

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Berikut ini merupakan tinjauan empiris mengenai penelitian serupa yang telah dilakukan dalam lima tahun terakhir.

Penelitian pertama oleh Zuhairunisa, Az-Zahra, & Syawli (2025), dalam jurnal “Penerapan Metode Moscow dalam Menetapkan Prioritas Kebutuhan Sistem di SMPN 1 Kedawung”. Fokus penelitian ini adalah pada peningkatan efisiensi operasional dan pengelolaan data akademik serta administrasi sekolah. Dengan menggunakan MoSCoW, peneliti berhasil mengelompokkan fitur sistem ke dalam empat kategori prioritas, sehingga pengembangan sistem dapat dilakukan secara bertahap dan terfokus. Hasil penelitian ini menunjukkan bahwa MoSCoW sangat efektif dalam mengidentifikasi kebutuhan kritis dan mendesak, serta membantu dalam alokasi sumber daya secara optimal.

Penelitian kedua oleh Kusuma & Ikhsan (2022) dalam jurnal “Elisitasi Kebutuhan Pengguna Untuk Meningkatkan Ekspektasi Pengguna Terhadap *Learning Management System*” menyoroti pentingnya proses elisitasi kebutuhan secara partisipatif untuk meningkatkan pengalaman pengguna LMS. Melalui

wawancara online dan analisis data pengguna, ditemukan bahwa fitur seperti notifikasi batas waktu tugas dan diskusi real-time sangat dibutuhkan namun belum tersedia. Penelitian ini menghasilkan rekomendasi pengembangan LMS yang lebih sesuai dengan harapan pengguna, membuktikan bahwa elisitasi kebutuhan efektif dalam meningkatkan kualitas sistem pembelajaran digital.

Penelitian ketiga dilakukan oleh Handriani & Wicaksono (2024), dalam jurnal "*Dokumen Software Requirement Specification Sistem Informasi Akuntansi Biaya (Studi Kasus PT. ABC)*". Melalui wawancara dan observasi dokumen internal, peneliti merumuskan kebutuhan fungsional (input transaksi biaya bahan baku, perhitungan otomatis, dan laporan biaya produksi) serta non- fungsional (kemudahan antarmuka, kecepatan, dan keamanan data). SRS terbukti efektif sebagai panduan teknis bagi tim pengembang agar sistem memenuhi harapan pengguna dan standar perusahaan.

Penelitian Keempat oleh Ratmoko et al. (2023), dalam jurnal "*Pemodelan Sistem Informasi Berbasis Web Menggunakan Metode Waterfall Dalam Pengelolaan Data Sistem Informasi Akademik (SIKAD) Sekolah Tinggi Ilmu Pelayaran Jakarta*", menunjukkan bahwa sistem SIKAD yang berbasis web di Sekolah Tinggi Ilmu Pelayaran Jakarta telah menerapkan metode *Waterfall* dengan efektif dalam pengelolaan data akademis. Sistem ini dapat menggantikan

cara manual yang sebelum ada, sehingga pengelolaan data menjadi lebih cepat, terstruktur, dan tepat. Pengujian, terutama melalui metode *black box*, menunjukkan bahwa sistem ini dapat memenuhi kebutuhan serta fitur yang diinginkan, sekaligus memberikan akses yang mudah bagi pengguna, baik mahasiswa maupun dosen, terhadap informasi seperti jadwal kuliah, kehadiran, dan nilai.

Penelitian kelima oleh Kasabera et al. (2022), dalam jurnal “*An Automated Approach To Validate Requirements Specification*”. Menyoroti pentingnya validasi spesifikasi kebutuhan sebelum implementasi. Peneliti mengusulkan metode validasi otomatis berbasis logika formal dan heuristik untuk mendeteksi inkonsistensi dan ambiguitas. Meskipun berfokus pada otomatisasi, prinsip utamanya tetap relevan, yaitu memastikan validitas kebutuhan sistem.

Berbeda dengan penelitian sebelumnya yang difokuskan pada sekolah, perusahaan, atau platform pembelajaran online, penelitian ini memberikan kontribusi di bidang perguruan tinggi kesehatan, khususnya pada pengembangan Sistem Informasi Akademik (SIKA) di STIKES PEMKAB Jombang. Metode yang digunakan mengkombinasikan elicitasi lewat wawancara dengan model MoSCoW untuk mengurutkan kebutuhan berdasarkan prioritas, yang

kemudian didokumentasikan dalam bentuk spesifikasi kebutuhan perangkat lunak. Dengan demikian, penelitian menghasilkan output yang lebih praktis dan sesuai konteks, serta dapat langsung menjadi landasan untuk pengembangan sistem akademik yang memenuhi tuntutan institusi.

**Tabel 2.1 Rekapitulasi Penelitian Terdahulu**

No	Peneliti & Tahun	Tujuan	Ruang Lingkup	Hasil Penelitian	Perbedaan dengan Penelitian yang Akan Dilakukan
1	Zuhairunisa, Az-Zahra, Syawli (2025)	Untuk memprioritaskan kebutuhan sistem informasi akademik menggunakan metode MoSCoW di SMPN 1 Kedawung berdasarkan urgensi, manfaat, dan biaya	Sistem informasi sekolah Tingkat SMP, focus pada fitur seperti e-raport, presensi, e-learning, perpustakaan, dan pengumuman sekolah.	Pengelompokan fitur ke dalam kategori MoSCoW (must have, should have, could have, won't have) dan roadmap implementasi pengembangan sistem.	Penelitian pada jurnal ini dilakukan di tingkat SMP, sedangkan penelitian yang akan dilakukan dilakukan di pendidikan tinggi (STIKES). Selain itu, penelitian yang akan dilakukan juga memiliki fokus yang lebih luas pada sistem informasi akademik kampus dan menggunakan model pengembangan waterfall.
2	Ikhsan, Kusuma (2022)	Mengidentifikasi kebutuhan pengguna untuk meningkatkan ekspektasi terhadap Learning Management System (LMS), khususnya dalam pembelajaran jarak jauh	Kebutuhan mahasiswa dan dosen terhadap LMS dalam pembelajaran jarak jauh. Fokusnya adalah pada pengembangan fitur seperti notifikasi dan pemisah tugas, berdasarkan wawancara dan analisis kebutuhan.	Mahasiswa dan dosen mengalami kendala dalam penggunaan LMS selama pembelajaran jarak jauh. Solusinya adalah pengembangan fitur notifikasi dan pemisah tugas, yang membantu mahasiswa memantau progres tugas dan dosen memantau akses materi.	Penelitian pada jurnal ini dilakukan untuk sistem <i>Learning Management System</i> (LMS) di tingkat perguruan tinggi, dengan fokus pada interaksi pembelajaran online dan pengalaman pengguna. Sementara penelitian yang akan dilakukan fokus pada Sistem Informasi Akademik (SIKAD) di STIKES PEMKAB Jombang, serta menggunakan model MoSCoW untuk menentukan kebutuhan prioritas, bukan hanya elisitasi biasa
3	Handriani & Wicaksono (2024)	Membuat dokumen Software Requirement Specification (SRS) untuk sistem informasi akuntansi biaya di PT. ABC sebagai dasar pengembangan sistem informasi yang lebih terarah, terintegrasi, dan sesuai kebutuhan perusahaan.	Sistem informasi akuntansi biaya manufaktur, fokus pada pencatatan biaya produksi, transaksi barang, laporan biaya, dan integrasi proses bisnis manual ke digital	Dokumen SRS disusun sesuai standar ISO/IEC/IEEE mencakup kebutuhan fungsional, antarmuka, dan Performa sistem akuntansi biaya manufaktur. Hasil ini menjadi dasar desain sistem yang efisien untuk mendukung pengambilan keputusan.	Penelitian dalam jurnal ini focus pada system informasi akuntansi biaya di perusahaan manufaktur dan pembuatan SRS nya menggunakan standar ISO/IEC/IEEE 29148:2018. Sedangkan penelitian yang akan dilakukan focus pada Sistem Informasi Akademik (SIKAD) di institusi pendidikan dan penelitiannya menggunakan metode MoSCoW untuk memprioritaskan kebutuhan pengguna.

No	Peneliti & Tahun	Tujuan	Ruang Lingkup	Hasil Penelitian	Perbedaan dengan Penelitian yang Akan Dilakukan
4	Ratmoko, Laisa, Hidayatullah, Moningka, Gusti (2023)	Membangun dan mengimplementasikan sistem informasi akademik berbasis web untuk meningkatkan efisiensi dan aksesibilitas data akademik menggunakan metode Waterfall	Sistem informasi akademik pada STIP Jakarta, meliputi data pengguna, mata kuliah, jadwal, KRS, presensi, nilai, dan laporan perkuliahan	Sistem SIAKAD berhasil dikembangkan dan diuji dengan metode <i>black-box</i> , dan memberikan kemudahan akses bagi dosen serta mahasiswa dalam pengelolaan data akademik.	Penelitian ini fokus pada pengembangan sistem secara teknis dan implementasi fitur berbasis web, sedangkan penelitian yang akan dilakukan menitikberatkan pada analisis prioritas kebutuhan perangkat lunak menggunakan metode MoSCoW dalam tahap awal pengembangan SIAKAD. Namun, kedua penelitian ini menggunakan metode pengembangan Waterfall dan konteksnya sama-sama di lingkungan pendidikan tinggi.
5	Kasabera, Alzyadat, Alhroob, Showarah, Thunibat (2020)	Mengembangkan pendekatan otomatis untuk memvalidasi spesifikasi kebutuhan perangkat lunak agar benar, lengkap, dan konsisten, menggunakan diagram use case UML dan tools terintegrasi	Validasi otomatis dokumen kebutuhan perangkat lunak berbasis use case UML. Fokus pada aspek kebenaran, kelengkapan, dan konsistensi menggunakan studi kasus sistem belanja online.	Pendekatan otomatis berhasil mendeteksi kebutuhan yang tidak lengkap, tidak konsisten, dan salah dalam dokumen SRS. Pengujian pada <i>use case "Online Shopping"</i> menunjukkan peningkatan akurasi dan efisiensi validasi, serta membantu pengembang memperbaiki kesalahan secara cepat.	Penelitian ini fokus pada metode validasi otomatis berdasarkan aturan formal dan heuristic untuk mendeteksi ambiguitas dalam dokumen kebutuhan perangkat lunak, sedangkan penelitian yang akan dilakukan menggunakan validasi manual oleh pemangku kepentingan (Wakil Ketua I, Wakil Ketua II, dan Pakar)

## 2.2 Teori Terkait

### 2.2.1 Sistem Informasi Akademik (SIKAD)

Sistem Informasi akademik (SIKAD) merupakan sebuah platform layanan akademik yang dibuat untuk mempermudah mahasiswa dalam mendapatkan berbagai informasi akademis selama proses belajar. Layanan ini mencakup akses ke Kartu Rencana Studi (KRS), Kartu Hasil Studi (KHS), jadwal kuliah yang sedang diikuti, data pribadi mahasiswa, dan juga agenda harian untuk perkuliahan. Penggunaan teknologi informasi memiliki peran yang signifikan sebagai alat bantu untuk mencapai tujuan organisasi. Namun, efektivitas penggunaannya akan maksimal jika semua anggota organisasi memiliki kemampuan dan keterampilan yang sesuai dalam menggunakan teknologi tersebut. *(Taufandri et al., (2022)).*

Tujuan utama dari SIKAD adalah mendukung penyelenggaraan pendidikan dengan memberikan layanan informasi yang lebih baik kepada civitas akademika baik di dalam maupun di luar kampus.

SIKAD juga memiliki beberapa fungsi utama yang sangat penting dalam pengelolaan informasi akademik, diantaranya:

- a. SIAKAD berfungsi untuk mengelola data mahasiswa secara terpusat dan saling terintegrasi. mencakup informasi pribadi, data akademik, serta data keuangan mahasiswa.
- b. SIAKAD mendukung proses penyusunan dan pengelolaan jadwal kuliah, sehingga memudahkan mahasiswa dan dosen dalam memperoleh informasi jadwal perkuliahan
- c. Melalui sistem SIAKAD, mahasiswa dapat melakukan pemilihan mata kuliah yang akan diambil secara daring, yang dikenal sebagai layanan KRS Online.

Dengan demikian, SIAKAD memiliki peran strategis dalam meningkatkan kualitas layanan akademik dan efisiensi proses administrasi di perguruan tinggi, karena sistem yang terintegrasi, cepat diakses, dan andal mampu meningkatkan kepuasan civitas akademika secara signifikan.

*(Saputri & Mulyani, 2022).*



### 2.2.2 Pengembangan Perangkat Lunak

Pengembangan perangkat lunak adalah proses yang melibatkan serangkaian tahapan, termasuk penelitian, perencanaan, desain, pengembangan, pengujian, dan pemeliharaan produk perangkat lunak. Metodologi pengembangan perangkat lunak mencakup seperangkat aturan dan pedoman yang digunakan dalam proses ini, serta nilai-nilai inti yang dijunjung oleh tim proyek dan alat yang digunakan dalam perencanaan, pengembangan, dan implementasi (DESPA, 2014).

Prinsip-prinsip pengembangan perangkat lunak yang dapat membantu memastikan keberhasilan proyek dan memenuhi kebutuhan pengguna:

- a. Pengguna harus terlibat dalam setiap tahap pengembangan untuk memastikan bahwa hasil akhir pengembangan proyek memenuhi kebutuhan mereka.
- b. Setiap proyek harus dimulai dengan perencanaan yang jelas untuk mengidentifikasi tujuan, ruang lingkup, dan sumber daya yang diperlukan.
- c. Proses pengembangan harus bersifat iteratif.

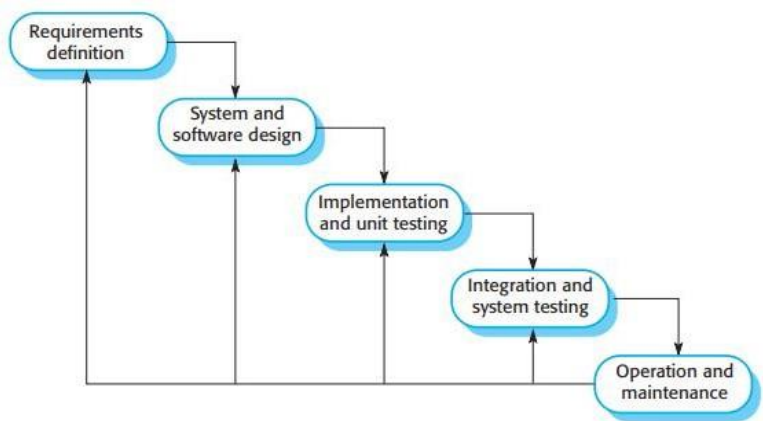
- d. Dokumentasi yang jelas dan terperinci sangat penting untuk memfasilitasi pemeliharaan dan pengembangan di masa depan.

*Software Development Life Cycle* (SDLC) didefinisikan sebagai serangkaian tahapan yang diakui secara umum dalam pengembangan perangkat lunak. SDLC berfungsi sebagai kerangka kerja yang membantu dalam mengatur dan mengelola proses pengembangan perangkat lunak, memastikan bahwa setiap aspek dari pengembangan diperhatikan dan dikelola dengan baik (DESPA, 2014).

SDLC memberikan kerangka kerja yang sistematis untuk pengembangan perangkat lunak, membantu memastikan bahwa perangkat lunak yang dihasilkan berkualitas tinggi dan memenuhi kebutuhan pengguna. Dengan mengikuti tahapan ini, tim pengembang dapat mengelola proyek perangkat lunak dengan lebih baik dan meminimalkan risiko kegagalan.

Metode dan pendekatan dalam pengembangan perangkat lunak yang juga digunakan untuk pengembangan SIAKAD, yakni:

- a. **Waterfall** adalah metode pengembangan perangkat lunak yang mengatur proses urutan dalam urutan yang jelas dan terstruktur. Setiap tahap, seperti analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan, harus diselesaikan sebelum melanjutkan ke tahap berikutnya (Sommerville, 2011) .

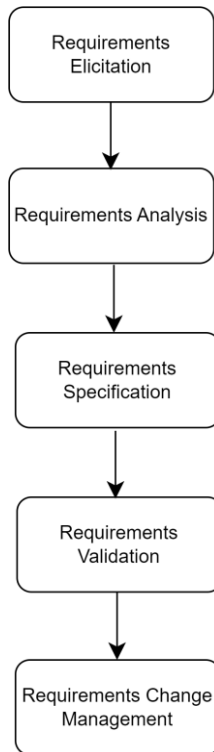


Gambar 2.1. Tahapan Pengembangan Metode Waterfall  
Sumber: Ian Sommerville (2016 : 47)

Metode waterfall sangat cocok untuk pengembangan SIAKAD karena memiliki jalur kerja yang sistematis dan terstruktur. Analisis kebutuhan, perancangan, implementasi, dan pengujian adalah semua tahap pengembangan yang dilakukan secara bertahap dan harus diselesaikan sebelum melanjutkan ke tahap berikutnya. Metode ini memudahkan tim pengembang untuk membuat dokumentasi yang lengkap dan mengurangi kemungkinan perubahan diperlukan selama proses. Metode waterfall memungkinkan pengembangan dilakukan dengan lebih terkontrol dan sesuai dengan standar prosedur institusi pendidikan karena sistem akademik biasanya memiliki alur kerja yang jelas dan persyaratan yang relatif stabil.

### **2.2.3 Pengelolaan Kebutuhan Perangkat Lunak**

Pengelolaan kebutuhan perangkat lunak menurut Sommerville, mencakup beberapa tahapan kunci yang sangat penting untuk memastikan bahwa perangkat lunak yang dikembangkan memenuhi kebutuhan pengguna dan pemangku kepentingan. Berikut merupakan tahapan pengelolaan kebutuhan perangkat lunak menurut Sommerville yang diuraikan dalam bukunya *Software Engineering (10th ed.)*.



Gambar 2.2. Pengelolaan Kebutuhan Perangkat Lunak menurut Sommerville

Adapun penjelasannya sebagai berikut:

- a. **Elicitasi** Kebutuhan (Requirement Elicitation), menurut Sommerville, elicitation adalah fase kritis dalam proses rekayasa kebutuhan yang bertujuan untuk mengumpulkan dan memahami kebutuhan dari para pemangku kepentingan.

Sommerville juga menjelaskan bahwa elicitation adalah proses yang berpusat pada manusia, karena memerlukan interaksi langsung dengan para pemangku kepentingan untuk memahami kebutuhan, dan preferensi penggunaanya.

- b. **Analisis Kebutuhan (*Requirements Analysis*)**, adanya fase analisis kebutuhan menurut Sommerville adalah untuk memastikan bahwa semua kebutuhan yang telah dikumpulkan dianalisis secara menyeluruh dan dipahami dengan baik. Fase ini bertujuan untuk mengidentifikasi, mengklarifikasi, dan mendokumentasikan kebutuhan sistem secara sistematis, sehingga kebutuhan yang diidentifikasi menjadi lengkap, dan dapat dipahami oleh semua pihak yang terlibat.
- c. **Spesifikasi Kebutuhan (*Requirements Specification*)**, adalah proses mendokumentasikan kebutuhan pengguna dan sistem dalam bentuk dokumen kebutuhan. Dokumen ini berfungsi sebagai acuan untuk pengembangan perangkat lunak dan harus mencakup semua kebutuhan yang telah diidentifikasi dan dianalisis pada tahap sebelumnya.

- d. **Validasi Kebutuhan (*Requirements Validation*)**, menurut Sommerville, validasi kebutuhan (*Requirements Validation*) adalah proses penting dalam rekayasa kebutuhan yang bertujuan untuk memastikan bahwa kebutuhan yang telah diidentifikasi dan didokumentasikan memenuhi kriteria tertentu dan dapat diterima oleh semua pemangku kepentingan.
- e. **Manajemen Perubahan Kebutuhan (*Requirements Change Management*)**, menurut Sommerville *Requirements Change Management* adalah salah satu proses dalam pengelolaan kebutuhan perangkat lunak yang sangat diperlukan karena kebutuhan dapat berubah seiring waktu akibat berbagai faktor, seperti perubahan dalam lingkungan organisasi, atau ditemukannya bug yang harus diperbaiki. *Requirements Change Management* bertujuan untuk memastikan bahwa perubahan tersebut dikelola dengan cara yang efisien dan terkontrol. (*Sommerville, 2006*).

#### 2.2.4 Validasi Kebutuhan Perangkat Lunak & Peran Validator

Validasi kebutuhan menurut Ian Sommerville adalah proses untuk memastikan bahwa kebutuhan yang didefinisikan benar-benar mencerminkan apa yang diinginkan oleh pelanggan atau pemangku kepentingan. Ini penting agar proyek yang dikembangkan sesuai dengan kebutuhan nyata, tujuan bisnis, dan harapan pengguna.

Tujuan validasi kebutuhan adalah untuk memastikan kesesuaian dengan kebutuhan pengguna dan pemangku kepentingan, sehingga kebutuhan yang dikumpulkan sesuai ekspektasi dari segi fungsi dan performa. Proses ini juga bertujuan mengidentifikasi serta mengurangi kesalahan, ambiguitas, atau ketidaklengkapan sejak awal sebelum pengembangan lebih lanjut. Dengan kebutuhan yang valid dan jelas, perangkat lunak yang dikembangkan akan lebih stabil, andal, dan sesuai spesifikasi.

Metode validasi kebutuhan menurut Ian Sommerville:

- a. **Inspeksi (*Requirement Review*)**, Pemeriksaan dokumen kebutuhan secara sistematis oleh tim yang terdiri dari analis, pengembang, pengguna, dan pihak terkait lainnya.



- Tujuannya adalah menemukan kesalahan, ketidakkonsistenan, ambiguitas, atau bagian yang kurang jelas dalam dokumen kebutuhan sebelum tahap pengembangan dimulai. Metode ini biasanya dilakukan melalui rapat tinjauan (*review meeting*) yang terstruktur dengan daftar pemeriksaan (*checklist*).
- b. **Prototyping**, Membuat model awal (*prototype*) dari sistem yang akan dikembangkan untuk memberikan gambaran nyata kepada pengguna tentang bagaimana sistem akan bekerja. *Prototyping* membantu mengklarifikasi kebutuhan yang masih samar, memvalidasi fungsionalitas yang direncanakan, dan memungkinkan pengguna memberikan umpan balik langsung sebelum sistem dibangun secara penuh.
- c. **Test-case Generation**, Mengembangkan *test case* atau skenario pengujian berdasarkan kebutuhan yang telah ditentukan. Metode ini digunakan untuk memeriksa apakah setiap kebutuhan dapat diuji secara teknis (*verifiable*) dan realistis untuk diimplementasikan. Jika kebutuhan sulit atau tidak mungkin diuji, maka kebutuhan tersebut perlu diperjelas atau direvisi.

### 2.2.5 Model MoSCoW

Model MoSCow merupakan salah satu pendekatan yang digunakan untuk mengelompokkan sistem kebutuhan berdasarkan tingkat prioritasnya. Metode ini membantu tim pengembang dalam menentukan fitur mana yang harus diprioritaskan dan dikerjakan terlebih dahulu, serta menjelaskan bagaimana penyelesaian fitur tersebut dapat mendukung pencapaian tujuan organisasi, seperti meningkatkan efisiensi operasional, mengurangi biaya, meningkatkan produktivitas, serta meningkatkan kepuasan pengguna (Zuhairunisa, et al, 2025).

Menurut mekanisme MoSCoW, daftar kebutuhan diklasifikasikan ke dalam empat kategori prioritas, diantaranya:

- a. **Must Have (M):** Kebutuhan yang harus ada untuk sistem dapat berfungsi. Tanpa fitur ini, sistem tidak dapat memenuhi tujuan utamanya. Contoh, fitur pendaftaran mahasiswa, pengelolaan data akademik, dan laporan hasil studi.

- b. ***Should Have (S)***: Kebutuhan yang penting tidak kritis, fitur ini diinginkan dan sebaiknya ada dalam sistem, tetapi dapat ditunda jika waktu atau sumber daya terbatas. Contoh, fitur integrasi dengan sistem lain atau laporan tambahan yang tidak mendesak.
- c. ***Could Have (C)***: Kebutuhan yang diinginkan tetapi tidak mendesak. Fitur ini dapat ditambahkan jika ada waktu dan sumber daya yang tersedia, tetapi tidak mempengaruhi operasi utama sistem.
- d. ***Won't Have (W)***: Kebutuhan yang tidak diprioritaskan dalam pengembangan saat ini. Fitur ini mungkin dipertimbangkan untuk versi mendatang tetapi tidak termasuk dalam rencana saat ini.

Model MoSCoW menyediakan kerangka kerja yang jelas untuk memprioritaskan kebutuhan dalam pengembangan perangkat lunak. Dengan mengklasifikasikan kebutuhan ke dalam kategori ini, tim pengembang dapat fokus pada fitur yang paling penting dan mendesak, sehingga meningkatkan efisiensi dan efektivitas pengembangan sistem.

Menurut buku Mike Cohn *“User Stories Applied: For Agile Software Development”* metode MoSCoW digunakan sebagai teknik untuk memprioritaskan fitur dalam pengembangan perangkat lunak. Cohn menjelaskan bahwa teknik MoSCoW memungkinkan tim pengembang dan user untuk lebih mudah berkomunikasi dan mencapai kesepakatan tentang apa yang harus dimasukkan dalam sistem. Ini membantu mengelola prioritas serta memungkinkan perencanaan sistem yang lebih efisien (Cohn, 2004).

Penerapan metode MoSCoW dalam penelitian ini akan dilakukan sesuai dengan tahapan kebutuhan perangkat lunak menurut Ian Sommerville, yang meliputi:

**a. Elicitasi Kebutuhan**

Pengumpulan data dari pemangku kepentingan STIKES PEMKAB Jombang adalah bagian dari tahapan ini. Untuk mengetahui kebutuhan dan keinginan user metode seperti wawancara dan observasi dokumen akan digunakan. Tahap elicitasi ini akan menghasilkan kumpulan kebutuhan mentah yang belum diprioritaskan.

## b. Analisis Kebutuhan

Pada langkah ini semua kebutuhan yang telah dikumpulkan akan dianalisis dan dikategorikan menggunakan model MoSCoW. Metode ini akan membagi kebutuhan menjadi empat kategori prioritas, yaitu:

- ***Must Have***, fitur yang sangat penting dan harus ada agar sistem dapat beroperasi, seperti daftar mata kuliah.
- ***Should Have***, fitur yang diinginkan dan penting tetapi tidak kritis.
- ***Could Have***, fitur tambahan yang dapat ditambahkan jika ada cukup waktu dan sumber daya.
- ***Won't Have***, fitur yang tidak akan digunakan pada pengembangan saat ini tetapi mungkin akan dipertimbangkan untuk pengembangan masa depan.

Di sini, metode MoSCoW membantu menetapkan prioritas kebutuhan, sehingga developer dapat berkonsentrasi pada fitur yang lebih penting.

**c. Spesifikasi / Dokumentasi**

Kebutuhan Setelah prioritas kebutuhan ditetapkan, semua kebutuhan didokumentasikan secara sistematis dalam Dokumen Persyaratan Kebutuhan Software (SRS). Dokumen ini berfungsi sebagai referensi utama bagi tim developer dan memastikan bahwa semua pemangku kepentingan telah menyetujuinya.

**d. Validasi Kebutuhan**

Untuk mencapai tujuan ini, perlu dilakukan validasi kebutuhan dengan pemangku kepentingan untuk memastikan bahwa kebutuhan yang telah selesai dikategorikan sesuai dengan harapan mereka.

Metode MoSCoW digunakan dalam setiap tahapan ini untuk memastikan bahwa kebutuhan perangkat lunak yang paling penting dan mendesak dilaksanakan terlebih dahulu, dan jika waktu dan sumber daya memungkinkan, kebutuhan tambahan dapat disesuaikan.