

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Teori Dasar**

##### **2.1.1 Pengertian Rancang Bangun**

Rancangan bangun sistem informasi merupakan proses perencanaan, pengembangan, dan implementasi suatu sistem yang dalam penggunaannya dapat mengumpulkan, memproses, menyimpan, dan mendistribusikan informasi (C. Laudon dan Jane P. Laudon, 2020). Proses ini dimulai dengan analisis kebutuhan, di mana desainer dan analis sistem mengumpulkan informasi melalui observasi, wawancara, dan survei untuk memahami kebutuhan operasional dan menentukan fitur-fitur utama yang harus ada di dalam sistem. Setelah kebutuhan diketahui, tahap berikutnya adalah perancangan konseptual. Pada tahap ini, konsep dan kebutuhan diterjemahkan menjadi desain yang lebih konkret, meliputi arsitektur sistem, perancangan basis data, alur kerja, antarmuka pengguna, hingga memperhatikan aspek keamanan, skalabilitas, dan interoperabilitas agar sesuai standar teknis.

Selanjutnya, dibuatlah prototipe sebagai versi awal sistem untuk dievaluasi dan diuji langsung oleh pengguna. Prototipe ini membantu pengembang dan desainer mendapatkan umpan balik secara cepat, sehingga perbaikan dan penyesuaian bisa dilakukan secara iteratif sebelum pengembangan penuh dimulai. Setelah prototipe disempurnakan, masuk ke tahap implementasi teknis, yakni pengembangan sistem sesuai desain, meliputi pengkodean, pembangunan basis data, hingga pengaturan infrastruktur. Pengujian dilakukan secara berkesinambungan pada tahap ini untuk memverifikasi bahwa seluruh komponen sistem berfungsi sebagaimana mestinya dan telah terhubung dengan baik. (Darma Suhesti et al., 2023)

Pada akhirnya, tahap terakhir adalah pengujian dan dokumentasi. Sistem diuji dalam berbagai kondisi operasional untuk memastikan kualitas dan keandalannya. Selain itu, disusun dokumentasi lengkap seperti panduan pengguna, dokumentasi teknis, dan prosedur operasional agar sistem mudah digunakan dan dirawat di

kemudian hari. Dengan mengikuti seluruh tahapan ini, rancang bangun mampu menghadirkan sistem informasi yang efektif, efisien, dan sesuai kebutuhan.

### **2.1.2 Pengertian Aplikasi**

(Adlan Al Hawari Nasution & Suryana, 2023) Aplikasi merupakan sebuah program komputer atau perangkat lunak yang dirancang dan dikembangkan untuk menjalankan fungsi-fungsi tertentu sesuai dengan sistem yang digunakan. Istilah "aplikasi" berasal dari bahasa Inggris *application*, yang secara umum berarti penerapan atau penggunaan. Secara sederhana, aplikasi dapat dipahami sebagai bentuk implementasi dari perangkat lunak yang dibuat dengan tujuan membantu pengguna dalam menyelesaikan tugas-tugas spesifik.

Aplikasi terbagi menjadi beberapa jenis, seperti aplikasi desktop yang dijalankan di komputer, aplikasi mobile yang dioperasikan melalui ponsel pintar, serta aplikasi web yang diakses melalui browser tanpa perlu diunduh. Masing-masing jenis aplikasi memiliki keunggulan dan kegunaan tersendiri, tergantung pada kebutuhan pengguna dan platform yang digunakan.

### **2.1.3 Pengertian Pelacakan Tanggal**

Pelacakan tanggal merupakan elemen krusial dalam sistem informasi manajemen yang bertugas merekam, memantau, serta mengelola data terkait waktu atau tanggal dalam suatu proses operasional atau kegiatan bisnis. Fungsinya adalah untuk menjamin bahwa setiap aktivitas yang bergantung pada waktu dapat dilakukan secara teratur, sesuai jadwal, dan terdokumentasi dengan akurat. Menurut (Kenneth C. Laudon, 2003), sistem informasi berperan penting dalam mendukung aktivitas pengambilan keputusan, koordinasi, serta pengendalian dalam organisasi melalui pencatatan data yang akurat dan real-time.

(Verenich et al., 2018) Dalam konteks bisnis dan teknologi informasi, pelacakan tanggal digunakan untuk mengelola berbagai elemen penting seperti penjadwalan tugas, batas waktu proyek, masa berlaku dokumen, serta riwayat transaksi atau aktivitas pengguna. Fitur ini banyak diterapkan dalam sistem manajemen proyek, perangkat lunak akuntansi, sistem informasi kepegawaian, dan sistem inventaris

berbasis web. Dengan adanya pelacakan waktu yang sistematis, organisasi dapat meningkatkan efisiensi operasional, mengurangi risiko keterlambatan, serta menyediakan data historis yang mendukung analisis kinerja dan pengambilan keputusan strategis.

#### **2.1.4 Pengertian Obat Kadaluarsa**

Menurut (Syamsuni, 2005) Obat merupakan zat atau kombinasi dari beberapa zat yang digunakan untuk membantu proses diagnosis, pencegahan, peringanan, penyembuhan, maupun penghilangan penyakit, gejala penyakit, cedera, atau gangguan baik fisik maupun mental pada manusia ataupun hewan. Selain itu, obat juga dapat digunakan untuk tujuan memperindah penampilan tubuh atau bagian tertentu dari tubuh manusia.

Obat kadaluarsa atau *Expired Date* adalah obat yang telah melewati tanggal kadaluarsa yang tertera pada kemasan yang menjadi tanda bahwa obat tersebut sudah tidak layak dikonsumsi atau digunakan. Terdapat dua jenis tanggal kadaluarsa pada obat yakni tanggal kadaluarsa yang dapat tertulis pada kemasan obat dan tanggal kadaluarsa setelah obat dibuka dari kemasan primernya atau biasa disebut *Beyond Use Date* (Nur Alinda & Karuniawati, 2020).

#### **2.2.5 Pengertian Website**

Menurut (Elgamar, 2020) website adalah suatu media yang terdiri dari beberapa halaman yang saling berkaitan satu sama lain, dan berfungsi sebagai media untuk menampilkan suatu informasi, baik berbentuk gambar, video, teks, suara, ataupun gabungan dari semuanya. Website bersifat multiplatform yang artinya dapat dibuka dari segala perangkat atau device yang terhubung dengan jaringan internet. Walaupun teknologi ini sudah cukup lama digunakan, namun saat ini masih banyak sekali perusahaan-perusahaan yang masih menggunakan website dalam menampilkan profil perusahaan (*company profile*), menjual produk, ataupun sebagai sistem yang dapat digunakan oleh pelanggan.

Kesimpulannya, website merupakan media digital yang fleksibel dan multiguna, mampu menampilkan berbagai jenis informasi dalam bentuk teks, gambar, video, maupun suara.

### **2.1.6 Pengertian Apotek**

Menurut Peraturan Pemerintah Republik Indonesia Nomor 51 Tahun 2009 tentang Pekerjaan Kefarmasian, apotek merupakan fasilitas pelayanan di bidang farmasi yang menjadi tempat bagi apoteker menjalankan praktik kefarmasiannya. ( Menkes, 2009 )

Disini diperlukan peran Apoteker yang sangat penting untuk memilihkan obat yang sesuai dengan kondisi pasien. Penelitian ini dilakukan untuk mengetahui distribusi apotek di Kota Bengkulu yang melakukan pelayanan kefarmasian berdasarkan implementasi standar pelayanan kefarmasian dan beberapa faktor yang mempengaruhinya(Standar et al., n.d.).

### **2.1.7 Software Development Life Cycle (SDLC)**

SDLC (Software Development Life Cycle) adalah sebuah proses yang digunakan dalam pengembangan maupun modifikasi sistem perangkat lunak dengan memanfaatkan berbagai model dan metode yang umum digunakan dalam dunia rekayasa perangkat lunak. SDLC juga dapat diartikan sebagai suatu pola kerja sistematis yang terdiri dari beberapa tahapan, yaitu perencanaan (*planning*), analisis kebutuhan (*analysis*), perancangan sistem (*design*), implementasi (*implementation*), pengujian (*testing*), serta pemeliharaan (*maintenance*).(Fitria et al., n.d.)

Salah satu manfaat utama dari SDLC adalah kemampuannya untuk memberikan struktur yang jelas dalam pengembangan perangkat lunak, sehingga tim pengembang dapat bekerja lebih terorganisir dan efisien. Dengan adanya pendekatan ini, pengembang dapat mengidentifikasi dan memperbaiki masalah lebih awal sebelum produk akhir dirilis(Sommerville, 2016), sehingga mengurangi biaya perbaikan di kemudian hari. Metode SDLC juga dapat diterapkan dalam berbagai model pengembangan, seperti Waterfall, Agile, dan DevOps, yang masing-masing memiliki keunggulan tersendiri tergantung pada kebutuhan proyek.

(Kendall, 2010)Dalam praktiknya, pemilihan model SDLC yang tepat sangat bergantung pada kompleksitas proyek, sumber daya yang tersedia, serta kebutuhan

bisnis dan teknis. Misalnya, metode Agile lebih cocok untuk proyek yang memerlukan fleksibilitas dan iterasi yang cepat, sementara metode Waterfall lebih sesuai untuk proyek dengan persyaratan yang jelas sejak awal. Dengan menerapkan SDLC secara efektif, Organisasi dapat memastikan bahwa proses pengembangan perangkat lunak berjalan efektif, memenuhi ekspektasi pengguna, serta mengedepankan standar kualitas dan keamanan yang tinggi.

Dalam SDLC, terdapat beberapa langkah utama yang harus dilalui, yaitu:

A. *Planning* (Perencanaan)

Tahap awal yang bertujuan untuk menentukan ruang lingkup proyek, sumber daya yang dibutuhkan, jadwal kerja, serta identifikasi risiko. Perencanaan yang baik akan menjadi landasan keberhasilan tahap-tahap selanjutnya.

B. *Analysis* (Analisis)

Fokus dari tahap ini adalah menghimpun informasi terkait kebutuhan pengguna dan menelaah sistem sebelumnya (bila ada), guna memastikan sistem baru dapat dirancang sesuai dengan kebutuhan tersebut.

C. *Design* (Perancangan)

Berdasarkan hasil analisis, tahap ini menghasilkan rancangan sistem baik dari sisi antarmuka, arsitektur, basis data, hingga struktur program. Desain ini akan menjadi panduan teknis dalam proses implementasi.

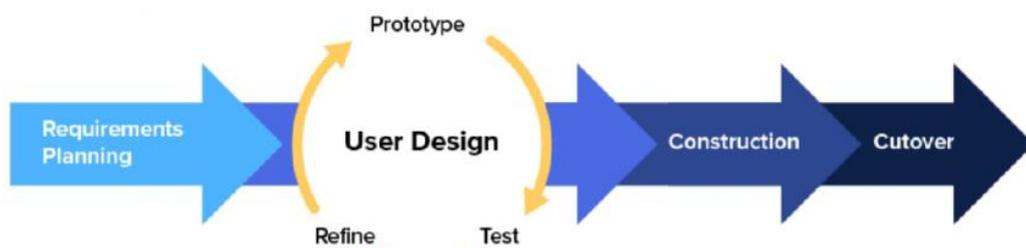
D. *Implementation* (Implementasi)

Tahap di mana desain yang telah dibuat diubah menjadi kode program yang dapat dijalankan. Tim pengembang mulai membangun sistem sesuai spesifikasi desain.

### **2.1.8 Rapid Application Development (RAD)**

Menurut (Scott Tilley, 2016). Rapid Application Development (RAD) adalah teknik untuk mempercepat pengembangan sistem informasi dengan fokus pada pengembangan sistem yang berfungsi. Berbeda dengan JAD yang menghasilkan model persyaratan, produk akhir RAD adalah sistem informasi baru.

Kesimpulannya, Rapid Application Development (RAD) adalah pendekatan pengembangan sistem yang menekankan kecepatan dan fungsionalitas, dengan tujuan utama menghasilkan sistem informasi yang dapat langsung digunakan. Tidak seperti metode lain seperti JAD yang lebih berfokus pada dokumentasi kebutuhan, RAD mengedepankan pembuatan model awal (*prototype*) dan siklus pengembangan berulang untuk menciptakan sistem yang sesuai dengan keinginan pengguna dalam waktu yang lebih efisien. (R. S. , & M. B. R. Pressman, 2020).



Gambar 2.1 Tahapan RAD(Habibi, 2020)

#### A. Perencanaan Kebutuhan

*Requirements Planning*, atau perencanaan kebutuhan, melibatkan proses mengidentifikasi permasalahan, kebutuhan informasi, serta hambatan yang ada. Tahap ini digunakan untuk merumuskan tujuan, batasan sistem dan masalah, serta menyusun berbagai alternatif solusi. Tujuannya adalah untuk membangun pemahaman yang sama antara pengembang dan pengguna mengenai sistem yang akan dibuat(Dennis, 2015).

#### B. User Design

Langkah kedua, perhatian terutama diberikan pada pembentukan struktur dan spesifikasi teknis sistem sesuai dengan kebutuhan yang sudah diidentifikasi sebelumnya(Aidil Ma et al., n.d.). Proses ini bersifat iteratif, di mana pengujian dan penyempurnaan dilakukan berulang kali hingga desain sistem sesuai dengan harapan pengguna. Siklus ini memungkinkan sistem dikembangkan secara fleksibel dan cepat, sambil tetap menjaga relevansi terhadap kebutuhan pengguna.

#### **D. Construction**

Tahap konstruksi dalam metode RAD merupakan pengembangan perangkat lunak berdasarkan desain pengguna yang dilakukan pada tahap sebelumnya (Aidil Ma et al., n.d.). Karena tahap ini mengikuti hasil dari proses iteratif sebelumnya, pengembangan dapat dilakukan dengan lebih efisien. Tahap ini juga memungkinkan penyesuaian cepat terhadap perbaikan minor yang mungkin masih diperlukan sebelum implementasi.

#### **E. Cutover**

(Wiredu, 2024) Pada tahap penutup, dilakukan proses pengujian guna memastikan bahwa perangkat lunak telah siap untuk dioperasikan. Tujuan dari pengujian ini adalah untuk menilai apakah sistem yang dikembangkan mudah digunakan serta mampu memberikan pengalaman yang baik bagi penggunanya. Selain itu, tahap ini juga berfungsi untuk memastikan bahwa peralihan dari sistem lama ke sistem baru dapat berlangsung dengan mulus tanpa menimbulkan gangguan terhadap aktivitas operasional yang sedang berjalan.

#### **2.1.9 Object-Oriented Analysis and Design**

Metode OOAD merupakan pendekatan modern dalam menyelesaikan masalah dengan membangun pemodelan yang didasarkan pada konsep dari dunia nyata. OOAD melibatkan analisis dan desain menggunakan pendekatan berorientasi objek, yang mencakup analisis objek dan desain objek (Natsir & Kurniawan, 2025).

(Booch, 2007) Selain itu, analisis dalam OOAD juga mencakup identifikasi atribut dan perilaku yang dimiliki oleh setiap objek. Atribut adalah informasi yang dimiliki oleh objek, sementara perilaku merujuk pada tindakan yang dapat dilakukan objek tersebut. Dengan memahami objek, atribut, dan perilaku ini, desainer dapat merancang sistem yang lebih terstruktur dan modular, yang akan lebih mudah dipelihara dan dikembangkan di masa depan. Pendekatan ini sangat berguna dalam menciptakan sistem perangkat lunak yang kompleks, karena memungkinkan

desainer untuk menangani elemen-elemen sistem dengan cara yang lebih realistis, dekat dengan dunia nyata, dan lebih mudah dipahami.

### 2.2.1 UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*)(Mubarak et al., 2019)

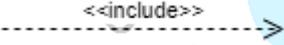
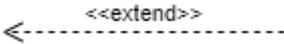
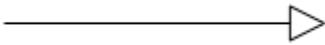
UML menyediakan seperangkat notasi standar yang digunakan untuk merepresentasikan berbagai aspek dari sistem perangkat lunak, mulai dari kebutuhan fungsional hingga desain implementasi. Dengan menggunakan diagram-diagram seperti Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram, pengembang dapat memodelkan struktur, perilaku, serta interaksi antar komponen dalam sistem(G. , R. J. , & J. I. Booch, 2005).

#### A. Use Case Diagram

Use Case Diagram:menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”(Grady Booch Robert A. Maksimchuk Michael W. Engle Bobbi J. Young, n.d.). Dengan sistem melalui serangkaian use case (kasus penggunaan) yang merepresentasikan layanan atau fungsi yang dapat diakses oleh aktor. Masing-masing use case menggambarkan tujuan spesifik yang ingin dicapai oleh aktor melalui sistem.

Tabel 2. 1 Notasi Use Case Diagram(Billy, 2023)

Notasi	Keterangan
--------	------------

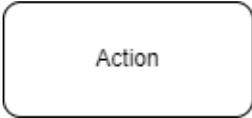
	<p>Merupakan entitas eksternal, baik manusia, perangkat, maupun sistem lain, yang memiliki interaksi langsung dengan sistem yang sedang dikembangkan. Aktor berperan sebagai pemicu maupun penerima dari suatu proses dalam sistem.</p>
	<p>Menggambarkan fungsionalitas atau layanan spesifik yang ditawarkan oleh sistem kepada aktor. Setiap <i>use case</i> mewakili satu aktivitas utama yang bernilai bagi aktor.</p>
	<p>Menunjukkan koneksi langsung antara aktor dan <i>use case</i>, menandakan bahwa aktor tersebut terlibat dalam pelaksanaan fungsi tertentu dalam sistem.</p>
	<p>Menyatakan bahwa suatu <i>use case</i> selalu mencakup proses atau fungsi dari <i>use case</i> lain sebagai bagian dari alur kerjanya. Digunakan ketika ada proses umum yang dibagikan oleh beberapa <i>use case</i>.</p>
	<p>Menunjukkan bahwa suatu <i>use case</i> dapat diperluas oleh <i>use case</i> lain dalam kondisi tertentu. Relasi ini bersifat opsional dan terjadi hanya jika kondisi pemicu terpenuhi.</p>
	<p>Digunakan untuk memperlihatkan hubungan hierarki, baik antar aktor maupun antar <i>use case</i>. Biasanya</p>

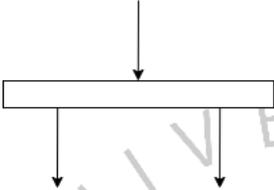
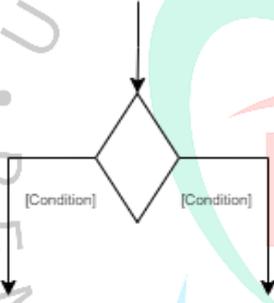
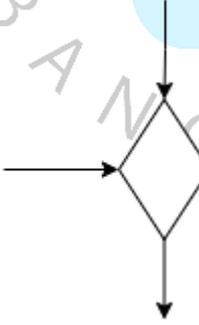
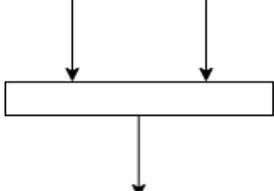
	menunjukkan pewarisan tanggung jawab atau perilaku.
	Direpresentasikan dalam bentuk kotak besar pada <i>Use Case Diagram</i> , simbol ini menunjukkan batas-batas sistem yang dianalisis, serta mencakup semua <i>use case</i> yang berada di dalam ruang lingkup sistem tersebut.

## B. Activity Diagram

Menurut (Grady Booch Robert A. Maksimchuk Michael W. Engle Bobbi J. Young, n.d.) Activity diagram memberikan gambaran visual mengenai aliran aktivitas, baik dalam sistem, bisnis, alur kerja, maupun proses lainnya. Diagram ini menyoroti aktivitas-aktivitas yang dilakukan serta siapa (atau apa) yang bertanggung jawab dalam menjalankan aktivitas tersebut. Elemen utama dari activity diagram meliputi *action node*, *control node*, dan *object node*.

Tabel 2. 2 Notasi Activity Diagram (Billy, 2023)

Notasi	Keterangan
	Merupakan titik awal dalam <i>activity diagram</i> , biasanya digambarkan sebagai lingkaran kecil dengan satu panah keluar yang mengarah ke aktivitas pertama dalam proses.
	Menggambarkan suatu aktivitas atau langkah kerja yang dilakukan dalam alur proses. Setiap <i>action</i>

	merepresentasikan satu unit kerja tertentu dalam sistem.
	Menunjukkan arah atau aliran dari satu aktivitas ke aktivitas berikutnya. Digunakan untuk menghubungkan berbagai <i>action</i> dalam diagram.
	Simpul ini digunakan untuk membagi alur proses menjadi beberapa jalur paralel yang dapat berjalan secara bersamaan.
	Berbentuk belah ketupat (diamond), simpul ini digunakan untuk merepresentasikan titik pengambilan keputusan, di mana alur proses bercabang berdasarkan kondisi tertentu.
	Merupakan titik penggabungan beberapa jalur yang sebelumnya bercabang, dan disatukan kembali ke dalam satu alur tunggal. Simbolnya berupa diamond dengan banyak panah masuk dan satu panah keluar.
	Digunakan untuk menyatukan jalur-jalur proses paralel yang sebelumnya dipisahkan oleh <i>fork</i> . Biasanya digambarkan dengan garis-garis yang bergabung ke satu simpul.

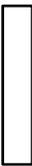
	<p>Menunjukkan akhir dari seluruh rangkaian aktivitas. Simbolnya berupa lingkaran dengan batas tebal tanpa panah keluar, menandakan bahwa tidak ada aktivitas lanjutan setelah titik ini.</p>
---	---

### C. Sequence Diagram

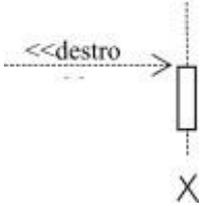
Menurut (Sandi Ashriel Nugraha et al., 2025) Sequence diagram menggambarkan kelakuan ssobjek pada Use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

Tabel 2. 3 Notasi Sequence Diagram (Billy, 2023)

Simbol	Nama	Keterangan
 nama aktor atau <div style="border: 1px solid black; padding: 2px; display: inline-block;">atau</div> <u>Nama aktor</u> tanpawaktu aktif	Aktor/ <i>actor</i>	<p>Aktor adalah entitas eksternal, baik manusia, perangkat, maupun sistem lain, yang memiliki interaksi dengan sistem informasi yang dikembangkan.</p> <p>Meskipun simbol aktor digambarkan sebagai manusia, aktor tidak selalu berarti individu—bisa juga berupa mesin, perangkat, atau sistem. Biasanya, nama aktor diawali dengan kata benda untuk</p>

		menggambarkan fungsinya secara jelas.
	Garis hidup/ <i>lifeline</i>	Representasi dari <i>lifeline</i> dalam <i>sequence diagram</i> , yang menunjukkan waktu hidup suatu objek selama interaksi berlangsung.
	Waktu aktif	Menunjukkan <i>activation bar</i> atau <i>focus of control</i> pada <i>sequence diagram</i> , menggambarkan durasi ketika suatu objek sedang

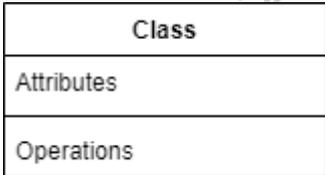
		melakukan proses atau metode tertentu.
	Pesan tipe <i>create</i>	Ini adalah representasi dari <i>message create</i> dalam <i>sequence diagram</i> , dan panggilan metode harus sesuai dengan metode yang didefinisikan dalam <i>class diagram</i> pada kelas objek yang dituju.
	Pesan tipe <i>send</i>	Ini menggambarkan <i>message</i> dalam <i>sequence diagram</i> , di mana satu objek memberi input atau instruksi kepada objek lainnya.
	Pesan tipe <i>return</i>	Menunjukkan bahwa setelah suatu objek menyelesaikan eksekusi sebuah metode, objek tersebut mengirimkan nilai kembali atau respon kepada objek yang memanggilnya.

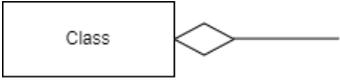
	<p>Pesan tipe <i>destroy</i></p>	<p>Menjelaskan bahwa sebuah objek dapat dihentikan atau dihapus dari sistem oleh objek lain, dan hal ini perlu digambarkan dalam diagram.</p>
---	----------------------------------	---

#### D. Class Diagram

Class diagram merupakan salah satu jenis diagram dalam pemodelan sistem yang digunakan untuk menggambarkan sejumlah kelas yang terdapat dalam perangkat lunak atau sistem yang sedang dirancang (Sandi Ashriel Nugraha et al., 2025). Setiap kelas yang ditampilkan mewakili suatu entitas dalam sistem, lengkap dengan atribut (properti) dan metode (fungsi) yang dimilikinya. Dengan bantuan class diagram, pengembang dapat memperoleh gambaran yang jelas mengenai hubungan antar objek dalam sistem, serta distribusi data dan logika yang terlibat dalam proses kerja sistem tersebut.

Tabel 2. 4 Notasi Class Diagram (Billy, 2023)

Notasi	Keterangan
	<p>Merupakan representasi dari sekelompok objek yang memiliki sifat (atribut) dan perilaku (metode) yang serupa. Dalam diagram, kelas digambarkan sebagai persegi panjang yang terbagi menjadi tiga bagian: nama kelas, daftar atribut, dan daftar operasi atau metode.</p>
	<p>Menunjukkan hubungan umum antara dua kelas. Relasi ini menggambarkan bahwa suatu objek dari satu kelas memiliki koneksi atau interaksi dengan objek dari kelas lainnya.</p>

	<p>Merupakan hubungan di mana sebuah kelas tersusun dari beberapa objek kelas lain, namun tetap memungkinkan objek-objek tersebut berdiri sendiri di luar struktur tersebut. Relasi ini menggambarkan konsep “bagian dari” (has-a) dengan sifat yang longgar.</p>
	<p>Adalah bentuk khusus dari agregasi dengan keterikatan yang lebih kuat. Dalam relasi ini, objek yang menjadi bagian tidak dapat berdiri sendiri dan akan dihapus jika objek induknya dihapus.</p>
	<p>Menyatakan bahwa sebuah kelas menggunakan kelas lain, baik sebagai parameter dalam metode maupun sebagai tipe data dari atribut. Perubahan pada kelas yang digunakan dapat berdampak langsung pada kelas yang bergantung padanya.</p>
	<p>Menggambarkan hubungan antara dua kelas di mana kelas turunan (subclass) mewarisi atribut dan metode dari kelas induk (superclass). Relasi ini digunakan untuk menyatakan konsep generalisasi-spesialisasi.</p>
	<p>Digunakan untuk menunjukkan bahwa suatu kelas konkret mengimplementasikan satu atau lebih metode yang telah didefinisikan dalam sebuah antarmuka (<i>interface</i>). Hubungan ini umum ditemukan dalam bahasa pemrograman berorientasi objek.</p>

(Larman, 2005) Class diagram sangat berguna dalam tahap perancangan karena membantu mengorganisir struktur kode, mendeteksi dependensi antar komponen, serta menjadi dasar dalam proses implementasi berbasis objek. Diagram ini juga membantu tim pengembang menjaga konsistensi dan skalabilitas sistem ketika proyek berkembang.

### **2.2.2 Database**

Basis data (database) merupakan sekumpulan data atau informasi yang disusun secara sistematis, sering kali dengan mempertimbangkan keterkaitan antar komponennya, sehingga memudahkan dalam pengelolaan dan pengambilan informasi. Dengan bantuan sistem ini, pengguna dapat dengan mudah mencari, menyimpan, dan mengelola informasi. Untuk menilai kualitas suatu informasi dalam basis data, dapat digunakan lima kriteria utama, yaitu: keakuratan, ketepatan waktu, kelengkapan, relevansi, dan konsistensi. (Doi & Nurul, 2023).

### **2.2.3 Black Box Testing**

(R. S. Pressman, 2010) Pengujian black box merupakan teknik pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas sistem berdasarkan spesifikasi yang telah ditentukan, tanpa melihat struktur internal atau kode sumber dari program tersebut. Dalam pendekatan ini, penguji hanya mengevaluasi hasil keluaran berdasarkan data masukan, tanpa mengetahui proses yang terjadi di dalam sistem. Karena pengujian dilakukan dari perspektif pengguna akhir, metode ini sangat sesuai untuk memastikan bahwa sistem telah memenuhi kebutuhan fungsional yang diharapkan.

Metode ini menguji berbagai aspek perangkat lunak seperti validasi input, keluaran sistem, antarmuka pengguna, dan perilaku sistem secara keseluruhan. Beberapa teknik yang umum digunakan dalam black box testing antara lain equivalence partitioning (pembagian nilai input ke dalam kelas yang dianggap setara), boundary value analysis (pengujian nilai batas), dan decision table testing. Karena pengujian ini tidak memperhatikan kode sumber, maka sangat cocok digunakan dalam tahap pengujian fungsional (functional testing) atau acceptance testing.

## **2.3 Tinjauan Studi**

Penelitian-penelitian sebelumnya yang relevan dalam bidang rancang bangun aplikasi manajemen stok obat di apotek menunjukkan berbagai pendekatan

dalam pengembangan sistem, terutama dalam pemanfaatan teknologi informasi untuk meningkatkan efisiensi operasional dan keselamatan pasien.

Penelitian oleh Aidil Ma et al. dalam *Sistem Informasi Penjualan Obat Berbasis Website pada Apotek Banten Farma Menggunakan Metode RAD* menyoroti bagaimana metode Rapid Application Development (RAD) digunakan untuk merancang sistem informasi penjualan obat yang lebih cepat dan adaptif terhadap kebutuhan pengguna. Penelitian ini menjadi landasan penting dalam memilih metode RAD sebagai pendekatan pengembangan karena memungkinkan iterasi prototipe dan partisipasi aktif pengguna dalam proses desain.

Selanjutnya, studi oleh Annisa et al. (2024) yang berjudul *Transformasi Digital di Dunia Farmasi: Aplikasi Web untuk Pengelolaan Persediaan Obat di Apotek* menekankan pentingnya digitalisasi dalam pengelolaan stok obat. Penelitian ini menunjukkan bahwa penerapan aplikasi web tidak hanya meningkatkan efisiensi manajemen inventaris, tetapi juga membantu dalam pemantauan masa kadaluarsa obat secara real-time.

Penelitian oleh Karim et al. (2025) mengenai *Pengembangan Sistem Informasi Persediaan Obat Berbasis Web Menggunakan Metode Prototype* di Rumah Sakit Otanaha Kota Gorontalo juga relevan. Studi ini membahas bagaimana sistem berbasis web dapat mengoptimalkan pengelolaan obat dan meminimalkan kesalahan manual, sekaligus meningkatkan akurasi pelacakan ketersediaan obat.

Natsir dan Kurniawan (2025) juga memberikan kontribusi melalui studi mereka yang mengimplementasikan pendekatan Object-Oriented Analysis and Design (OOAD). Meskipun studi mereka difokuskan pada pemesanan pakaian, prinsip-prinsip desain berbasis objek yang digunakan sangat relevan dalam membangun aplikasi pelacakan kadaluarsa obat yang modular dan dapat dikembangkan lebih lanjut.

Terakhir, dalam jurnal oleh Doi dan Nurul (2023) yang berjudul *Analisis Penggunaan Database dalam Sistem Informasi*, peneliti menekankan pentingnya manajemen basis data yang baik dalam menjaga konsistensi dan integritas

informasi dalam sistem informasi, termasuk dalam konteks pengelolaan data farmasi.

Selain dari jurnal, berbagai buku juga menjadi rujukan penting dalam mendukung pengembangan sistem.

Buku karya C. Laudon dan Jane P. Laudon (2020) *Management Information Systems* membahas prinsip dasar sistem informasi manajemen yang menjadi pondasi dalam merancang aplikasi farmasi yang efektif dan efisien.

Selanjutnya, Scott Tilley dan Harry J. Rosenblatt (2016) dalam bukunya *Systems Analysis and Design* menjelaskan secara komprehensif metode RAD serta pentingnya fase analisis dan desain dalam membangun sistem berbasis kebutuhan pengguna.

Buku oleh Grady Booch et al., *Object-Oriented Analysis and Design with Applications*, menjelaskan pendekatan desain berbasis objek yang berguna dalam mendefinisikan class diagram dan relasi antar objek dalam sistem pelacakan obat.

Elgamar (2020) dalam buku *Buku Ajar Konsep Dasar Pemrograman Website dengan PHP* memberikan panduan teknis dalam implementasi web menggunakan bahasa PHP, yang sangat relevan dalam pengembangan aplikasi pelacakan tanggal kedaluwarsa obat.

Terakhir, buku oleh Fitria et al., *Buku Ajar Rekayasa Perangkat Lunak*, memberikan pemahaman menyeluruh mengenai siklus pengembangan perangkat lunak dan teknik rekayasa perangkat lunak modern yang sangat berguna dalam penerapan metode RAD.