

BAB II TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Pengertian Rancang Bangun

Rancang bangun (*design*) adalah proses teknis yang mengubah persyaratan bisnis menjadi representasi perangkat lunak yang dapat diimplementasikan (Kendall & Kendall, 2018). Dengan kata lain, rancang bangun merupakan proses penciptaan dan pembuatan suatu aplikasi ataupun sistem yang belum ada atau memperbaiki sistem yang lama.

2.1.2 Pengertian Aplikasi

Aplikasi adalah jenis program komputer yang dirancang untuk melakukan tugas-tugas tertentu. Aplikasi bisa berupa perangkat lunak yang berjalan di berbagai media seperti komputer dan ponsel cerdas. Fungsinya dapat mencakup produktivitas, hiburan, dan kebutuhan bisnis (Stair & Reynolds, 2018). Selain itu, aplikasi juga bisa memfasilitasi interaksi pengguna dengan sistem yang kompleks seperti basis data atau jaringan komputer. Penggunaan aplikasi telah menjadi bagian penting dalam kehidupan sehari-hari, memungkinkan tugas-tugas dilakukan dengan lebih efisien. Proses pembuatan aplikasi melibatkan perancangan, pengkodean, pengujian, dan implementasi. Tujuan penggunaan aplikasi bervariasi, termasuk meningkatkan produktivitas dan pengalaman pengguna dengan teknologi. Aplikasi terus mengalami inovasi dan pembaruan untuk memenuhi kebutuhan yang berkembang. Dengan demikian, aplikasi memainkan peran krusial dalam mendukung aktivitas manusia di era digital.

2.1.3 Pengertian Pengajuan

Menurut Kamus Besar Bahasa Indonesia (KBBI), pengertian pengajuan adalah tindakan atau proses mengajukan sesuatu kepada pihak yang berwenang untuk dipertimbangkan atau diputuskan. Dalam konteks ini, pengajuan bisa merujuk pada berbagai hal seperti permohonan, proposal, atau usulan yang disampaikan kepada pihak yang berwenang, baik

itu dalam lingkup formal maupun informal. Proses pengajuan seringkali melibatkan tahapan tertentu sesuai dengan kebijakan atau prosedur yang berlaku untuk memastikan bahwa permintaan atau proposal tersebut diterima dan diproses dengan baik.

2.1.4 *Systems Development Life Cycle (SDLC)*

Software Development Life Cycle (SDLC) adalah sebuah pola yang terdiri dari serangkaian rencana untuk mengembangkan sistem perangkat lunak. SDLC memberikan panduan melalui tugas-tugas untuk menganalisis, merancang, mengimplementasikan, dan memelihara sistem informasi (Pressman, 2014). SDLC adalah kerangka kerja yang membantu pengembang untuk membangun dan memelihara sistem informasi secara sistematis dan efektif. Berbagai model SDLC memiliki kesamaan dalam tahapan-tahapan yang dilalui, namun terdapat perbedaan dalam penekanan dan detailnya. Tahap-tahap tersebut meliputi:

A. Perencanaan (*Planning*)

Tahapan awal kebutuhan pengguna diidentifikasi, tujuan proyek ditetapkan, dan rencana pengembangan disusun (Pressman, 2014).

B. Analisis (*Analysis*)

Proses pemahaman lebih mendalam terhadap kebutuhan pengguna dan persyaratan fungsional serta non-fungsional dari perangkat lunak yang akan dikembangkan (Sommerville, 2016).

C. Desain (*Design*)

Merancang struktur dan arsitektur perangkat lunak berdasarkan hasil analisis, termasuk antarmuka pengguna, struktur *database*, dan alur kerja sistem (Pressman, 2014).

D. Implementasi (*Implementation*)

Tahap kode perangkat lunak sebenarnya dibuat berdasarkan desain yang telah dirancang sebelumnya (Sommerville, 2016).

2.1.5 Object Oriented Analysis and Design (OOAD)





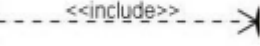
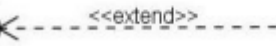
Object Oriented Analysis and Design (OOAD) adalah pendekatan analisis dan desain sistem yang menggunakan konsep pemrograman berorientasi objek untuk mengembangkan perangkat lunak (Kendall & Kendall, 2019). OOAD merupakan teknik pengembangan perangkat lunak yang berfokus pada identifikasi, klasifikasi, dan pemodelan objek-objek dalam sistem. Pertama, OOAD mengidentifikasi objek-objek relevan dalam sistem, seperti pengguna, produk, dan transaksi. Kemudian, objek-objek ini dikelompokkan ke dalam kelas-kelas berdasarkan kesamaan sifatnya. Lalu, OOAD memodelkan objek-objek tersebut menggunakan diagram untuk menggambarkan atribut, hubungan, dan perilakunya.

2.1.6 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk menggambarkan dan membangun perangkat lunak berorientasi objek (Kendall & Kendall, 2019). UML menjadi metodologi yang paling banyak digunakan saat ini untuk analisis dan perancangan sistem berorientasi objek, seiring dengan penggunaan bahasa pemrograman berorientasi objek. Meskipun demikian, model-model dalam UML dapat dikategorikan berdasarkan sifatnya, yaitu statis atau dinamis. Berikut beberapa jenis diagram UML :

2.1.6.1 Use Case Diagram

Menggambarkan interaksi antara sistem dan penggunanya. Diagram ini menunjukkan fungsionalitas sistem dari sudut pandang pengguna, membantu memahami berbagai aksi atau skenario yang dapat dilakukan oleh pengguna (aktor) dalam sistem, dan bagaimana sistem berinteraksi dengan mereka. *Use Case Diagram* umumnya terdiri dari aktor, use case (kasus penggunaan), dan hubungan antara keduanya.





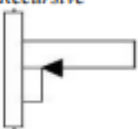


Simbol	Deskripsi
<i>Use case</i> 	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja
<i>Aktor / actor</i> 	<i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i>
<i>Asosiasi / association</i> 	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data
<i>Asosiasi / association</i> 	Asosiasi antara aktor dengan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
<i>Include</i> 	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> contohnya adalah pemanggilan sebuah fungsi program
<i>Extend</i> 	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi

Gambar 2. 1 Notasi *Use Case Diagram*

Sumber (Dennis, Wixom, dan Tegarden, 2015)

2.1.6.2 *Sequence Diagram*



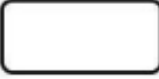



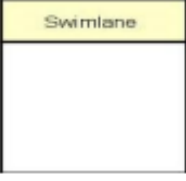
Berfokus pada interaksi antar objek dalam sistem secara berurutan. Diagram ini menunjukkan bagaimana objek-objek berkomunikasi satu sama lain dan saling bertukar pesan atau panggilan fungsi. *Sequence Diagram* membantu memahami proses aliran kontrol atau proses yang bersifat berurutan dalam sistem, serta proses dari awal hingga akhir dan hubungan antar objek dalam sistem.

Simbol	Deskripsi
 : Entity Class	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
 Boundary Class	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan form cetak
 Control Class	<i>Control Class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
	<i>Message</i> , simbol mengirim pesan antar kelas
 <i>Recursive</i>	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
 <i>Activation</i>	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi
 <i>Lifeline</i>	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek sepanjang <i>lifeline</i> terdapat <i>activation</i>

Gambar 2. 2 Notasi Sequence Diagram
Sumber (Dennis, Wixom, dan Tegarden, 2015)

2.1.6.3 Activity Diagram

Activity Diagram berfungsi sebagai diagram alur untuk menggambarkan serangkaian kegiatan atau tindakan yang terjadi dalam suatu proses atau aktivitas. Diagram ini menunjukkan aliran kerja sistem atau prosedur tertentu dari satu kegiatan ke kegiatan berikutnya, menggunakan simbol-simbol seperti aktivitas, keputusan, fork, join, dan garis aliran untuk menggambarkan alur kerja. *Activity Diagram* membantu memvisualisasikan proses bisnis atau prosedur yang kompleks dan memahami urutan langkah-langkah yang diperlukan dalam suatu aktivitas atau proses.

Simbol	Deskripsi
<i>Start Point</i> 	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
<i>End Point</i> 	<i>End Point</i> , akhir aktivitas
<i>Activities</i> 	<i>Activities</i> , menggambarkan suatu proses atau kegiatan bisnis
<i>Fork</i> atau Percabangan 	<i>Fork</i> atau percabangan, digunakan untuk menunjukan kegiatan yang dilakukan secara paralel untuk menggabungkan dua kegiatan paralel menjadi satu
<i>Join</i> atau Penggabungan 	<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukan adanya dekomposisi
<i>Decision Points</i> 	<i>Decision points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>
<i>Swimlane</i> 	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukan siapa melakukan apa

Gambar 2. 3 Notasi *Activity Diagram*
 Sumber (Dennis, Wixom, dan Tegarden, 2015)

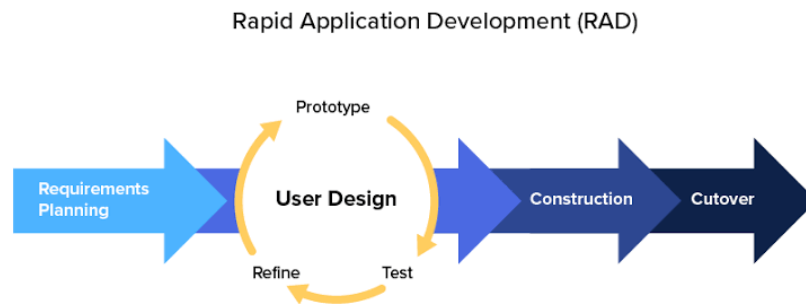
Dapat disimpulkan bahwa *Unified Modeling Language* merupakan bahasa visual yang digunakan untuk mendefinisikan, memvisualisasikan, dan mendokumentasikan sistem perangkat lunak. UML telah menjadi standar bahasa yang banyak digunakan di dunia industri untuk membantu para developer dalam mengembangkan perangkat lunak yang kompleks.

2.1.7 *Rapid Application Development (RAD)*

Rapid Application Development (RAD) adalah metodologi pengembangan sistem informasi yang menekankan kecepatan, fleksibilitas,

dan partisipasi aktif pengguna. RAD menggunakan prototipe untuk membantu pengguna memvisualisasikan dan memvalidasi kebutuhan mereka, serta untuk mempercepat proses pengembangan dan meningkatkan kualitas sistem.

Proses pengembangan dengan metode RAD melibatkan empat fase utama yaitu perencanaan kebutuhan, desain pengguna, konstruksi cepat dan cutover (Zen & Iswavigra, 2023).



Gambar 2. 4 Rapid Application Development (RAD)

Sumber (Zen & Iswavigra, 2023)

A. Perencanaan Kebutuhan (*Requirements Planning*)

Pada tahap ini, fokusnya adalah mempelajari dan merencanakan kebutuhan pengguna dengan cermat. Tim pengembang akan mengidentifikasi persyaratan fungsional dan non-fungsional untuk perangkat lunak yang akan dikembangkan.

B. Desain Pengguna (*User Design*)

Tahap ini berfokus pada interaksi langsung dengan pengguna. Tim pengembang bekerja sama dengan pengguna untuk merancang antarmuka pengguna dan memperjelas persyaratan fungsional yang telah ditetapkan sebelumnya.

C. Kontruksi Cepat (*Rapid Construction*)

Pada tahap ini, pengembangan perangkat lunak dilakukan dengan cepat. Biasanya, metode pengembangan RAD menggunakan

alat-alat atau teknik yang memungkinkan iterasi dan pengembangan cepat, seperti pengkodean berbasis prototipe.

D. Implementasi (*Cutover*)

Tahap ini merupakan tahap implementasi di mana perangkat lunak yang telah dikembangkan diperkenalkan secara penuh ke lingkungan produksi. Ini melibatkan peluncuran resmi produk, pelatihan pengguna, dan transisi dari sistem sebelumnya jika ada.

Model RAD menawarkan beberapa keuntungan dibandingkan dengan model pengembangan tradisional, seperti waktu pengembangan yang lebih singkat, kualitas yang lebih tinggi, dan fleksibilitas yang lebih besar. Secara keseluruhan, RAD adalah model pengembangan perangkat lunak yang menarik ketika kecepatan, fleksibilitas, dan partisipasi pengguna menjadi prioritas.

2.1.8 *Database*

Database adalah koleksi data yang diorganisir dan disimpan dalam pengelolaan informasi. Sistem *database* adalah sistem perangkat lunak yang memungkinkan pengguna untuk membuat dan memelihara *database* (Elmasri & Navathe, 2017).

Dalam definisi ini, "data" merujuk pada informasi yang tersimpan dalam berbagai format, seperti teks, angka, gambar, dan lainnya. Data-data ini kemudian diatur dan terhubung secara logis dalam *database*, yang merupakan wadah untuk menyimpan informasi terstruktur. *Databases Management Systems* (DBMS) merupakan perangkat lunak yang bertanggung jawab atas manajemen, penyimpanan, dan pengelolaan akses terhadap data dalam *database*. DBMS menyediakan antarmuka untuk pengguna dan aplikasi untuk berinteraksi dengan *database*, termasuk operasi seperti penyimpanan, pengambilan, pembaruan, dan penghapusan data. Dengan adanya DBMS, pengguna dapat mengakses dan memanipulasi data dengan efisien, serta memastikan keamanan, konsistensi, dan integritas data dalam *database*. Dengan demikian, definisi *database* menurut Elmasri

dan Navathe menekankan pentingnya penyimpanan data yang terstruktur dan akses yang terkelola dengan baik melalui perangkat lunak DBMS.

2.2 Tinjauan Studi

Tesis berjudul "**Pengembangan Sistem Informasi Biro Kemahasiswaan Dan Manajemen Organisasi Kemahasiswaan Di Universitas Widyatama**" yang ditulis oleh Alvi Agrian Munazar pada tahun 2019, membahas tentang pengembangan sistem informasi yang dirancang untuk beasiswa dan kegiatan organisasi kemahasiswaan di lingkungan Universitas Widyatama. Latar belakang dikembangkannya sistem ini adalah biro kemahasiswaan yang mengalami kesulitan dalam penyebaran informasi terkait beasiswa maupun kegiatan kemahasiswaan. Proses pengajuan beasiswa dan proposal kegiatan kemahasiswaan dilakukan dengan cara bertemu langsung dengan biro kemahasiswaan sehingga memakan banyak waktu dan tenaga. Oleh karena itu, diperlukan sistem informasi yang dapat mengatasi permasalahan dalam kemahasiswaan. Pengembangan sistem informasi ini menggunakan pendekatan pengembangan sistem dengan tahapan analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Pendekatan tersebut dirasa mampu memberikan solusi yang sistematis dan efektif dalam menangani permasalahan yang ada pada biro kemahasiswaan dan manajemen organisasi kemahasiswaan. Sistem informasi yang dikembangkan dalam penelitian ini mencakup berbagai fitur mengelola informasi beasiswa dan juga proposal pengajuan kegiatan kemahasiswaan. Sistem ini dirancang untuk dapat diakses oleh berbagai pihak, termasuk mahasiswa, staf biro kemahasiswaan, dan pengurus organisasi kemahasiswaan. Dengan adanya sistem ini, proses pengelolaan data dan kegiatan kemahasiswaan menjadi lebih terstruktur dan efisien.

Jurnal berjudul "**Perancangan Sistem Informasi Manajemen Kegiatan Mahasiswa Berbasis Web**" oleh Ana Juita Oktasari dan Denny Kurniadi, diterbitkan pada jurnal VOTEKNIKA Vol. 7, No. 4, Desember 2019, membahas terkait perancangan sistem informasi manajemen kegiatan mahasiswa. Latar belakang dikembangkannya sistem ini berasal dari alur pengajuan proposal pada Universitas Negeri Padang. Sebelumnya Unit

Kegiatan Mahasiswa harus menemui beberapa pihak untuk pengecekan dan persetujuan proposal kegiatan. Hal tersebut membutuhkan waktu tenaga dan biaya dalam proses mengadakan suatu kegiatan. Oleh karena itu, diperlukan sistem informasi untuk mempercepat dan mempermudah proses mengajukan proposal dan pelaporan pertanggung jawaban kegiatan mahasiswa. Perancangan sistem informasi ini menggunakan metode pengembangan sistem dengan pendekatan *Waterfall*, yang melibatkan tahapan-tahapan seperti analisis kebutuhan, desain sistem, implementasi, pengujian, dan pemeliharaan. Tahapan-tahapan ini dipilih karena memberikan alur yang sistematis dan terstruktur dalam pengembangan perangkat lunak, terutama dalam konteks sistem informasi. Sistem informasi yang dikembangkan mampu mengelola berbagai jenis kegiatan mahasiswa, mulai dari pendaftaran kegiatan, pemantauan pelaksanaan, hingga pelaporan kegiatan. Sistem ini dirancang agar dapat diakses oleh berbagai pihak terkait, termasuk mahasiswa, dosen, dan panitia kegiatan. Salah satu fitur penting yang diimplementasikan dalam sistem ini adalah manajemen jadwal kegiatan, yang memungkinkan pengguna untuk melihat dan mengelola jadwal kegiatan secara *real-time*.

Jurnal berjudul **“Rancang Bangun Sistem Informasi Manajemen Organisasi Kemahasiswaan (Studi Kasus: Universitas Madura)”** oleh Nilam Ramadhani, Abd. Wahab Syahroni, dan Rian Wahyudi, diterbitkan dalam jurnal ANTIVIRUS Vol. 16, No. 2, November 2022, membahas terkait perancangan sistem informasi manajemen organisasi kemahasiswaan Universitas Madura. Latar belakang perancangan sistem ini adalah adanya kesalahan informasi dalam proposal kegiatan maupun laporan pertanggung jawaban kegiatan. Sebelumnya, pembuatan proposal masih dalam bentuk *hardcopy* dan belum terekam secara digital, sehingga akses informasi menjadi terbatas. Oleh karena itu, diperlukan pembuatan sistem informasi yang dapat mengelola administrasi, data kepengurusan dan keanggotaan, serta kegiatan kemahasiswaan. Penelitian ini menggunakan metode pengembangan sistem berbasis web dengan pendekatan *waterfall*. Tahapan-tahapan yang dilakukan meliputi analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Data yang dibutuhkan diperoleh melalui wawancara, observasi, dan

dokumentasi terkait dengan kebutuhan organisasi kemahasiswaan. Pengujian sistem menggunakan 2 jenis metode uji, yaitu *blackbox testing* dan *User Acceptance Test (UAT)* untuk memastikan bahwa sistem bekerja sesuai dengan kebutuhan pengguna dan spesifikasi yang telah dirancang. Hasil dari penelitian ini adalah sebuah Sistem Informasi Manajemen Organisasi Kemahasiswaan yang berbasis web. Sistem ini memungkinkan pengelolaan data keanggotaan, pengajuan kegiatan, laporan keuangan, dan komunikasi internal antaranggota organisasi menjadi lebih terstruktur dan mudah diakses. Selain itu, sistem ini juga dilengkapi dengan fitur untuk monitoring dan pelaporan kegiatan organisasi.

Jurnal berjudul **“Rancang Bangun Sistem Informasi Manajemen Proposal Kemahasiswaan”** oleh Fajar Nugraha, Muhammad Arifin, dan Arif Harjanto, diterbitkan dalam jurnal SIMBADA Vol. 3, No. 1, Maret 2020 membahas terkait perancangan sistem informasi manajemen proposal kemahasiswaan. Latar belakang pengembangan sistem ini adalah masalah pada proses pengajuan proposal dan administratif dokumen. Sebelumnya, proses pengajuan proposal memerlukan waktu yang lama dan masalah administratif dokumen, seperti format proposal yang tidak sesuai, prosedur penyerahan usulan proposal kegiatan yang tidak terstruktur serta tidak terdokumentasinya pelaporan kegiatan. Oleh karena itu, diperlukannya sebuah sistem informasi manajemen yang dapat digunakan untuk pengelolaan usulan proposal kemahasiswaan. Penelitian ini menggunakan metode pengembangan sistem berbasis Web dengan pendekatan *Software Development Life Cycle (SDLC)* model *waterfall*. Tahapan-tahapan yang dilalui meliputi analisis kebutuhan sistem, desain sistem, implementasi, serta pengujian. Penulis juga melakukan wawancara dan studi literatur untuk memahami kebutuhan pengguna dan spesifikasi sistem yang optimal. Hasil dari penelitian ini adalah terbentuknya sebuah sistem informasi manajemen proposal kemahasiswaan yang berbasis web. Sistem ini memfasilitasi pengelolaan proposal mulai dari pengajuan oleh mahasiswa, penilaian oleh dosen atau pihak terkait, hingga *monitoring* dan penyimpanan data proposal.

Jurnal berjudul **“Rancang Bangun Aplikasi Sistem Informasi Organisasi Mahasiswa (SIOMAH)”** oleh Swono Sibagariang, Afdhol Dzikri, Dodi Prima Resda, dan Jhon Hericson Purba dalam jurnal Mahajana Inforansi Vol. 6, No. 2, Desember 2022 membahas terkait perancangan aplikasi sistem informasi organisasi mahasiswa. Tujuan penelitian ini merancang dan mengembangkan Aplikasi Sistem Informasi Organisasi Mahasiswa (SIOMAH). Tujuan utama dari pengembangan aplikasi ini adalah untuk membantu organisasi mahasiswa dalam mengelola data keanggotaan, kegiatan, dan administrasi secara lebih efisien dan terstruktur. Aplikasi ini dirancang untuk mempermudah proses dokumentasi dan komunikasi dalam organisasi mahasiswa. Proses pengembangan dimulai dari analisis kebutuhan, yang melibatkan identifikasi kebutuhan organisasi mahasiswa melalui wawancara dan observasi. Selanjutnya, dilakukan perancangan sistem dengan membuat desain arsitektur, antarmuka, dan basis data. Hasil penelitian menunjukkan bahwa aplikasi ini mampu menjadi solusi digital yang relevan bagi organisasi mahasiswa, dengan rekomendasi pengembangan lebih lanjut, seperti penambahan fitur komunikasi antaranggota, integrasi dengan media sosial, dan penerapan teknologi cloud untuk meningkatkan skalabilitas dan kinerja sistem.