**IMPLEMENTASI APLIKASI SISTEM AKADEMIK**

**BERBASIS MOBILE DENGAN**

**FRAMEWORK IONIC**

**(Studi Kasus: SMK Darussalam Karangpucung)**

# HALAMAN JUDUL

**Skripsi**



Disusun oleh

**Afif Waliyudin**

**21SA1255**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS AMIKOM PURWOKERTO**

**PURWOKERTO**

**2024**

**BAB II  
TINJAUAN PUSTAKA**

## **Landasan Teori**

1. Sistem Akademik

Sistem akademik adalah sistem yang dirancang untuk memenuhi kebutuhan akademik yang menginginkan layanan pendidikan yang terkomputerisasi untuk meningkatkan kinerja, kualitas pelayanan, daya saing dan kualitas sumber daya manusia yang dihasilkannya. (PERANCANGAN SISTEM INFORMASI AKADEMIK BERBASIS MOBILE PADA SEKOLAH DASAR NEGERI SUKATANI 3, t.t.-a) Sistem Akademik bisa diartikan sebuah sistem informasi yang di bangun atas komponen yang terdiri dari data-data siswa, data-data guru, dan lain-lain yang saling berhubungan antara satu dengan yang lainnya untuk membentuk suatu sistem. (Marlina Ariansyah & Wijaya Sekolah Tinggi Manajemen Informatika dan Komputer Prabumulih, 2021).

1. Aplikasi Mobile

Aplikasi *mobile* merupakan perangkat berbasis mobile seperti handphone dan tablet yang dapat diakses melalui internet serta diunduh oleh pengguna. (Ziha Fidela dkk., t.t.).

1. *Ionic Framework*

*Ionic Framework* digunakan sebagai alat kerja utama yang digunakan untuk mengembangkan aplikasi sistem akademik. Ionic adalah sekumpulan teknologi yang dikembangkan untuk membangun aplikasi

mobile hybrid yang powerful, cepat, mudah dan juga memiliki tampilan yang menarik. (Suhaidi dkk., t.t.) . Ionic mengunakan Cordova untuk dapat bekerja sebagai native app. *Ionic framework* di bangun untuk menghasilkan performa terbaik pada perangkat mobile. *Ionic framework* menggunakan HTML, CSS, dan Javascript, dalam rangka menciptakan framework yang paling cocok untuk pengembangan aplikasi mobile. (*jm\_elektro,+28986-60902-1-ED+(3)ok*, t.t.)

1. Metode *Rapid Application Development (RAD)*

*Rapid Application Development* (RAD) adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik *incremental* atau bertingkat (Pengembangan and Hartono M.Kom n.d.) Nurman mengatakan, *Rapid Application Development* (RAD) merupakan salah satu model dalam kerangka kerja *System Development Life Cycle* (SDLC) yang mengadopsi proses linear sekuensial dengan menekankan siklus pengembangan yang sangat singkat. RAD seringkali digunakan karena dapat menjadi acuan dalam pengembangan system informasi karena unggul dalam hal kecepatan, ketepatan dan biaya yang relatif rendah. (Hidayat & Hati, t.t.) Metode RAD ini mampu mempersingkat proses dari tahapan pengembangan sistem informasi yang mana melalui proses perencaaan, desain dan implementasi. (Novrian dkk., 2022)



Gambar 2.1. Tahapan Metode RAD

Berdasarkan gambar tahapam RAD dapat dipaparkan beberapa langkah – langkah penerapan RAD untuk pengembangan sistem.

* + - 1. *Analysis & Quick Design*

Pada tahap awal dilakukan perencanaan kebutuhan dari pengembangan aplikasi dengan cara melakukan indentifikasi permasalahan dan pengumpulan data dari perancang untuk menetapkan tujuan akhir yang dibutuhkan oleh sistem. Dimana pada tahap ini Peneliti mengidentifikasi secara rinci fitur-fitur yang dibutuhkan dalam penjualan kayu. Serta memastikan bahwa semua kebutuhan bisnis dan teknis telah terindentifikasi secara jelas sebelum memulai pengembangan, sehingga dapat mengurangi resiko perubahan untuk tahap selanjutnya.

* + - 1. *Prototype Cycles*

Pada tahap ini, perancang melakukan perancangan sistem atau *prototype* dan melakukan pengujian. Jika rencana awal belum sesuai dengan kebutuhan, dapat disempurnakan atau diperbaiki. Sehingga pada tahap ini menghasilkan seluruh fitur yang akan dibuat. Sehingga dapat memungkinkan pengembangan untuk tervisualisasi, dengan mendapatkan unpan balik awal serta melakukan perbaikan sebelum memasuki tahap pengembangan selanjutnya.

* + - 1. *Deployment*

Pada proses *deployment*, desain sistem direalisasikan dan diaplikasikan serta memastikan bahwa sistem telah dibangun dengan kebutuhan. Pada tahap ini menghasilkan pengujian yang mencakup hasil dari berbagai tes, termasuk fungsionalitas, keamanan dan kinerja aplikasi. Sehingga diharapkan setelah keluar dari metode testing ini, aplikasi akan berjalan baik, aman dan memenuhi kebutuhan pengguna.

* + - 1. Implementation

Tahap implementasi melibatkan implementasi program, apakah metode program sesuai dengan kebutuhan sistem. Tahap ini juga merupakan tahap final Dimana sistem diterapkan dan dapat dijalankan. Hasil pada tahap deployment ini berupa sebuah aplikasi dan panduan penggunaan aplikasi. (P dkk., 2022).

1. MySQL

MySQL adalah satu bahasa pemrograman tingkat tinggi yang digunakan dalam perangkat lunak. MySQL adalah perangkat lunak sistem manajemen basis data SQL multithreaded dan multi-pengguna. (Hamni dkk., 2022) MySQL memiliki keunggulan dalam hal skalabilitas, kehandalan, kemampuan untuk mengintegrasikan data dari berbagai sumber, dan fitur-fitur keamanan yang mumpuni. (*[2] d1f851b277f26700f15b3d9782d71267*, t.t.)

1. ERD (Entity Relationship Diagram)

Entity Relationship Diagram (ERD) adalah diagram berbentuk notasi grafis yang berada dalam pembuatan database yang menghubungkan antara data satu dengan yang lain. Fungsi ERD adalah sebagai alat bantu dalam pembuatan database dan memberikan gambaran bagaimana kerja database yang akan dibuat. (Afiifah dkk., t.t.)

ERD memiliki tiga konsep utama yaitu :

1. Entitas

Sebuah entitas dapat berupa orang, tempat, objek, atau kejadian yang dapat dianggap penting bagi sebuah organisasi atau perusahaan. Setiap entitas memiliki beberapa atribut yang mendeskripsikan karakteristik dari objek. Atribut yang ada dalam entitas harus disimpan dan dicatat dalam basis data. Entitas pada komponen ERD dapat dibedakan menjadi dua macam yaitu strong entitydan weak entity. Strong entity merupakan entitas yang tidak bergantung pada entitas lain atau entitas yang dapat berdiri sendiri. Sedangkan untuk *weak entity* merupakan entitas yang keberadaannya tergantung pada entitas lain.

1. Atribut

Setiap entitas memiliki karakteristik tertentu yang disebut dengan atribut. Atribut berfungsi untuk mendeskripsikan karakteristik yang ada pada entitas yang disimpan dalam basis data. Berdasarkan karakteristik sifatnya, atribut dapat dibedakan menjadi beberapa jenis yaitu simple *attribute* dan *composite* *attribute*, *single* *valued attribute* dan *multi value attribute*, *derived attribute*, *key attribute*. *Primary key* adalah nama untuk atribut yang digunakan dalam mengenali suatu entitas. Atribut dalam entitas yang merupakan *primary key* adalah kode identifikasi yang bersifat unik ditunjukkan berdasarkan masing-masing *record* pada sistem. *Primary key* bertujuan untuk memberitahu lokasi untuk tiap catatan pada suatu file tentang catatan-catatan yang sama

1. Relasi

Relasi adalah sebuah hubungan antara dua atau lebih entitas yang saling berkaitan. Relasi pada ERD dapat digambarkan dengan menggunakan simbol belah ketupat (*diamond*). Relasi memiliki beberapa jenis relasi yaitu *unary*, *binary*, *ternary*. Pemodelan ERD menggunakan notasi entitas berbentuk persegi, relasi berbentuk belah ketupat dan atribut berbentuk oval. (Togatorop dkk., 2021)

Berikut simbol ERD dan penjelasannya.

Table 2.1 Simbol ERD dan fungsinya

| **Simbol** | **Nama** | **Fungsi** |
| --- | --- | --- |
|  | *Entity* | Kumpulan dari Objek yang dapat diidentifikasi secara unik |
|  | *Weak Entity* | Entity yang keberadaannya sangat bergantung pada keberadaan entitas yang lainnya. |
|  | *Relationship* | Hubungan yang terjadi antara satu atau lebih entity. |
|  | *Identifying Relationship* | Suatu relationship dimana keberadaann anak entity bergantung pada induknya |
|  | Attribut | Karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas |

1. UML (*Unified Modelling Language*)

*Unified Modeling Language* (UML) adalah sebuah bahasa pemodelan perangkat lunak yang telah distandardisasi sebagai media penulisan cetak biru (*blueprints*) perangkat lunak (Pressman). UML bisa saja digunakan untuk visualisasi, spesifikasi, kontruksi dan dokumentasi beberapa bagian-bagian dari system yang ada dalam perangkat lunak. jelaskan pada bagian-bagian selanjutnya). (Sumiati dkk., t.t.)

UML dibagi menjadi beberapa jenis diantaranya :

1. *Use Case Diagram*

Use Case Diagram adalah komponen yang memberikan gambaran secara fungsional dalam suatu sistem, sehingga konsumen dan pembuat sistem dapat memahami alur sistem yang akan dibuat. Use Case Diagram sendiri merupakan proses penggambaran yang menunjukan hubungan alur sistem yang akan diterapkan antara pengguna dan sistem yang akan dirancang.

Tabel 2. 1 Simbol *Use Case Diagram*

|  |  |
| --- | --- |
| **Simbol** | **Keterangan** |
|  | Aktor : Mewakili peran orang , sistem yang lain, atau alat Ketika berkomunikasi dengan *use case*. |
|  | *Use case* : Abstraksi dan interaksi antara sistem dan aktor. |
|  | *Association* : Abstraksi dari penghubung antara aktor dengan *use case*. |
|  | *Generelisasi* : menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan *use case*. |
| <<include>> | Menunjukan bahwa suata *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya. |
| <<extend>> | Menunjukan bahwa suata *use case* seluruhnya merupakan tambahan fungsional dari *use case* lainya jika suatu kondisi terpenuhi. |

1. *Activity Diagram*

Activity diagram adalah diagram yang menjelaskan tentang alur aktivitas dalam program yang sedang dirancang, termasuk bagaimana alur dimulai, keputusan yang mungkin terjadi, dan bagaimana alur tersebut akan berakhir. Activity diagram juga dapat menggambarkan sistem paralel yang mungkin terjadi selama beberapa eksekusi.

Tabel 2. 2 Simbol *Activity Diagram*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Status awal | Sebuah diagram aktivitas memiliki sebuah status awal. |
|  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
|  | Percabangan / *decision* | Percabangan dimana ada pilihan aktivitas yang lebih dari satu. |
|  | Penggabungan / *join* | Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan menjadi satu. |
|  | Status akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah akhir |
|  | *Swimlane* | *Swimlane* memisahkan oraganisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

Activity diagram digunakan untuk mengilustrasikan berbagai tindakan yang dapat dieksekusi oleh pengguna dalam aplikasi, dengan menguraikan setiap langkah yang terkait dengan setiap menu.

1. *Sequence Diagram*

*Sequence diagram* adalah diagram yang menggambarkan interaksi antara objek dan menunjukkan komunikasi di antara objek - objek tersebut. *Sequence diagram* digunakan untuk menjelaskan perilaku dalam sebuah skenario dan menunjukkan bagaimana entitas dan sistem berinteraksi, termasuk pesan-pesan yang digunakan selama interaksi tersebut. Semua pesan digambarkan dalam urutan eksekusi.

Tabel 2. 3 Simbol Sequence diagram

|  |  |  |
| --- | --- | --- |
| **Gambar** | **Nama** | **Keterangan** |
|  | *Entity Class* | Gambaran sistem sebagai landasan dalam menyusun basis data. |
|  | *Boundry Class* | Menangani komunikasi antar lingkungan sistem. |
|  | *Control Class* | Bertanggung jawab terhadap kelas-kelas , terhadap objek yang berisi logika. |
|  | *Recursive* | Pesan untuk dirinya. |
|  | *Activation* | Mewakili proses durasi aktivasi sebuah operasi |
|  | *Life line* | Komponen yang digambarkan garis putus terhubung dengan objek |

Sequence diagram bertujuan untuk menunjukan memvisualisasikan alur interaksi antara berbagai objek dalam sistem selama eksekusi suatu proses atau fungsi tertentu.

1. *Class Diagram*

Class diagram adalah salah satu pemodelan penting dalam UML, yang berfungsi untuk membuat model logis dari sebuah sistem. Dalam sebuah laporan sistem, class diagram harus disertakan. Sequence diagram, sebagai bagian dari UML (*Unified Modeling Language*), juga wajib ada dalam laporan karena perannya yang sangat penting.

Tabel 2. 4 Simbol Class Diagram

| **Gambar** | **Nama** | **Deskripsi** |
| --- | --- | --- |
| +Atribut  Nama Kelas  +Metode | *Class* | Menggambarkan sebuah kelas pada sistem yang  terbagi menjadi 3 bagian. Bagian atas adalah nama  kelas. Bagian tengah adalah atribut kelas. Bagian  bawah adalah methode dari kelas. |
|  | *Association* | Hubungan statis antar kelas. menggambarkan kelas  yang memiliki atribut berupa kelas lain atau kelas  yang harus mengetahui eksistensi kelas lain. |
|  | *Agregation* | Hubungan yang menyatakan bahwa suatu kelas menjadi atribut bagi kelas lain. |
|  | *Composition* | Bentuk khusus dari agregation dimana kelas yang  menjadi bagian diciptakan setelah kelas whole  dibuat. |
|  | *Generalization* | Relasi antar kelas dengan makna generalisasispesialisasi (umum-khusus). |
|  | *Directed Assocoation* | Asosiasi dengan makna kelas yang satu digunakan  oleh kelas yang lain. |

*Class diagram* digunakan untuk memodelkan struktur statis sistem dengan menunjukkan kelas-kelas yang ada dalam sistem beserta atribut dan metode mereka. Ini memberikan gambaran yang jelas tentang elemen-elemen penyusun sistem.

1. *RESTful* API

*Restful* API merupakan arsitektur untuk penerapan *web service* dalam menerapkan konsep peralihan antar negara. (Hadinata & Stianingsih, 2024) Negara disini dapat dilustrasikan sebagai peramban yang meminta halaman web, pada sisi server akan mengirimkan keadaan halaman web saat ini ke peramban. Dengan *RESTful* API memungkinkan berbagai sistem untuk dapat berkomunikasi dan mengirim atau menerima data dengan cara yang cukup sederhana.

1. Figma

Figma adalah salah satu alat desain yang biasanya digunakan untuk membuat tampilan aplikasi *mobile, desktop, website* dan lain-lain. Figma bisa digunakan di sistem operasi windows, linux ataupun mac dengan terhubung ke internet. Umumnya Figma banyak digunakan oleh seseorang yang bekerja dibidang UI/UX, web design dan bidang lainnya yang sejenis. Dengan *tools* figma ini, pembuat *design* dapat dimudahkan dalam membuat desain tampilan pengguna dan juga figma memudahkan para pembuat design dalam merancang desain. (Ikhsandi dkk., t.t.)

1. Node JS

Node.JS adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi web. Aplikasi ini ditulis dalam campuran Bahasa C++ dan juga JavaScript, mempunyai model *event driven* (basis event) dan *asynchrounous I/O*. Tidak seperti kebanyakan Bahasa JavaScript yang dijalankan pada web browser, Node.JS dieksekusi sebagai aplikasi server. (Saiholau, 2024)

1. Express JS

Express.js adalah kerangka kerja web *backend* yang sangat populer dan ringan, dibangun di atas Node.js dan ditulis dalam JavaScript. Kerangka ini dirancang untuk memfasilitasi pengembangan aplikasi web dan API dengan memberikan alat serta struktur yang dibutuhkan. Salah satu keunggulan utama Express.js adalah kemudahan penggunaannya, yang didukung oleh sintaks yang intuitif dan fleksibel. Hal ini memungkinkan pengembang untuk membuat aplikasi web dengan cepat dan efisien, baik untuk proyek kecil maupun skala besar, tanpa mengorbankan kontrol atau performa. (Nahjan dkk., 2023)

## **Penelitian Sebelumnya**

Tabel 2.6. Penelitian Sebelumnya

| **No** | **Judul Penelitian** | **Peneliti** | **Hasil** | **Teknik/Basis/Metode/**  **Framework** | **Persamaan** | **Perbedaan** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Perancangan Sistem Informasi Akademik Berbasis Mobile Pada Sekolah Dasar Negeri Sukatani 3  (PERANCANGAN SISTEM INFORMASI AKADEMIK BERBASIS MOBILE PADA SEKOLAH DASAR NEGERI SUKATANI 3, t.t.-b) | A. Faoji | Berhasil mengembangkan sistem informasi  akademik pada SDN Sukatani 3  berbasis mobile yang dengan Teknik aplikasi pengembangan berbasis mobile Native Development sehingga data dapat terintegrasi  dengan database sehingga tidak terjadi  duplikasi data. | Sistem informasi dikembangkan dengan Teknik pengembangan aplikasi berbasis mobile. | Persamaan antara penelitian sebelumnya dan penelitian yang dilakukan yakni mengembangkan aplikasi sistem informasi berbasis mobile yang diterapkan pada perangkat seluler. | Penelitian sebelumnya mengembangkan aplikasi / sistem informasi berbasis mobile dengan Teknik Native Development dengan memanfaatkan Bahasa pemrograman java sedangkan penelitian yang dilakukan menggunakan Teknik Hybrid Development  untuk mengembangkan aplikasi mobile dengan teknologi web seperti HTML, CSS dan Javascript sehingga dapat lebih fleksibel dalam proses pengembangan. |
| 2 | Rancang Bangun Sistem Informasi Akademik SMK Negeri 1 Depok Berbasis Android dengan Pendekatan Rapid Application Development.  (A. Fauzi & Harli, 2019) | Ahmad Fauzi, Eko Harli | Sistem informasi dirancang dan dibangun dengan metode Rapid Application Development (RAD) untuk membuat sistem dengan cepat. Metode ini efektif untuk mengembangkan sistem informasi akademik berbasis Android dengan nama SisAkOne. | Menggunakan metode Pengembangan RAD (*Rapid Application Development*) sebagai metode yang cocok digunakan karena memerlukan pengembangan yang cepat, tapi tetap menyesuaikan kebutuhan pengguna. | Persamaan antara penelitian sebelumnya dengan yang sedang dikembangkan, sama sama menggunakan metode RAD (*Rapid Application Development*) | Penelitian sebelumnya menghasilkan sistem informasi akademik berbasis android yang melibatkan semua pihak area sekolah sedangkan penelitian ini menghasilkan aplikasi ejurnal untuk keperluan presensi kehadiran, pencatatan materi ajar serta optimalisasi jadwal mengajar yang melibatkan siswa dan guru. |
| 3 | Implementasi Algoritma Simulated Annealing Terhadap Rute  Perjalanan Pada Sistem Rekomendasi Objek Wisata | Rovianty Nugracia,  Dr. Kemas Muslim Lhaksmana, S.T., M.ISD. | Berdasarkan hasil pengujian dan analisis maka penulis dapat menyimpulkan bahwa Simulated Annealing  bekerja baik untuk pencarian rute berdasarkan kasus TSP. Penulis menggunakan algoritma Brute Force sebagai  algoritma pembanding dalam menganalisis performansi dari algoritma Simulated Annealing. Simulated  Annealing memiliki waktu eksekusi lebih sedikit jika dibandingkan dengan Brute Force dengan hasil yang  disarankan bersifat sub-optimal. Setelah beberapa kali iterasi apabila state atau rute yang dihitung selanjutnya  terus menghasilkan energy dalam jumlah besar maka iterasi dapat dihentikan sehingga dipilih state dengan  minimum energy untuk kemudian disarankan kepada user (wisatawan). Hasil yang disarankan kepada wisatawan  adalah rute yang baik, namun masih ada kemungkinan rute yang lebih baik akan ditemukan apabila iterasi  dilakukan hingga seluruh kemungkinan jarak dihitung. | Menggunakan algoritma simulated annealing untuk memecahkan masalah terkait optimalisasi rute perjalanan. | Penelitian sebelumnya dan penelitian yang dilakukan memiliki persamaan yakni menerapkan algoritma simulated annealing untuk mendapatkan solusi terbaik dari permasalahan optimasi. | Penelitian sebelumnya menerapkan simulated annealing yang berfokus untuk menentukan rute terbaik terhadap rekomendasi objek wisata, sedangkan penelitian yang dilakukan menerapkan simulated annealing demi mendapatkan penjadwalan mengajar yang optimal. |
| 4 | Implementasi Aplikasi Pinjaman Online Menggunakan Ionic  Framework Dengan Berbasiskan Android Pada  Bank Mandiri Taspen Cabang Nganjuk  (Renaldi1 & Putera2, T.T.) | Muhamad Renaldi, Andi Rahman Putera | Sistem baru ini akan menggantikan sistem lama dalam melayani peminjaman, yaitu  peminjaman secara online. Dimana sistem baru ini menggunakan aplikasi Pinjaman online  yang berbasis Android dimana nasabah nantinya bisa melakukan peminjaman secara  online, jadi nasbah yang ingin melakukan peminjaman cukup membuka aplikasi pinjaman  online, lakukan langkah-langkah yang terdapat dalam aplikasi tersebut, isi uang yang akan  diajukan oleh nasabah dan tenggang waktu setelah itu tunggu approval dari pihak Bank  Mandiri Taspen KCP Nganjuk, setelah mendapat approval, nasabah dipersilahkan  mengambil uang pinjaman di kantor Bank Mandiri Taspen KCP Nganjuk tanpa boleh  diwakilkan oleh orang lain. | Menggunakan Ionic Framework untuk pengembangan aplikasi berbasis android yang fleksibel. | Persamaan antara penelitian sebelumnya dengan yang sedang dikembangkan, sama sama menggunakan *ionic framework* sebagai kerangka kerja utama pengembangan aplikasi. | Penelitian sebelumnya menggunakan library Angular sebagai penerapan kode Ionic Framework, sedangkan penelitian yang dilakukan menggunakan library React JS untuk penerapan kode Ionic Framework. Perbedaan utama antara kedua penelitian tersebut adalah penggunaan library *front end* yang berbeda, yaitu Angular dan React JS, dalam mengembangkan aplikasi berbasis Ionic Framework. |