada Puskesmas Petumbukan	
5.	Juna Eska, Hambali (2021)	Sistem Pakar Metode Certainty Factor Dalam Diagnosa Penyakit Kanker Kelenjar Getah Bening Pada RSUD H.Abdul Manan Simatupang	Penelitian ini mampu mengatasi ketidakpastian diagnose dengan certainty factor dengan performa 80%

Adapun dari 5 penelitian tersebut dijadikan referensi bagi penulis dalam membuat penelitian terbaru mengenai diagnosa kasus *bullying*, beserta jenisnya, dan sekaligus mendapatkan solusi untuk menangani kasus *bullying* tersebut berdasarkan jenis kasusnya. Oleh karena itu penulis membangun sebuah sistem berbasis web yang dapat menyelesaikan permasalahan dalam mendiagnosa kasus *bullying* secara akurat. penulis tertarik melakukan penelitian berjudul “ Sistem Pakar Pendekatan Psikologis Terhadap Kasus *Bullying* Menggunakan Metode *Certainty Factor & Bayes* Berbasis Web” dengan melakukan perbandingan antara dua metode yakni *Certainty Factor* dan *Bayes* untuk menentukan metode mana yang memiliki performa yang lebih tinggi.