

Turnitin

by Turnitin

Submission date: 05-Mar-2025 10:23AM (UTC+0400)

Submission ID: 2605309334

File name: AhMQLNVb5qiKmQZ3ZAmC.docx (171.55K)

Word count: 2292

Character count: 15195

BAB 1

LANDASAN TEORI

1.1 Sistem Pendukung Keputusan (SPK)

Sistem adalah sekumpulan elemen yang memiliki satu tujuan dan menggabungkannya bersama-sama untuk mencapai tujuan tersebut. Sistem juga dapat diartikan sebagai sekumpulan unsur yang saling bergantung dan bersatu untuk bekerja sama dalam mencapai sebuah tujuan (Angelo & Ridho, 2022).

Sistem Pendukung Keputusan (SPK) adalah sistem interaktif berbasis komputer yang membantu pengambil keputusan dalam memanfaatkan data dan model untuk menyelesaikan masalah yang bersifat tidak terstruktur (Widolaras & Ikhsanto, 2022). SPK dirancang untuk menyediakan informasi, pemodelan, dan manipulasi data untuk membantu pengambil keputusan dalam situasi semi terstruktur dan tak terstruktur. SPK bukan alat pengambilan keputusan otomatis, melainkan sistem yang membantu dengan melengkapi informasi yang relevan dan diperlukan untuk membuat keputusan dengan lebih cepat dan akurat (Sumarno & Harahap, 2020).

1.1.1 Karakteristik Sistem Pendukung Keputusan

Merujuk kepada penelitian (Rosita dkk., 2020), ada beberapa karakteristik sistem pendukung keputusan, yaitu:

1. Interaktif

SPK harus memiliki antarmuka yang mudah digunakan agar pengguna dapat mengakses data dan informasi dengan cepat secara efisien.

2. Fleksibel

SPK mampu mengolah berbagai variabel masukan dan menghasilkan berbagai alternatif keputusan yang dapat membantu pengambil keputusan.

3. Data kualitas

SPK diharuskan dapat mengubah data subjektif menjadi bentuk kuantitatif, misalnya dengan memberikan nilai bobot pada aspek-aspek kualitatif seperti keindahan.

4. Prosedur Pakar

SPK perlu menerapkan prosedur yang dikembangkan berdasarkan pengetahuan dan pengalaman para ahli untuk memastikan keputusan yang lebih akurat.

1.1.2 Komponen-Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan (SPK) terdiri dari beberapa komponen utama yang berperan dalam membantuk proses pengambilan keputusan. Menurut (Ariantini dkk., 2023), komponen-komponen utama dalam SPK meliputi:

1. *Kepakaran (Expertise)*

Kepakaran mengacu pada pengetahuan dan keahlian yang diperoleh melalui pengalaman, pendidikan, atau penelitian dalam suatu bidang tertentu. Dalam SPK, kepakaran ini digunakan sebagai dasar dalam proses pengambilan keputusan, termasuk dalam menyusun strategi pemecahan masalah serta menerapkan teori-teori yang relevan.

2. *Pakar (Expert)*

Pakar adalah individu yang memiliki pengalaman, pengetahuan, atau keterampilan khusus dalam suatu disiplin ilmu tertentu. Peran pakar dalam SPK sangat penting, karena mereka memberikan wawasan dan informasi yang akurat untuk mendukung proses analisis dan pengambilan keputusan.

3. *Memilih keputusan terbaik*

Setelah sistem mengevaluasi berbagai alternatif keputusan, langkah berikutnya adalah memilih keputusan yang paling sesuai berdasarkan analisis dan evaluasi yang telah dilakukan. Keputusan ini harus mempertimbangkan tujuan organisasi, keterbatasan yang ada, serta konsekuensi yang mungkin timbul.

4. *Melaksanakan keputusan*

Tahap terakhir dalam proses SPK adalah pelaksanaan keputusan yang telah dipilih. Langkah ini mencakup perencanaan tindakan, pengalokasian sumber daya, serta pemantauan dan evaluasi terhadap implementasi keputusan yang diambil. Jika diperlukan, penyesuaian dan perubahan strategi dapat dilakukan sesuai dengan kondisi yang berkembang.

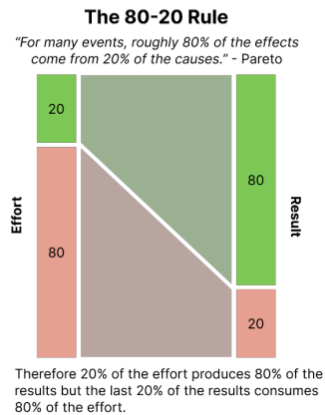
Komponen-komponen diatas bekerja secara terintegrasi dalam mendukung fungsi SPK, sehingga sistem dapat memberikan rekomendasi keputusan yang optimal bagi penggunanya.

1.2 Metode Pareto ABC

Metode Pareto ABC menggabungkan teori Pareto dan analisis ABC untuk mengelompokkan item berdasarkan mengelompokkan barang berdasarkan nilai pembeliannya. Analisis ABC menerapkan prinsip Pareto, yang berfokus pada nilai ekonomis suatu barang.

Prinsip Pareto dalam analisis ABC menekankan pada barang bernilai tinggi dengan jumlah yang lebih sedikit, sehingga tidak efisien jika barang berharga rendah dipantau dengan intensitas yang sama seperti barang berharga tinggi (Firdaus & Hadining, 2023).

Metode Pareto ABC merupakan pendekatan dalam manajemen persediaan yang didasarkan pada Prinsip Pareto 80/20, yang menyatakan bahwa 80% dari total nilai persediaan sering kali berasal dari hanya 20% dari total item yang tersedia (Sunarto, 2020). Prinsip Pareto menunjukkan bahwa 20 persen dari tindakan dan pemikiran dapat dimaksimalkan untuk mencapai 80 persen keberhasilan. Dengan memanfaatkan 20 persen waktu secara optimal, seseorang dapat meraih 80 persen kesuksesan (Saefullah dkk., 2023).



Gambar 1.1 Hukum 80-20
(Sumber: Siswanto, 2020: 3)

Dengan metode Pareto ABC ini, apotek dapat lebih fokus pada obat dengan nilai tinggi untuk meningkatkan efisiensi operasional. Pendekatan ini efektif karena membantu mengoptimalkan sumber daya yang terbatas dan mempermudah pengambilan keputusan terkait pengelolaan stok. Selain itu, metode ini juga membantu dalam perencanaan persediaan, sehingga barang dengan nilai tinggi dapat dipantau dengan lebih ketat dibandingkan barang dengan nilai rendah.

Tahapan dalam metode Pareto ABC dimulai dengan pengumpulan data nilai pembelian atau penggunaan setiap item dalam periode tertentu. Setelah itu, analisis data dilakukan dengan Langkah-langkah sebagai berikut (Firdaus & Hadining, 2023):

1. Menghitung jumlah pemakaian barang selama satu periode.
2. Membuat daftar harga barang yang telah digunakan.
3. Mengalikan setiap pemakaian barang dengan harga setiap barang.
4. Mengurutkan setiap barang dari yang mempunyai nilai rupiah pemakaian tertinggi hingga nilai rupiah pemakaian terendah.
5. Menghitung kumulatif untuk keseluruhan barang.
6. Menghitung hasil presentase kumulatif untuk setiap barang dengan persamaan berikut:

$$\text{Presentase Kumulatif} = \frac{\text{Nilai kumulatif barang}}{\text{Total nilai kumulatif}} \times 100$$
7. Setiap barang persediaan dikelompokkan berdasarkan hasil presentase kumulatif.
8. Barang dengan nilai presentase kumulatif 0 – 80% dikelompokkan ke dalam Kategori A. Kemudian barang dengan nilai presentase kumulatif 81 – 95% dikelompokkan ke dalam Kategori B. Dan barang dengan nilai presentase kumulatif 96 – 100% dikelompokkan ke dalam Kategori C.

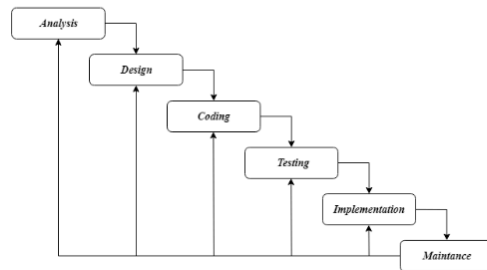
1.3 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) adalah pendekatan sistematis yang diterapkan dalam pengembangan perangkat lunak. Pendekatan ini membantu tim pengembangan dalam mengelola proses pembuatan, pengembangan, pengujian, dan pemeliharaan perangkat lunak dengan cara yang efisien dan terstruktur. Tujuan dari SDLC adalah untuk menghasilkan perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna, mematuhi tenggat waktu, dan tetap dalam batas anggaran yang telah ditetapkan (Permana dkk., 2023).

Beberapa model yang dapat dijadikan referensi sebagai model SDLC yang efektif antara lain adalah *Waterfall*, *Spiral*, *Rapid Application Development (RAD)*, *Prototype*, dan *Agile* (Permana dkk., 2023). Setiap model ini memiliki pendekatan dan karakteristik yang berbeda, sehingga dapat dipilih sesuai dengan kebutuhan proyek dan tim pengembangan. Pemilihan model yang tepat sangat penting untuk memastikan bahwa proses pengembangan perangkat lunak berjalan lancar dan menghasilkan produk yang sesuai dengan harapan pengguna.

13.1 Waterfall

Model *Waterfall* adalah salah satu model SDLC yang umum digunakan, sering disebut sebagai model reguler atau siklus hidup klasik dalam SDLC. Model ini menerapkan pendekatan yang sistematis dan berurutan, dimulai dari pengumpulan kebutuhan sistem, kemudian dilanjutkan dengan analisis, desain, pengkodean, pengujian/validasi, dan pemeliharaan (Adi Kurniyanti & Murdiani, 2022).



Gambar 1.2 Tahapan *Waterfall*
(Sumber: Gunawan & Diwiryono, 2020 : 2)

Metode *waterfall* dilakukan dengan tahapan sebagai berikut (Alif Ramadhan dkk., 2023) :

1. Perencanaan Konsep (*Requirement Analysis*)

Tahap ini bertujuan untuk menganalisis dan memahami kebutuhan pelanggan. Data dikumpulkan melalui wawancara langsung dengan pemangku kepentingan. Hasilnya berupa analisis kebutuhan sistem, yang mencakup semua persyaratan pengembangan perangkat lunak, serta spesifikasi kebutuhan sistem yang terdokumentasi.

2. Pemodelan sistem (*System Design*)

Pada tahap ini, analisis kebutuhan sistem yang telah dibuat sebelumnya dikembangkan menjadi desain sistem sebagai dasar untuk proses pengkodean.

3. Implementasi

Pada tahap ini, aktivitas yang dilakukan adalah pengkodean sistem. Penulisan kode program merupakan proses penerjemahan desain sistem yang telah dibuat ke dalam bentuk perintah yang dapat dipahami oleh komputer, menggunakan bahasa pemrograman. Tahapan ini merupakan fase nyata dalam pelaksanaan suatu sistem.

4. Pengujian

Tahapan pengujian (*testing*) berfokus pada perangkat lunak dari aspek logika dan fungsional, serta memastikan bahwa semua komponen telah diuji.

5. Pemeliharaan

Tahap pemeliharaan dilakukan ketika terdapat kerusakan pada sistem.

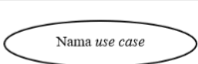


1.4 Unified Modeling Language (UML)


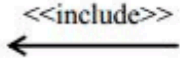
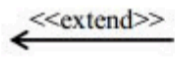
UML (*Unified Modeling Language*) adalah bahasa pemodelan perangkat lunak yang telah distandardisasi untuk membuat cetak biru (*blueprint*) dalam pengembangan perangkat lunak (Sumiati dkk., 2021). UML dirancang untuk mendukung berbagai pemodelan sistem perangkat lunak dengan menyediakan berbagai konstruksi untuk sistem dan aktivitas. Saat ini, UML merupakan bahasa pemodelan perangkat lunak yang paling sukses dan telah distandardisasi oleh *Object Management Group* (OMG). Oleh karena itu, UML sangat sesuai untuk semua tahap siklus hidup perangkat lunak, mulai dari analisis kebutuhan (Wayahdi & Ruziq, 2023).

1.4.1 Use Case Diagram

Use case diagram adalah model hasil analisis dan perancangan sistem yang digunakan untuk mendeskripsikan kebutuhan sistem. Kebutuhan ini akan diterapkan oleh pengguna, sehingga perancangan sistem dapat tergambarkan dengan jelas. Diagram ini memvisualisasikan interaksi antara pengguna (aktor) dan sistem, sehingga dapat memberikan gambaran yang jelas tentang konteks serta batasan sistem (Ramdany dkk., 2024).

Tabel 1.1 Use Case Diagram




No	Simbol	Nama	Deskripsi
1.		Use Case	Abstraksi dan interaksi antara sistem dan aktor
2.		Aktor	Mewakili peran orang, sistem yang lain, atau alat Ketika berkomunikasi dengan use case
3.		Association	Abstraksi dari penghubung antara aktor dengan use case



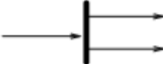

4.		Generalisasi	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
5.		Include	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
6.		Extend	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

1.4.2 Activity Diagram

Activity diagram menggambarkan alur fungsional dalam suatu sistem informasi. Diagram ini menunjukkan titik awal dan akhir workflow, aktivitas yang terjadi selama proses, serta urutan kejadian dari setiap aktivitas. Selain itu, *activity diagram* juga memungkinkan pemodelan proses yang berjalan secara paralel. Bagi yang terbiasa dengan analisis dan desain struktur tradisional, diagram ini menggabungkan konsep dari diagram alir data dan diagram alur sistem (Ramdany dkk., 2024).

Tabel 1.2 Activity Diagram

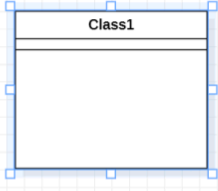
No	Notasi	Nama	Deskripsi
1.		Start Point	Menyatakan bahwa sebuah objek dibentuk atau diakhiri.
2.		Activity	Menyatakan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
3.		Decision	Menunjukkan penggambaran suatu


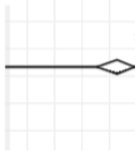

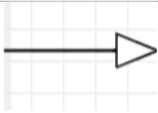

			keputusa/Tindakan yang harus diambil pada kondisi tertentu.
4.		<i>Control Flow</i>	Menunjukkan urutan eksekusi
5.		<i>Object Flow</i>	Menunjukkan aliran objek dari sebuah <i>action</i> atau <i>activity</i> ke <i>action</i> .
6.		<i>Fork</i>	Menyatakan untuk memcah <i>behavior</i> menjadi <i>activity</i> atau <i>action</i> yang parallel.
7.		<i>End Point</i>	Menyatakan bahwa sebuah objek dibentuk atau akhiri.

1.4.3 ⁶Class diagram

Class diagram menggambarkan hubungan antar kelas serta detail setiap kelas dalam model desain sistem, termasuk aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Diagram ini merepresentasikan struktur sistem serta alur *database* yang akan dikembangkan. *Class diagram* terdiri dari kumpulan kelas beserta relasinya, di mana setiap kelas direpresentasikan dalam bentuk persegi yang berisi nama kelas, atribut, dan metode. Sebagai inti dari pengembangan berbasis objek, *class diagram* menunjukkan spesifikasi yang, ketika diinstansiasi, akan menghasilkan objek (Ramdany dkk., 2024).

Tabel 1.3 *Class Diagram*




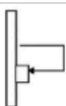


No	Notasi	Nama	Deskripsi
1.		<i>Class</i>	Class merupakan komponen utama dalam pemrograman berorientasi objek. Class direpresentasikan sebagai sebuah kotak yang terbagi menjadi tiga bagian.

2.		Association	Association merupakan hubungan yang menunjukkan interaksi antar class. Hubungan ini direpresentasikan dengan garis yang memiliki mata panah terbuka di ujungnya, menandakan aliran pesan satu arah.
3.		Aggregation	Aggregation menunjukkan hubungan antara keseluruhan dan bagian, yang sering disebut sebagai relasi
4.		¹ Composition	Jika sebuah class tidak dapat berdiri sendiri dan harus menjadi bagian dari class lain, maka class tersebut memiliki relasi Composition dengan class yang menjadi tempat ketergantungannya.
5.		Generalization	¹ Generalization merupakan hubungan antar class yang menunjukkan hubungan dari yang khusus ke yang lebih umum.
6.		Dependency	Dependency menunjukkan bahwa suatu class bergantung pada class lain dalam menjalankan operasinya.

1.4.4 Sequence diagram

Sequence diagram membantu memahami persyaratan sistem baru, mendokumentasikan proses, dan memvisualisasikan skenario teknis saat sistem berjalan (runtime). Diagram ini memungkinkan pengguna memahami serta memprediksi perilaku sistem (Rohmanto & Setiawan, 2022).

Tabel 1.4 ⁴*Sequence Diagram*

No	Notasi	Nama	Deskripsi
1.		<i>Entity Class</i>	Gambaran sistem sebagai landasan dalam Menyusun basis data.
2.		<i>Boundary Class</i>	Menangani komunikasi antar lingkungan sistem.
3.		<i>Control Class</i>	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika.
4.		<i>Recursive</i>	Pesan untuk dirinya.
5.		<i>Activation</i>	Mewakili proses durasi aktivasi sebuah operasi.
6.		<i>Life line</i>	Komponen yang digambarkan garis putus terhubung dengan objek.

1.5 Database

Database adalah kumpulan data terstruktur yang saling terkait, disimpan dalam media penyimpanan komputer, dan dikelola menggunakan perangkat lunak khusus. *Database* berfungsi untuk menyimpan, mengelola, dan mengorganisir data secara efisien, aman, serta terstruktur. Umumnya, *database* dikelola oleh sistem manajemen *database* (DBMS) untuk memastikan keterhubungan dan pengelolaan data yang optimal (Syahputri dkk., 2023).

1.6 MySQL

MySQL adalah *database server* populer yang termasuk dalam kategori *Relational Database Management System* (RDBMS). MySQL mendukung bahasa SQL, yang memiliki aturan standar yang ditetapkan oleh ANSI. Sebagai RDBMS, MySQL memungkinkan pengguna untuk membuat, mengelola, dan mengakses data dalam model relasional, di mana tabel-tabel dalam *database* saling terhubung satu sama lain (Hermiati dkk., 2021).

1.7 Hypertext Preprocessor (PHP)

Hypertext Preprocessor (PHP) adalah bahasa pelengkap HTML yang digunakan untuk membuat aplikasi dinamis dengan kemampuan pengolahan dan pemrosesan data. Semua sintaks PHP dieksekusi di server, sementara yang dikirim ke browser hanyalah hasilnya. Sebagai bahasa scripting yang berjalan di server, PHP menghasilkan konten yang dikirim ke client melalui browser. PHP juga menyatu dengan tag HTML dan digunakan untuk membuat halaman web dinamis, seperti *Active Server Pages* (ASP) atau *Java Server Pages* (JSP). Selain itu, PHP bersifat *Open Source* (Hermiati dkk., 2021).

1.8 HyperText Markup Language (HTML)

HTML (HyperText Markup Language) adalah bahasa yang digunakan untuk membangun struktur dasar halaman website. HTML menggunakan tag `< >` untuk menuliskan kode yang akan ditafsirkan oleh browser agar halaman ditampilkan sesuai dengan pengaturan. Sebagai pondasi awal dalam pengembangan website, HTML membantu menyusun kerangka halaman sebelum masuk ke tahap desain dan fungsionalitas. HTML juga dapat dikombinasikan dengan CSS untuk mempercantik tampilan website (Sari dkk., 2022).

1.9 Cascading Style Sheet (CSS)

CSS (Cascading Style Sheet) adalah bahasa yang digunakan untuk mengatur tampilan desain website, seperti warna, font, latar belakang, dan penyesuaian dengan ukuran layar. Dalam pengembangan website, CSS berkolaborasi dengan HTML untuk menciptakan tampilan yang lebih menarik dan responsif.

1.10 Bootstrap

Bootstrap adalah salah satu *framework* CSS paling populer yang digunakan untuk membangun tampilan (*frontend*) situs web atau aplikasi web. *Framework* ini menyediakan berbagai komponen dan gaya siap pakai, seperti *grid system*, tipografi, *form*, tombol, *navbar*, modal, dan lainnya (Wahyu Rhamadani dkk., 2023).

1.11 Black Box Testing

Blackbox Testing adalah metode pengujian *software* tanpa memperhatikan detail internalnya. Pengujian dilakukan dengan memasukkan data ke dalam setiap form untuk memastikan program berjalan sesuai kebutuhan perusahaan (Made dkk., 2021).

Menurut (Rifqi dkk., t.t.) *Black Box* digunakan dalam pengujian data untuk mendeteksi penyimpangan atau kesalahan dalam berbagai kategori.

- a. Kesalahan fungsi atau elemen yang hilang.
- b. Kesalahan pada tampilan antarmuka.
- c. Kesalahan dalam *database* eksternal.
- d. Kesalahan selama pengujian kinerja.
- e. Kesalahan yang menyebabkan terminasi sistem.

1.12 Apotek

Apotek merupakan salah satu lembaga kesehatan yang berfokus pada penyediaan layanan farmasi untuk masyarakat secara luas (Oktaviani & Sumarlinda, 2021). Menurut Menteri kesehatan Republik I tahun 2016, apotek adalah tempat dimana apoteker menjalankan praktik kefarmasian, didukung oleh apoteker pendamping atau tenaga teknis kefarmasian (Wahyuni dkk., 2020).

22%

SIMILARITY INDEX

21%

INTERNET SOURCES

6%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1	ejurnal.ubharajaya.ac.id Internet Source	4%
2	repository.uncp.ac.id Internet Source	3%
3	jurnal.sttkd.ac.id Internet Source	3%
4	eprints.upj.ac.id Internet Source	2%
5	repository.penerbiteureka.com Internet Source	2%
6	repository.usahidsolo.ac.id Internet Source	2%
7	journal.ipm2kpe.or.id Internet Source	1%
8	M. Rhifky Wayahdi, Fahmi Ruziq. "Pemodelan Sistem Penerimaan Anggota Baru dengan Unified Modeling Language (UML) (Studi Kasus: Programmer Association of Battuta)", Jurnal Minfo Polgan, 2023 Publication	1%
9	Submitted to Universitas Muria Kudus Student Paper	1%
10	digilib.unila.ac.id Internet Source	1%

Exclude quotes	On	Exclude matches	< 30 words
Exclude bibliography	On		