

BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

Halaman ini membahas konsep dasar dari literatur pengetahuan ilmiah tentang teori dan fakta yang digunakan dalam proses analitis Implementasi perancangan sistem pada topik masalah yang diajukan.

2.1.1. Sistem Pakar

Sistem pakar merupakan sebuah program komputer yang dirancang untuk meniru atau meniru pengetahuan serta keahlian seorang ahli dalam bidang tertentu. Sistem ini juga bertujuan untuk menyelesaikan masalah berdasarkan bidang keahlian yang dimilikinya (Aldo, Nur, Hulqi, Lanyak, & Hikmah, 2022).

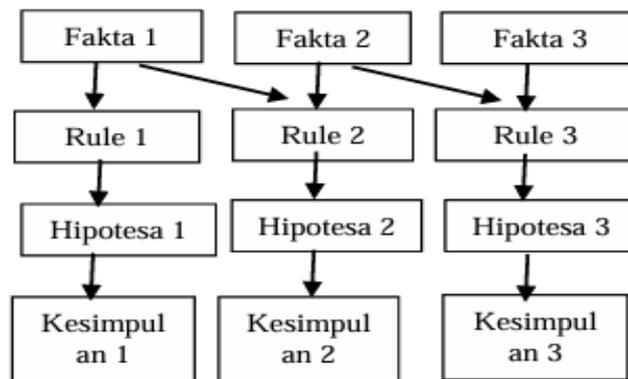
Menurut (Hayadi, 2018) sistem pakar merupakan program kecerdasan buatan (AI) yang dilengkapi dengan basis pengetahuan (*Knowledge Base*). Basis pengetahuan ini diperoleh dari pengalaman atau keahlian seorang pakar dalam menyelesaikan masalah di bidang tertentu. Sistem ini juga menggunakan mesin inferensi (*Inference Engine*) yang bertugas melakukan penalaran atau pelacakan berdasarkan fakta-fakta dan aturan-aturan yang ada dalam basis pengetahuan. Setelah proses tersebut, sistem akan menghasilkan kesimpulan yang sesuai.

2.1.2. Forward Chaining

Forward chaining merupakan salah satu teknik inferensi yang berawal dari fakta-fakta yang ada untuk menarik kesimpulan berdasarkan data (Ariska & Yulianton, 2023). Metode ini, juga dikenal sebagai urutan ke depan, dimulai dengan beberapa fakta yang telah diketahui. Untuk memperoleh informasi baru, sistem akan melakukan pencarian dengan menerapkan aturan-aturan yang premisnya sesuai dengan fakta yang sudah ada. Forward chaining adalah teknik pencarian dimulai dari fakta yang diketahui, kemudian mencocokkan fakta atau pernyataan dimulai dari bagian sebelah kiri IF dari rules IF-THEN. Bila fakta cocok dibagian

IF, maka rule akan tersebut dieksekusi. Apabila sebuah rule dieksekusi, maka setiap fakta baru (bagian THEN) 24 ditambahkan ke dalam database. Rule akan dimulai dari teratas dan setiap rule hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi rule yang bisa dieksekusi. Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis (Rosnelly, 2012).

Maka berdasarkan metode ini langkah langkah yang diambil:



Gambar 2. 1 Gambaran kerja Forward Chaining

R1 : IF A and C, THEN B

R2 : IF D and C, THEN F

R3 : IF B and E, THEN F

R4 : IF B, THEN C

R5 : IF F, THEN G

2.1.3. Autisme

Autis adalah kelainan perkembangan saraf yang didefinisikan oleh standar diagnostik yang termasuk kurangnya kemampuan untuk berkomunikasi sosial dan interaksi sosial, dan adanya kecenderungan perilaku, preferensi, atau tindakan berulang yang terbatas waktunya mampu bertahan sepanjang hidup dari kata "*auto*" yang mengacu pada diri sendiri (Pratama et al., 2019a). Autisme sering diartikan dengan cacat mental, kesulitan melakukan gerakan kompleks, kesulitan dalam mengarahkan perhatian (fokus anak), dan lambatnya perkembangan kepribadian pada anak (Ginting, Zarlis, & Rosnelly, 2021).

Penelitian sebelumnya menunjukkan pentingnya deteksi dini terhadap gejala autisme untuk memberikan intervensi yang lebih efektif pada anak (Lord et al., 2020). Dengan deteksi dini orang tua akan dapat memahami kebutuhan khusus anak dan memberikan dukungan yang sesuai dengan perkembangan dan pertumbuhan anaknya.

A. Tanda dan Gejala Autisme

Gejala autisme biasanya dibagi menjadi dua kategori utama: gangguan komunikasi dan interaksi sosial dan gangguan pola perilaku (Pittara, 2023). Gejala ini bervariasi dari satu orang ke orang lain dan dapat muncul pada usia dini.

a. Gangguan Komunikasi dan Interaksi Sosial

Anak-anak dengan autisme sering menghadapi masalah dalam berinteraksi dengan orang lain. Beberapa tanda yang dapat dikenali adalah:

1. Anak tidak menjawab saat namanya dipanggil: Meskipun pendengarannya normal, dia tidak menunjukkan reaksi.
2. Kurangnya ekspresi emosi: Anak-anak mungkin tidak menunjukkan perasaan mereka atau mungkin tidak sensitif terhadap perasaan orang lain.
3. Ketidakmampuan untuk memulai atau melanjutkan percakapan: Anak mungkin tidak dapat meminta sesuatu atau terlibat dalam diskusi.

4. Ekolalia: Sering mengulang kata-kata tanpa memahami artinya, yang termasuk perilaku latah, adalah contoh ekolalia.
 5. Menghindari kontak mata: Anak cenderung tidak melihat satu sama lain dan tidak mengekspresikan ekspresi wajah mereka.
 6. Ketidakmampuan untuk berbagi pengalaman: Anak tidak suka bermain atau berbagi pengalaman dengan teman sebaya mereka.
- b. Gangguan Pola Perilaku
- Anak-anak dengan autisme juga menunjukkan pola perilaku yang terbatas dan repetitif, seperti:
1. Sensitivitas terhadap rangsangan sensorik: Anak mungkin sangat sensitif terhadap suara, cahaya, atau sentuhan, tetapi tidak merespons rasa sakit.
 2. Pola aktivitas yang kaku: Anak mungkin marah jika rutinitasnya terganggu atau jika ada perubahan dalam lingkungan.
 3. Gerakan repetitif: Seperti mengibaskan tangan, berputar, atau berjalan dengan berjinjit.
 4. Minat terbatas: Anak-anak dengan autisme mungkin

Menurut (Association & others, 2000), tingkat keparahan autis dibagi menjadi 3 kategori yaitu sebagai berikut:

Tabel 2. 1 Tanda dan Gejala Autisme

Level keparahan autis	Komunikasi Sosial	Minat Tertentu
Level 3 “Memerlukan bimbingan yang sangat intensif”	Kapasitas untuk komunikasi verbal dan nonverbal, perilaku, dan kontak sosial telah rusak parah.	Disibukkan dengan kesendirian fokus pada pengulangan rutinitas dan perilaku, sulit fokus, terlihat sedih saat rutinitas berubah, dan sulit diarahkan jika terpaku pada satu hal.
Level 2 “membutuhkan	Ditandai dengan kurangnya kemampuan	Rutinitas dan tindakan yang berulang. Ketika

bimbingan intensif”	komunikasi baik verbal maupun non verbal. Hubungan sosial terbatas dan tanggap aneh pada suatu tawaran bantuan orang lain.	rutinitas terganggu, ada tanda-tanda kecemasan atau ketidakpuasan yang jelas, sulit untuk bergerak ketika seseorang berkonsentrasi pada sesuatu.
Level 1 ”membutuhkan bimbingan”	Ketika kurangnya keterampilan komunikasi anak lebih jelas terlihat. Kesulitan memulai hubungan sosial atau mendeskripsikan sesuatu dengan cara yang membuat mereka terlihat tidak tertarik.	Perilaku berulang sangat mengganggu kemampuan seseorang. Menolak untuk membantu orang lain, rendahnya kemandirian.

B. Jenis - Jenis Autisme

Menurut penelitian (Pratama, Astuti, & Buntoro, 2019) penderita autisme dapat dibedakan menjadi beberapa jenis berdasarkan waktu gangguan perkembangan muncul, diantaranya;

- a. Autis dari lahir: Anak-anak sudah menunjukkan perbedaan dengan anak-anak biasa dari lahir.
- b. Autis Regresif: Perkembangan anak sejak lahir normal seperti anak normal, tetapi setelah 1,5 hingga 2 tahun ada keterlambatan dalam perkembangan.
- c. Autis Persepsi: Karena kelainan ini sudah ada sejak lahir, autisme ini dianggap autisme asli atau disebut autisme internal (*endogenous*).
- d. Autis yang Reaktif : Penderita autis melakukan gerakan tertentu yang berulang, seringkali kejang. Jenis autis ini dapat diamati pada anak-anak usia enam sampai tujuh tahun ketika mereka mudah terpengaruh oleh lingkungan sekitar.
- e. Individu dengan autisme muncul kemudian: Kelainan ini biasanya diketahui setelah anak berusia kurang lebih dari 7 tahun, sehingga

sulit untuk mengajarkan anak untuk mengubah perilakunya karena sudah melekat. Ini ditambah dengan pengalaman baru dan mungkin karena kelainan jaringan otak janin.

C. Karakteristik *Autism Spectrum Disorder* (ASD)

Kesulitan bicara, bahasa, dan perilaku adalah tiga ciri dasar anak dengan gangguan spektrum autisme (ASD). Anak-anak dengan ASD juga menunjukkan ciri-ciri tambahan seperti gangguan kognitif, persepsi sensorik, keterampilan motorik, emosi, perilaku impulsif dan kekerasan, serta masalah tidur dan makan (Utami & Laksono, 2024).

2.1.4. Aplikasi Mobile

Menurut (Buyens, 2001), istilah "aplikasi mobile" berasal dari kata *application* dan *mobile*. *Application* berarti penerapan, penggunaan, atau lamaran, yang secara istilah merujuk pada program siap pakai yang dirancang untuk menjalankan fungsi tertentu bagi pengguna atau aplikasi lain. Sementara itu, *mobile* mengacu pada perpindahan dari satu tempat ke tempat lain, yang berarti bergerak atau berpindah. Menurut (Purnama, 2010), aplikasi mobile adalah aplikasi yang beroperasi pada perangkat mobile (*mobile device*). Dengan adanya aplikasi mobile, pengguna dapat dengan mudah melakukan berbagai aktivitas, seperti hiburan, berjualan, belajar, mengerjakan tugas kantor, browsing, dan lain sebagainya.

2.1.5. Website

Website atau *World Wide Web* (www) merupakan sekumpulan halaman dalam suatu domain yang menyediakan berbagai informasi yang dapat diakses dan dilihat oleh pengguna internet melalui mesin pencari. Informasi yang ditampilkan pada situs web biasanya mencakup gambar, ilustrasi, video, dan teks untuk berbagai keperluan (Yuhefizar & Hidayat, 2009).

Fungsi utama website adalah menyediakan informasi. Di era modern ini, situs web juga berperan sebagai platform pemasaran bagi pelaku bisnis untuk menjangkau calon pelanggan secara lebih luas melalui internet. Bagi individu, website sering dimanfaatkan sebagai alat komunikasi, media penyebaran informasi,

serta sarana untuk menjalankan bisnis online.

2.1.6. Java

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh James Gosling di *Sun Microsystems* dan pertama kali dirilis pada tahun 1995. Sesuai dengan filosofi "*write once, run anywhere*" (WORA), bahasa ini memungkinkan kode Java untuk dijalankan di berbagai platform tanpa perlu dikompilasi ulang (Jati et al., 2024). Ini berarti bahwa program yang ditulis dalam Java dapat berjalan tanpa dimodifikasi di setiap platform yang mendukung *Java Virtual Machine* (JVM).

Karena berorientasi objek, *multiplatform*, dan serbaguna (*multiplatform*), bahasa pemrograman Java Netbeans memudahkan desain, pengembangan, dan pengorganisasian kesalahan secara cepat, tepat, mudah, dan terorganisir. Ini menjadikannya sangat mudah bagi pemrogram untuk membangun aplikasi mereka.

2.1.7. MySQL

MySQL adalah sistem manajemen database SQL yang paling populer dan bersifat open source saat ini. Mysql adalah salah satu perangkat lunak sistem manajemen basis data (data base) SQL atau sering disebut sebagai DBMS, dan sistem database MySQL mendukung beberapa fitur seperti multi user. MySQL memiliki fitur multithread dan multiuser serta mendukung sistem jaringan, membedakannya dari basis data konvensional seperti .dat, .dbf, dan .mdb. MySQL tersedia secara gratis di bawah lisensi GNU *General Public License* (GPL), tetapi bagi mereka yang menginginkannya, ada versi komersial. MySQL adalah perangkat lunak database server yang banyak digunakan dalam pengoperasian basis data, seperti yang ditunjukkan di atas Kelebihan MySQL (Suhartono & Zahraen, 2022).

Beberapa manfaat serta kelebihan dari MySQL adalah sebagai berikut (Utami & Laksono, 2024):

1. Opensource yang memungkinkan mengembangkan dimasa mendatang.
2. Data disesuaikan dengan pemrograman yang berbeda.
3. *Safety* atau keamanan yang cukup.

4. Sangat mudah digunakan dan di pelajari.
5. Memiliki komunitas pengguna MySQL (*group*)
6. Kemampuan lintas Platform dapat berfungsi di berbagai OS.
7. Multiuser yang memungkinkan beberapa pengguna menggunakan MySQL secara bersamaan tanpa menemui kendala.

2.1.8. Flowchart

Flowchart merupakan representasi grafis yang menggambarkan langkah-langkah dan urutan prosedur dalam suatu program. Alat ini memecah masalah menjadi bagian-bagian yang lebih kecil dan membantu analisis dalam mengevaluasi berbagai alternatif operasional (Ridlo, 2017).

Flowchart memegang peranan penting dalam menentukan langkah-langkah atau fungsionalitas suatu proyek pembuatan program, terutama ketika melibatkan banyak orang secara bersamaan. Selain itu, diagram alur ini memudahkan pemahaman terhadap proses program, membuatnya lebih ringkas, dan mengurangi risiko kesalahpahaman. Dalam konteks pemrograman, *flowchart* juga berfungsi sebagai jembatan yang menghubungkan kebutuhan teknis dan non-teknis (Adelia & Setiawan, 2011).

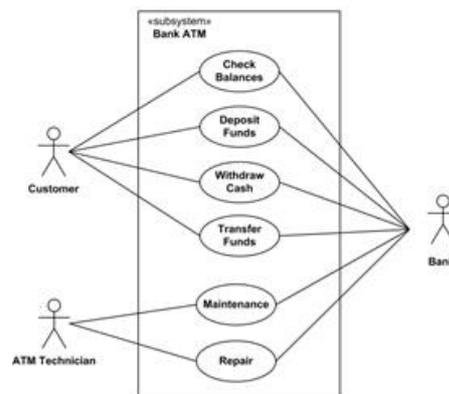
2.1.9. Use case Diagram

Use Case Diagram adalah salah satu jenis diagram *Unified Modeling* pengguna (user) dan sistem dalam suatu lingkungan tertentu (Pranoto, Sutiono, Nasution, & others, 2024). *Use case* diagram digunakan sebagai representasi fungsionalitas sistem secara visual, sehingga memudahkan pemahaman dan komunikasi antara pengembang perangkat lunak dengan klien atau pengguna. Dalam *Use Case Diagram* terdiri dari beberapa konsep utama yaitu:

- a. Aktor: Pihak atau entitas yang terlibat dalam interaksi dengan sistem. Aktor dapat berupa pengguna, sistem lain, atau entitas lain yang terkait dengan sistem.
- b. *Use Case*: *Use case* merepresentasikan fungsionalitas atau aktivitas yang dapat dilakukan oleh sistem. *Use case* biasanya digambarkan sebagai elips

pada *use case* diagram.

- c. Relasi antar Aktor dan *Use Case*: Hubungan antar aktor dengan *use case* digambarkan dengan garis. Garis tersebut menggambarkan bagaimana aktor berinteraksi dengan *use case* dalam sistem.
- d. Relasi antar *use case*: Hubungan antar *use case* digambarkan dengan panah. Panah tersebut menggambarkan bagaimana satu *use case* terikat dengan *use case* lainnya dalam sistem.
- e. *Include* dan *Extend*: Merupakan dua jenis hubungan antar *use case* yang digunakan untuk menggambarkan keterikatan antar *use case* dalam sistem. *Include* digunakan untuk menggambarkan *use case* yang terkait secara langsung dengan *use case* utama, sedangkan *Extend* digunakan untuk menggambarkan *use case* yang hanya dilakukan dalam kondisi tertentu.



An example of use case diagram for Bank ATM subsystem - top level use cases.

Sumber: socs.binus.ac.id

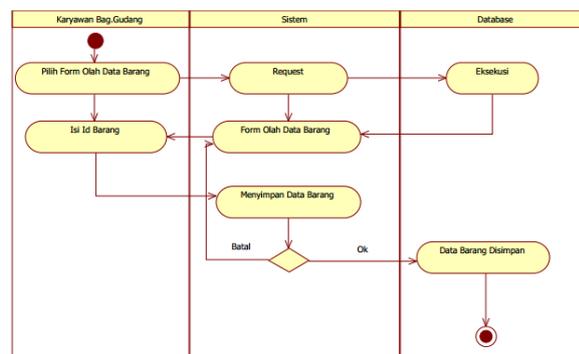
Gambar 2. 2 Contoh *Use Case* Diagram

2.1.10. Activity Diagram

Activity diagram adalah jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memodelkan berbagai proses atau alur kerja dalam suatu sistem. Diagram ini memvisualisasikan bagaimana sistem beroperasi, menampilkan urutan langkah-langkah atau aktivitas dari awal hingga akhir. Berbeda dengan diagram alir (*flowchart*), *activity* diagram lebih fokus pada pemodelan proses bisnis atau alur kerja dalam perangkat lunak dan biasanya

digambarkan secara vertikal untuk menunjukkan tahapan yang berurutan.

Menurut (Shalahuddin & Rosa, 2013), *activity* diagram adalah alat yang berguna untuk merepresentasikan *workflow* atau aliran kerja dalam suatu sistem atau proses bisnis. *Workflow* ini mencakup serangkaian aktivitas yang saling terkait dan melibatkan berbagai komponen dalam sistem, baik manusia, perangkat lunak, maupun perangkat keras. Dengan *activity* diagram, pengembang dapat memahami, menganalisis, dan mengkomunikasikan alur proses, termasuk keputusan-keputusan yang mungkin terjadi selama alur kerja.



Sumber: tutorialkampus.com

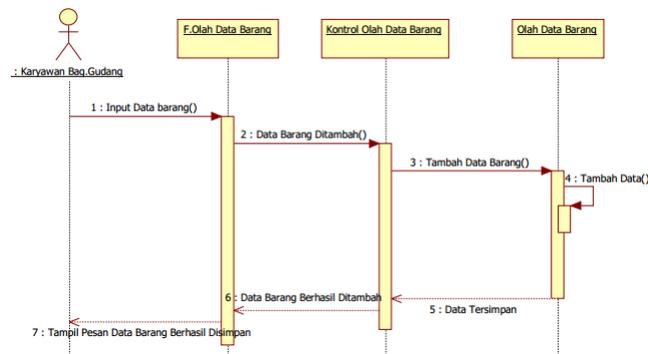
Gambar 2. 3 Contoh Activity Diagram

2.1.11. Sequence Diagram

Menurut (Hendini, 2016), *sequence* diagram adalah diagram yang menggambarkan perilaku objek dalam suatu *use case* dengan menampilkan waktu hidup (*lifeline*) objek serta pesan yang dikirim dan diterima antar objek. Diagram ini memvisualisasikan interaksi antara objek-objek dalam sistem secara berurutan berdasarkan waktu, sehingga memudahkan pemahaman tentang alur komunikasi dan proses yang terjadi.

Selain itu, (Maxim & Pressman, 2014) menjelaskan bahwa *sequence* diagram menunjukkan pemanggilan metode melalui panah horizontal yang mengarah dari objek pemanggil ke objek yang dipanggil. Diagram ini juga dapat menampilkan detail seperti parameter, tipe data, dan tipe kembalian (*return type*) dari metode yang dipanggil, meskipun informasi tersebut bersifat opsional dan dapat disesuaikan sesuai kebutuhan. Dengan demikian, *sequence* diagram menjadi alat

yang efektif untuk memodelkan interaksi dinamis antarobjek dalam sistem.



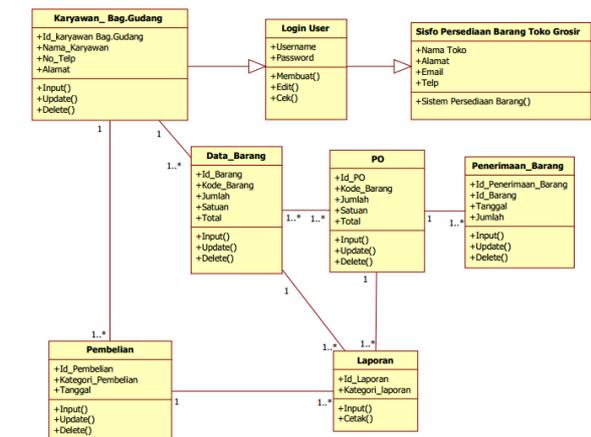
Sumber: tutorialkampus.com

Gambar 2. 4 Contoh *Sequence Diagram*

2.1.12. Class Diagram

Class diagram merupakan salah satu jenis diagram struktur statis dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan struktur suatu sistem. Diagram ini menampilkan hubungan antara kelas-kelas dalam sistem, termasuk atribut, metode, dan objek yang terkait. Sebagai bagian dari diagram struktur, class diagram berfokus pada komponen-komponen yang membentuk sistem dan bagaimana komponen-komponen tersebut saling berinteraksi.

Menurut (Shalahuddin & Rosa, 2013), *class* diagram berfungsi untuk mendefinisikan struktur sistem dengan cara memvisualisasikan kelas-kelas yang dibutuhkan dalam pembangunan sistem. Setiap kelas dalam diagram ini memiliki atribut yang merepresentasikan data atau properti, serta metode operasi yang menggambarkan perilaku atau fungsi yang dapat dilakukan oleh kelas tersebut. Dengan demikian, *class* diagram tidak hanya menunjukkan komponen-komponen sistem, tetapi juga menjelaskan hubungan dan tanggung jawab masing-masing kelas dalam sistem yang dimodelkan.



Sumber: tutorialkampus.com

Gambar 2. 5 Contoh Class Diagram

2.2. Teori Perancangan Basis Data

Pada sub bab berikut akan membahas mengenai teori basis data yang akan digunakan berdasarkan penelitian penulis.

2.2.1. Basis Data

Basis data adalah kumpulan data yang saling berhubungan satu sama lain. Sebagai salah satu komponen penting dalam sistem informasi, basis data atau database berfungsi sebagai fondasi penyedia informasi bagi penggunanya (Ramadhani & others, 2024). Pada dasarnya, basis data merupakan media penyimpanan data yang dirancang untuk memungkinkan pengaksesan data secara mudah, cepat, dan efisien. Dengan demikian, basis data memainkan peran kunci dalam mendukung pengelolaan dan pemanfaatan informasi dalam berbagai konteks, baik bisnis, akademik, maupun organisasi lainnya.

2.2.2. Tujuan dan Manfaat Basis Data

Tujuan utama dalam database adalah untuk mempermudah dan mempercepat pengambilan data yang dicari. Basis data digunakan untuk tujuan sebagai berikut :

1. Kecepatan dan kemudahan (*Speed*)

Database memungkinkan Anda untuk menyimpan data, membuat perubahan (manipulasi), dan menyegarkan data dengan cepat dan mudah.

2. Efisien ruang penyimpanan (*Space*)

Dengan Database data dapat mengurangi jumlah iterasi data dengan melakukan penekanan jumlah pengulangan data dengan menerapkan sejumlah pengkodean.

3. Keakuratan (*Acuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data dengan menegakkan aturan atau batasan tipe data dapat diterapkan ke database untuk membantu mendeteksi ketidakakuratan input atau penyimpanan.

4. Keamanan (*Security*)

Dalam sejumlah sistem (aplikasi) pengelola database tidak menerapkan aspek keamanan dalam penggunaan database. Tetapi untuk sistem yang besar dan serius, pertimbangan keamanan juga berlaku. Ini memungkinkan Anda untuk menentukan siapa yang diizinkan menggunakan database dan jenis operasi apa yang diizinkan.

5. Terpeliharanya keselarasan data (*Consitant*)

Apabila ada perubahan data pada aplikasi yang berbeda maka secara otomatis perubahan itu berlaku untuk keseluruhan

6. Data dapat dipakai secara bersama (*shared*)

Data dapat dipakai secara bersama-sama oleh beberapa program aplikasi (secara *batch* maupun online) pada saat bersamaan.

7. Dapat diterapkan standarisasi (*standardization*)

Dengan adanya pengontrolan yang terpusat maka DBA dapat menerapkan standarisasi data yang disimpan sehingga memudahkan pemakaian, pengiriman maupun pertukaran data

2.2.3. Entity Relationship Diagram (ERD)

ERD (Entity Relationship Diagram) adalah diagram yang digunakan untuk merancang database dengan mendetailkan hubungan antara objek atau entitas beserta atributnya. Menurut (Al Fatta & others, 2009) , ERD merupakan model jaringan yang memanfaatkan susunan abstrak dari data yang tersimpan dalam suatu

sistem. Sementara itu, (Shalahuddin & Rosa, 2013) menjelaskan bahwa ERD saat ini sedang dikembangkan dalam bidang matematika, khususnya berdasarkan teori himpunan.

Dari penjelasan tersebut, dapat disimpulkan bahwa ERD (*Entity Relationship Diagram*) adalah model jaringan yang menggunakan susunan abstrak dari data yang tersimpan dalam suatu sistem. Selain itu, ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika, menjadikannya alat yang efektif untuk merancang dan memvisualisasikan struktur database serta hubungan antar-entitasnya.

2.2.4. Logical Record Structure (LRS)

Menurut (Ladjamudin, 2013) , "Transformasi ERD/LRS sering disebut sebagai pemetaan ERD ke basis data relasional." Sementara itu, (Hasugian & Shidiq, 2012) menjelaskan, "*Logical Record Structure (LRS)* adalah model sistem yang dijelaskan oleh diagram ER yang mengikuti pola dan aturan pemodelan tertentu terkait dengan transformasi ke LRS."

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *Logical Record Structure (LRS)* merupakan hasil penerjemahan pemetaan ERD ke dalam basis data relasional, yang digambarkan melalui diagram ER dengan mengikuti pola dan konvensi pemodelan tertentu.

2.3. Teori Pengujian Sistem

Pada sub bab berikut akan membahas tentang teori pengujian sistem yang digunakan dalam pengujian sistem pakar untuk Pengenalan Gejala Autisme Pada Anak Menggunakan Metode *Forward Chaining* Berbasis Android :

2.3.1. Black Box Testing

Black box testing, atau yang juga dikenal sebagai *Behavioral Testing*, adalah metode pengujian yang dilakukan pada tahap akhir pengembangan perangkat lunak. Tujuannya adalah untuk memastikan bahwa perangkat lunak berfungsi dengan benar dengan memeriksa hasil input dan output tanpa perlu mengetahui struktur internal kode.

Pengujian ini berfokus pada kebutuhan fungsional perangkat lunak. Teknik *black box testing* bukanlah pengganti dari metode *white box testing*, melainkan pendekatan yang saling melengkapi. *Black box testing* dapat mengungkap jenis kesalahan yang berbeda dari yang ditemukan melalui *white box testing*. Menurut (Maxim & Pressman, 2014), teknik *black box testing* memungkinkan penentuan serangkaian kondisi input yang dapat sepenuhnya menguji semua persyaratan fungsional suatu program. Dengan kata lain, metode ini membantu memastikan bahwa perangkat lunak memenuhi kebutuhan pengguna dari segi fungsionalitas tanpa mempertimbangkan detail implementasi kode.

2.3.2. White Box Testing

White box testing adalah metode pengujian perangkat lunak yang melibatkan analisis dan pemeriksaan struktur internal serta kode program. Berbeda dengan *black box testing* yang hanya berfokus pada hasil input dan output tanpa mempertimbangkan kode, *white box testing* memperhatikan aliran input-output perangkat lunak secara mendalam. Untuk melakukan pengujian ini, penguji harus memiliki pemahaman yang baik tentang kode program yang sedang diuji.

Menurut (Maxim & Pressman, 2014), metode *white box testing* memungkinkan penguji untuk menjamin bahwa semua aspek internal program telah diuji secara menyeluruh, sehingga meningkatkan kualitas dan keandalan perangkat lunak. Pendekatan ini melengkapi *black box testing* dengan fokus pada detail implementasi kode, sementara *black box testing* berfokus pada fungsionalitas dari perspektif pengguna.

2.4. Penelitian Terdahulu

Dalam penyusunan proposal skripsi ini, penulis sedikit banyak terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan latar belakang masalah pada skripsi ini. Berikut ini penelitian terdahulu yang berhubungan dengan penulisan ini antara lain :

Tabel 2. 2 Penelitian Terdahulu

No	Peneliti	Judul	Tahun	Tujuan Penelitian
1.	Ricko Irawan, Agus Raharjo, Aris Mulyono, dan Sesaria Nisa Afifi	Aplikasi Praktis dan Mudah Mengenali Gejala Anak Autisme Sejak Dini	2022	Pengenalan aplikasi SDA-03 untuk mengenalkan ciri-ciri anak yang mengalami gejala dini autisme pada orang tua atau keluarga yang putra/putrinya terdampak penyakit autisme (Irawan, Raharjo, Mulyono, & Afifi, 2022)
2.	Fadhila Purboningrum, Prof. Dr. Eng. Fitri Utaminingrum, S.T., M.T., dan Fais Al Huda, S.Kom., M.Kom.	Sistem Deteksi Dini <i>Autism Spectrum Disorder (ASD)</i> Berbasis <i>Face Recognition</i> Menggunakan Metode <i>Transfer Learning Resnet50</i>	2024	Demonstrasikan potensi <i>face recognition</i> dan model <i>deep learning</i> dalam membantu proses diagnosis <i>Autism Spectrum Disorder (ASD)</i> dengan akurasi tinggi, serta untuk mendukung intervensi dini dan meningkatkan kesadaran terhadap ASD (Purboningrum et al., 2024).
3.	Atika Faizah	Perancangan <i>Prototype</i> Aplikasi Untuk Pendamping Anak Penyandang Autisme	2023	merancang <i>prototype</i> aplikasi berbasis mobile yang dapat membantu

		Dengan Pendekatan <i>Participatory Design</i>		pendamping dalam memenuhi kebutuhan anak autis dan mengetahui tingkat <i>usability</i> dari <i>prototype</i> yang telah dirancang (Faizah, 2023).
4.	Evans Fuad, Rabiah Aminullah, Soni, dan Yoze Rizki	<i>Expert System</i> Diagnosa Gangguan Autisme Secara Dini Pada Anak dengan Metode <i>Forward</i> <i>Chaining</i>	2022	merancang dan membangun suatu <i>expert system</i> (sistem pakar) diagnosa gangguan Autisme pada anak usia dini berbasis web dengan Metode <i>Forward Chaining</i> untuk memberikan informasi dan solusi bagi orang tua tentang jenis gangguan autisme pada anak usia dini berdasarkan gejala yang dimasukkan (Fuad et al., 2022).
5.	Hindariyani Indri Dwita Utami, Sularso Budi Laksono	Rancang Bangun Aplikasi Skala Penilaian Perilaku Untuk Membantu Diagnosa <i>Autism</i> <i>Spectrum Disorder</i> Pada Anak Berbasis Web	2024	Membantu diagnosa gejala autis dan tingkat keparahannya dengan menggunakan logika <i>fuzzy</i> (Utami & Laksono, 2024).