

BAB II

LANDASAN TEORI

2.1. Penelitian Terdahulu

Dari penelitian terdahulu menjadi sebuah landasan bagi penulis untuk dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Berdasarkan penelitian terdahulu yang dilakukan oleh Irwan Syahruji, yang membahas tentang Sistem Informasi Pengolahan Data Pada Asrama Mahasiswa Kalimantan Selatan Demang Lehman (AMKSDL) Berbasis Web, pada penelitian tersebut Irwan Syahruji melakukan pengamatan langsung dan wawancara kepada pihak yang terkait perihal permasalahan pengolahan data di asrama, dengan kata lain penelitian yang dilakukan oleh irwan syahruji menggunakan metode kualitatif. Dan model yang digunakan dalam pengembangan menggunakan metode *waterfall*. Tujuan penelitian yang dilakukan oleh Irwan Syahruji untuk pengembangan membuat sistem pengelolaan pendaftaran, pembayaran, layanan pengaduan dan feedback[2].

Penelitian selanjutnya yaitu telah dilakukan oleh sa'idatul Uswah, yang membahas mengenai Sistem Informasi Pembayaran Asrama Terintegrasi *Payment Gateway* Berbasis Web Studi Kasus: Asrama Xi Muzamzmah chosiyah Ppdu Jombang, Metode pengumpulan data yang digunakan pada penelitian tersebut menggunakan metode lapangan, bisa juga disebut metode kualitatif. dan model yang digunakan dalam pengembangan menggunakan metode *waterfall*. Tujuan dari penelitian tersebut adalah memberikan kemudahan dalam proses

pembayaran yang akurat, memberi kemudahan secara cepat dalam proses pembayaran, ketepatan akurasi data dan meningkatkan efektivitas pelayanan kepada santri maupun wali santri[3].

Meskipun kedua penelitian tersebut memiliki persamaan dalam metode pengumpulan data (kualitatif) dan model pengembangan sistem (*waterfall*), fokus subjek penelitian dan tujuan penelitian keduanya berbeda. Penelitian Sa'idatul Uswah lebih berfokus pada sistem pembayaran asrama, sementara penelitian Irwan Syahrui lebih berfokus pada pengolahan data di asrama mahasiswa.

2.2. Teori Pendukung

Teori pendukung adalah konsep-konsep atau kerangka pemikiran yang digunakan dalam penelitian. Tujuan teori pendukung adalah memberikan landasan dan pemahaman yang mendukung penelitian yang dilakukan oleh penulis. Teori pendukung membantu penulis dalam menganalisis fenomena yang diteliti, menjelaskan hubungan antara variabel-variabel yang ada, dan merumuskan hipotesis atau pertanyaan penelitian. Dengan menggunakan teori pendukung, penulis dapat memperluas pengetahuan yang ada, mengidentifikasi kekosongan dalam penelitian sebelumnya, dan memberikan dasar pemikiran yang kuat dalam penelitian yang dilakukan.

2.2.1 Sistem Informasi

Menurut Krismaji yang disitasi oleh Reza Sangga Rasefta dan Shinta Esabella menjelaskan bahwa sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan

didistribusikan kepada pemakai[4]. Sedangkan menurut Harold Koontz dan Heinz Weihrich: "Sistem adalah kumpulan elemen yang saling berinteraksi dan saling mempengaruhi satu sama lain untuk mencapai tujuan yang diinginkan"

dari kedua pendapat tersebut adalah bahwa sistem adalah suatu kesatuan yang terdiri dari elemen-elemen yang saling berinteraksi dan mempengaruhi satu sama lain untuk mencapai tujuan tertentu. Dalam kedua definisi tersebut, pentingnya interaksi dan ketergantungan antara elemen-elemen dalam sistem ditekankan. Sistem tidak hanya terdiri dari elemen-elemen yang terpisah, tetapi juga melibatkan hubungan dan ketergantungan antara elemen-elemen tersebut. Tujuan yang ingin dicapai juga menjadi faktor penting dalam pemahaman tentang sistem, di mana elemen-elemen dalam sistem bekerja bersama untuk mencapai tujuan yang diinginkan.

Adapun definisi informasi menurut Sri Mulyani yang disitasi oleh Agus Ramdhani Nugraha dan Gati Pramukasari, adalah data yang sudah diolah yang ditujukan untuk seseorang, organisasi, ataupun siapa saja yang membutuhkannya. Informasi akan menjadi berguna apabila objek yang menerima informasi membutuhkan informasi tersebut [5].

Berdasarkan pengertian di atas, dapat disimpulkan bahwa informasi merupakan hasil pengolahan data yang memberikan nilai tambah bagi penerima informasi. Informasi memiliki peran penting sebagai alat bantu dalam pengambilan keputusan. Dengan adanya informasi yang terorganisir, objek yang menerimanya dapat menggunakan informasi tersebut sebagai landasan yang kuat untuk mengambil keputusan yang tepat dan efektif.

Menurut Krismaji yang disitasi oleh Reza Sangga Rasefta dan Shinta Esabella, sistem informasi adalah sebuah rangkaian prosedur formal di mana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai[6].

Di sisi lain, Agung Soetedjo dan Rangga Sidik menjelaskan bahwa sistem informasi merupakan sistem yang bekerja secara berurutan, saling berkoordinasi satu sama lain guna mengolah data dan menghasilkan informasi yang berguna [4].

Titis Prasetyo dan Rani Puspita Dhaniawaty mendefinisikan sistem informasi sebagai komponen dari sistem yang saling berelasi sehingga dapat mengorganisir data, mengolah data, dan menghasilkan informasi atau makna yang bermanfaat dan dipahami oleh penerimanya [7].

Berdasarkan definisi-definisi tersebut, dapat disimpulkan bahwa sistem informasi merupakan sebuah sistem yang saling berelasi dan berkoordinasi untuk mengolah data menjadi informasi yang dapat dimanfaatkan oleh penerima informasi.

2.2.2 Pembayaran

Pembayaran merupakan proses dimana penghuni asrama melakukan pembayaran iuran secara tepat waktu dan sesuai dengan ketentuan yang berlaku. Proses ini melibatkan identifikasi jumlah iuran yang harus dibayarkan oleh setiap penghuni, pemberitahuan atau tagihan pembayaran kepada penghuni, serta metode pembayaran yang dapat digunakan seperti transfer bank atau pembayaran tunai. Pembayaran yang dilakukan oleh penghuni kemudian akan diverifikasi dan dicatat oleh pengurus asrama untuk memastikan bahwa pembayaran iuran telah dilakukan.

2.2.3 Penagihan

Penagihan adalah proses yang dilakukan oleh pengurus asrama untuk memastikan penghuni asrama membayar iuran secara tepat waktu. Proses ini meliputi pengumpulan dan pemrosesan pembayaran dari penghuni asrama. Pengurus asrama mengidentifikasi jumlah iuran yang harus dibayarkan oleh setiap penghuni, kemudian melakukan tagihan kepada penghuni dengan rincian jumlah yang harus dibayarkan dan batas waktu pembayaran. Tujuan dari proses penagihan ini adalah untuk memastikan bahwa penghuni asrama membayar iuran sesuai dengan ketentuan yang berlaku.

2.2.4 Laporan

Laporan adalah hasil dari pengumpulan dan pengolahan data pembayaran dari penghuni asrama dan pengeluaran asrama menjadi informasi yang berguna. Laporan ini dapat mencakup total pendapatan dari pembayaran iuran asrama, rincian pengeluaran, dan saldo keuangan asrama. Proses penyusunan laporan ini bertujuan untuk memberikan gambaran yang jelas tentang kondisi keuangan asrama dan membantu pengurus dalam pengambilan keputusan terkait manajemen keuangan.

2.3. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem adalah proses mengidentifikasi, memahami, dan mendefinisikan kebutuhan-kebutuhan yang harus dipenuhi oleh sistem yang akan dikembangkan. Tujuan dari analisis kebutuhan sistem adalah memastikan bahwa sistem yang dibangun akan memenuhi harapan dan kebutuhan pengguna serta mencapai tujuan bisnis yang diinginkan.

Berikut adalah penjelasan lebih detail tentang konsep analisis kebutuhan sistem, tahap-tahap analisis kebutuhan sistem, teknik pengumpulan data dalam analisis kebutuhan sistem, dan dokumen yang dihasilkan dalam analisis kebutuhan sistem:

2.3.1. Konsep Analisis Kebutuhan Sistem

Analisis kebutuhan sistem melibatkan pemahaman mendalam tentang lingkungan bisnis, tujuan organisasi, kebutuhan pengguna, dan kendala-kendala yang ada. Konsep utama dalam analisis kebutuhan sistem meliputi:

1. Memahami konteks bisnis: Memahami proses bisnis yang ada, struktur organisasi, dan tujuan bisnis yang ingin dicapai.
2. Memahami kebutuhan pengguna: Mengidentifikasi dan memahami kebutuhan, harapan, dan masalah yang dihadapi oleh pengguna sistem.
3. Mengidentifikasi kebutuhan fungsional dan non-fungsional: Mengidentifikasi kebutuhan yang terkait dengan fitur dan fungsi sistem (kebutuhan fungsional) serta kebutuhan terkait performa, keamanan, dan keandalan (kebutuhan non-fungsional).
4. Membuat pemodelan sistem: Membuat model yang menggambarkan proses bisnis, entitas-entitas yang terlibat, dan hubungan antar entitas.

2.3.2. Tahap-tahap Analisis Kebutuhan Sistem

Tahap-tahap dalam analisis kebutuhan sistem meliputi:

1. Identifikasi kebutuhan: Mengumpulkan informasi tentang kebutuhan pengguna dan tujuan bisnis yang ingin dicapai.

2. Analisis kebutuhan: Menganalisis dan memahami kebutuhan yang teridentifikasi, serta mengklarifikasi dan mengelaborasi kebutuhan tersebut.
3. Verifikasi kebutuhan: Memastikan bahwa kebutuhan yang diidentifikasi dan dianalisis benar-benar memenuhi harapan pengguna dan tujuan bisnis yang diinginkan.
4. Validasi kebutuhan: Mengkonfirmasi bahwa kebutuhan yang diidentifikasi dan dianalisis sesuai dengan konteks bisnis, memungkinkan untuk diimplementasikan, dan konsisten dengan kendala dan batasan yang ada.

2.4.Basis Data

Dalam dunia digital, basis data atau *database* dapat disamakan dengan sebuah "arsip digital" yang menampung dan mengorganisir beragam data. Seperti perpustakaan yang menyimpan buku-buku dengan topik yang berbeda, basis data juga mengumpulkan informasi dalam satu tempat menggunakan perangkat keras dan memprosesnya melalui perangkat lunak khusus. Proses pendefinisian basis data melibatkan penentuan jenis data yang akan disimpan, struktur yang digunakan, dan batasan yang diterapkan pada setiap informasi yang ada. Dalam konteks sistem informasi, *database* memiliki peran sentral yang tak tergantikan. Ia menjadi landasan utama dalam menyediakan informasi yang dibutuhkan oleh para pengguna. Sebagai penghubung antara data, *database* memungkinkan pengguna untuk mendapatkan informasi yang relevan, terkini, dan akurat. Dengan demikian, *database* berperan penting dalam menyelenggarakan sistem informasi yang efisien dan efektif[8].

2.4.1. Struktur Basis Data

Struktur basis data mengacu pada cara data disimpan dan diorganisir dalam database. Terdapat beberapa struktur dasar dalam basis data, antara lain:

1. **Tabel:** Tabel merupakan struktur utama dalam basis data yang terdiri dari baris dan kolom. Setiap kolom mewakili atribut atau jenis data tertentu, sedangkan setiap baris berisi data spesifik yang berkaitan dengan atribut-atribut tersebut.
2. **Field:** *Field* atau kolom dalam tabel merupakan bagian terkecil dari struktur basis data. Field mendefinisikan jenis data yang akan disimpan, misalnya teks, angka, tanggal, dan sebagainya.
3. **Record:** *Record* atau baris dalam tabel merupakan kumpulan field yang saling terkait. Setiap *record* mewakili entitas atau objek yang unik dalam basis data.
4. **Kunci:** Kunci adalah atribut atau kombinasi atribut yang digunakan untuk mengidentifikasi secara unik setiap *record* dalam tabel. Kunci dapat digunakan untuk menghubungkan tabel-tabel yang berbeda dalam basis data.

2.4.2. Jenis-jenis Basis Data

Terdapat beberapa jenis basis data yang umum digunakan, antara lain:

1. **Basis Data Relasional:** Basis data relasional menggunakan model relasional untuk menyimpan data dalam bentuk tabel yang terkait satu sama lain melalui kunci-kunci. Contoh DBMS yang populer untuk basis data relasional adalah MySQL, Oracle, dan Microsoft SQL Server.

2. **Basis Data Hierarki:** Basis data hierarki mengorganisir data dalam bentuk struktur hierarkis, mirip dengan struktur pohon. Data dihubungkan melalui relasi induk-anak. Contoh DBMS yang populer untuk basis data hierarki adalah IBM's *Information Management System (IMS)*.
3. **Basis Data Jaringan:** Basis data jaringan mengorganisir data dalam bentuk grafik, di mana setiap entitas dapat terhubung dengan entitas lain melalui hubungan yang kompleks. Contoh DBMS yang populer untuk basis data jaringan adalah *Integrated Data Store (IDS)*.
4. **Basis Data Berorientasi Objek:** Basis data berorientasi objek menggabungkan paradigma pemrograman berorientasi objek dengan basis data. Data disimpan dalam bentuk objek dengan properti dan metode. Contoh DBMS yang populer untuk basis data berorientasi objek adalah PostgreSQL dan MongoDB.

2.4.3. Relasi Antar Tabel dalam Basis Data

Relasi antar tabel dalam basis data mengacu pada hubungan atau keterkaitan antara tabel-tabel dalam basis data relasional. Terdapat tiga jenis relasi antar tabel yang umum digunakan:

1. *One-to-One* (Satu-satu): Setiap *record* dalam satu tabel berkorelasi dengan tepat satu *record* dalam tabel lain.
2. *One-to-Many* (Satu-ke-Banyak): Setiap *record* dalam satu tabel berkorelasi dengan banyak *record* dalam tabel lain.
3. *Many-to-Many* (Banyak-ke-Banyak): Setiap *record* dalam satu tabel berkorelasi dengan banyak *record* dalam tabel lain, dan sebaliknya.

Relasi antar tabel ini diimplementasikan melalui penggunaan kunci-kunci yang menghubungkan atribut-atribut yang relevan antara tabel-tabel tersebut.

2.5. Perancangan *Database*

Perancangan *database* adalah proses merencanakan struktur, format, dan hubungan antar tabel dalam sebuah *database*. Tujuannya adalah untuk menyimpan data dengan efisien, mengurangi redundansi, dan memastikan integritas data. Proses perancangan *database* melibatkan beberapa langkah, termasuk analisis kebutuhan, normalisasi, desain *Entity-Relationship Diagram* (ERD), dan penentuan skema *database*.

2.5.1. Normalisasi *Database*

Normalisasi *database* adalah proses mengorganisir struktur tabel dalam *database* untuk menghilangkan redundansi data dan memastikan integritas data. Normalisasi dilakukan dengan mengaplikasikan aturan normalisasi, seperti *First Normal Form* (1NF), *Second Normal Form* (2NF), dan seterusnya. Setiap normalisasi bertujuan untuk menghindari duplikasi data dan anomali seperti update, insert, atau delete yang tidak konsisten.

2.5.2. Desain *Entity-Relationship Diagram* (ERD):

Entity-Relationship Diagram (ERD) adalah diagram yang digunakan untuk memodelkan entitas, atribut, dan hubungan antar entitas dalam sebuah *database*. ERD menggunakan notasi seperti entitas, atribut, hubungan, dan kardinalitas untuk menggambarkan struktur data secara visual. ERD membantu dalam memahami hubungan antar entitas dan menjadi dasar untuk merancang skema *database*.

2.5.3. Skema *Database*:

Skema *database* adalah representasi struktur *database* yang terdiri dari tabel, atribut, kunci primer, kunci asing, dan hubungan antar tabel. Skema *database* menentukan bagaimana tabel dan atribut diorganisir, serta hubungan antar tabel yang memungkinkan penggabungan data. Skema *database* juga mencakup definisi tipe data, batasan integritas, dan aturan lainnya yang mengatur data dalam *database*.

2.6. Perangkat Lunak Pendukung

Perangkat lunak pendukung adalah alat-alat atau program komputer yang digunakan untuk membantu penulis atau peneliti dalam menyelesaikan penelitian atau tugas tertentu. Perangkat lunak ini dapat mempermudah dan meningkatkan efisiensi dalam pengumpulan, analisis, dan presentasi data. Berikut ini adalah beberapa contoh perangkat lunak pendukung yang sering digunakan dalam penelitian.

2.6.1. XAMPP

XAMPP (X, Apache, MySQL, PHP, Perl) adalah sebuah paket pengembangan web berbasis *open-source* yang dikembangkan oleh Apache Friends. XAMPP menyediakan lingkungan pengembangan yang lengkap untuk membangun dan menjalankan aplikasi web secara lokal[9].

Dengan menggunakan XAMPP, pengguna tidak perlu menginstal program-program pendukung seperti Apache, MySQL, PHP, dan Perl secara terpisah. Semua kebutuhan tersebut telah terintegrasi dalam paket XAMPP, sehingga pengguna dapat dengan mudah menginstal dan mengonfigurasi lingkungan pengembangan web hanya dengan beberapa langkah.

XAMPP tersedia dalam berbagai versi untuk berbagai sistem operasi, seperti Windows, macOS, dan Linux. Pengguna dapat memilih jenis XAMPP yang sesuai dengan sistem operasi yang digunakan, kemudian melakukan ekstraksi dan instalasi paket aplikasi XAMPP.

Dengan XAMPP, pengembang web dapat menghemat waktu dan upaya dalam mengatur dan mengintegrasikan program-program pendukung yang diperlukan untuk membangun aplikasi web. XAMPP menyediakan lingkungan yang siap digunakan dengan konfigurasi standar yang telah ditetapkan.

2.6.2. Apache

Apache adalah salah satu server web yang paling populer dan digunakan secara luas di dunia. Penulis menggunakan Apache versi 2.0 dalam membangun aplikasi ini. Apache adalah sebuah perangkat lunak open source, yang berarti siapa pun dapat menggunakannya, mengambil, dan bahkan mengubah kode programnya sesuai kebutuhan.

Tugas utama Apache adalah menyajikan halaman web yang benar kepada pengguna berdasarkan kode PHP yang ditulis oleh pembuat halaman web. Apache dapat menjalankan skrip PHP dan menghasilkan halaman web yang dinamis. Selain itu, Apache juga dapat berinteraksi dengan basis data seperti MySQL untuk mendukung pembuatan halaman web yang lebih kompleks dan berhubungan dengan data yang tersimpan.

Apache memiliki fitur-fitur keamanan dan kinerja yang kuat, serta mendukung berbagai protokol seperti HTTP, HTTPS, dan FTP. Dengan

fleksibilitasnya yang tinggi, Apache dapat dikonfigurasi dan disesuaikan sesuai kebutuhan pengguna.

2.6.3. MySQL

MySQL adalah salah satu program *database* server yang sangat populer dan efisien. Dengan kemampuannya dalam menerima dan mengirimkan data secara cepat, serta dukungan untuk perintah SQL standar (*Structured Query Language*), MySQL menjadi pilihan yang kuat dalam pengelolaan *database*.

MySQL memiliki dua bentuk lisensi, yaitu *Free Software* dan *Shareware*. Produk MySQL AB dikategorikan sebagai *database open source* terpopuler di dunia dan berada di bawah lisensi GNU/GPL (*General Public License*). Versi awal MySQL, yaitu versi 1.0, dirilis secara terbatas pada bulan Mei 1996. Kemudian, pada bulan Oktober, versi 3.11.0 dirilis secara publik. Meskipun awalnya MySQL tidak dilisensikan di bawah GPL, tetapi menggunakan lisensi khusus yang memberikan akses terbatas, saat ini MySQL telah berpindah ke lisensi GPL yang memungkinkan akses terbuka terhadap source code dan penggunaan gratis untuk kebutuhan nonkomersial[10].

2.6.4. PHP

PHP (*Hypertext Preprocessor*), sebuah bahasa pemrograman sisi server, telah mendapatkan keunggulan yang signifikan dalam pemrograman web. Awalnya disebut sebagai singkatan dari *Personal Home Page*, PHP tumbuh melampaui tujuan awalnya pembuatan situs web pribadi[7]. Saat ini, ini adalah bahasa serbaguna yang digunakan untuk membangun situs web oleh berbagai platform terkenal seperti Wikipedia, WordPress, dan Joomla.

Pengembang dapat memanfaatkan PHP untuk membuat halaman web interaktif yang berinteraksi dengan *database* dan sumber daya lainnya, menampilkan kemampuannya yang luar biasa dan fleksibel dalam pengembangan web. PHP, tersedia secara gratis dan dianut sebagai *open-source*.

2.7.Black box Testing

Teknik pengujian *black box* terutama berfokus pada fungsionalitas perangkat lunak dan tidak memperhitungkan struktur internal atau detail implementasinya. Penguji berinteraksi dengan perangkat lunak dalam pengujian kotak hitam. Berbagai input diberikan untuk mengamati output. Tujuannya adalah untuk menentukan apakah perangkat lunak berfungsi dengan benar dan memberikan keluaran yang diharapkan untuk situasi masukan yang berbeda. Tes ini berfokus pada perilaku eksternal perangkat lunak, termasuk responsnya terhadap input yang berbeda dan kepatuhannya terhadap persyaratan fungsional. Dalam *black box* testing, penguji tidak memiliki pengetahuan tentang bagaimana sistem melakukan proses internalnya[11]. Teknik-teknik yang umum digunakan dalam *black box* testing adalah sebagai berikut:

1. Teknik *Equivalence Partitioning*: Data input dibagi menjadi kumpulan nilai yang setara, baik valid maupun tidak valid. Pengujian dilakukan pada setiap kumpulan nilai untuk memastikan perilaku yang konsisten.
2. Teknik *Boundary Value*: Pengujian dilakukan pada nilai batas atas dan batas bawah suatu variabel untuk menemukan error pada batas tersebut. Hal ini penting karena seringkali masalah terjadi pada nilai batas.

3. Teknik *All-pair Testing*: Seluruh kombinasi data diskrit diuji. Teknik ini berguna untuk menguji aplikasi yang menggunakan input dengan banyak pilihan, seperti kotak centang atau tombol radio.
4. Teknik *Decision Table*: Kombinasi input disusun dalam bentuk tabel untuk melakukan pengujian secara sistematis. Teknik ini cocok untuk menguji fungsi yang memiliki hubungan logis antara dua atau lebih input.
5. Teknik *State Transition*: Pengujian dilakukan untuk melihat bagaimana sistem bereaksi terhadap perubahan input yang mengubah keadaan sistem. Teknik ini berguna pada aplikasi yang memiliki logika transisi keadaan, seperti sistem yang memberikan jumlah percobaan terbatas.
6. Teknik *Cause-Effect*: Grafik digunakan untuk menggambarkan hubungan antara penyebab error dan efeknya.
7. Teknik *Error Guessing*: Pengujian dilakukan berdasarkan pengalaman dan pengetahuan penguji untuk mengidentifikasi potensi error dalam aplikasi.
8. Teknik *Use Case*: Setiap fungsi sistem diuji dengan menjalankan sistem dari awal hingga akhir untuk memastikan semua fitur berfungsi dengan baik.

Black box testing memiliki keuntungan seperti dapat dilakukan tanpa pengetahuan tentang pemrograman, efektif untuk menguji aplikasi yang besar dan kompleks, serta dapat mendeteksi error pada tahap awal pengujian[12]. Keuntungan dari *black box testing* meliputi:

1. Independensi: Penguji dapat melakukan *black box testing* tanpa harus memiliki pengetahuan tentang kode internal atau detail implementasi,

sehingga cocok untuk pengujian oleh individu atau tim yang tidak terlibat dalam proses pengembangan perangkat lunak.

2. Fokus pada perspektif pengguna: *Black box testing* membantu mengevaluasi perangkat lunak dari perspektif pengguna, memastikan bahwa perangkat lunak memenuhi fungsionalitas dan persyaratan yang diharapkan.
3. Mendorong pengujian yang menyeluruh: Penguji tidak terbatas oleh struktur internal perangkat lunak, memungkinkan mereka untuk menguji semua skenario dan input yang mungkin, termasuk kasus uji ekstrem dan situasi yang tidak terduga.
4. Pengujian awal: *Black box testing* dapat dilakukan pada tahap awal siklus pengembangan perangkat lunak, bahkan sebelum implementasi perangkat lunak yang lengkap, berdasarkan spesifikasi dan persyaratan yang telah ditetapkan.

Namun, *black box testing* juga memiliki keterbatasan, seperti ketidakmampuannya untuk mengungkap jenis cacat tertentu yang terkait dengan logika internal atau implementasi perangkat lunak. Selain itu, sulit untuk merancang kasus uji yang mencakup semua skenario yang mungkin.