

### BAB III METODOLOGI PENELITIAN

Pada bab ini akan menjelaskan tentang tahapan penelitian dan metode yang akan digunakan. Secara garis besar hasil dari penelitian ini adalah rancang bangun dari sisi *backend* sistem informasi akademik yang berbentuk *Application Programming Interface*. Tahapan penelitian tersebut dapat dilihat pada gambar dibawah.



**Gambar 3.1** Tahapan Penelitian

Dalam membangun *Application Programming Interface* (API) pada aplikasi sistem informasi akademik ini, pemilihan bahasa pemrograman dan database merupakan hal

yang penting. Nodejs dan postgresql dari sisi backend development dapat diterapkan untuk di konsumsi pada client website[10]. Node.js berjalan diatas bahasa JavaScript, sehingga pembangunan program dilakukan menggunakan bahasa javascript[15]. Rincian tahapan penelitian menggunakan metode pengembangan TTD dapat dilihat pada penjelasan berikut.

### **3.1 Identifikasi Masalah**

Pada Pada tahap awal penelitian ini dilakukan identifikasi suatu masalah yang dihadapi dalam konteks penelitian ini. Identifikasi masalah akan dilakukan dengan mengidentifikasi kesenjangan pengetahuan atau tantangan yang relevan yang perlu dipecahkan atau diteliti lebih lanjut. Data yang digunakan pada penelitian rancang bangun Application Programming Interface (API) ini menggunakan sampling data yang didapatkan dari hasil wawancara terhadap pihak tata usaha SMK Muhammadiyah 7 Gondanglegi.

### **3.2 Studi Literatur**

Pada tahap studi literatur peneliti melakukan sebuah studi pustaka dan kajian penelitian terdahulu dari beberapa literatur seperti jurnal, thesis, buku dan artikel-artikel yang menunjang proses penelitian. Studi literatur membantu peneliti dalam merumuskan teori-teori yang dapat menyelesaikan masalah yang ada dalam penelitian ini. Adapun pembahasan mengenai studi literatur telah dijelaskan pada bab sebelumnya.

### **3.3 Analisis Kebutuhan**

Pada fase ini untuk mendapatkan analisis kebutuhan dibutuhkan pemahaman terhadap masalah, identifikasi terhadap stakeholder yang terlibat, pengumpulan informasi melalui wawancara dan studi literatur. Pada analisa kebutuhan sistem aplikasi sistem informasi akademik (SIKAD), tujuan utama yang ingin dicapai adalah bagaimana peneliti dapat memahami kebutuhan pengguna yang terlibat pada sistem yang akan dikembangkan. Hal ini mencakup identifikasi kebutuhan fungsional, non fungsional dan penentuan persyaratan dan batasan sistem[16][11]. Pada tahap ini juga digambarkan interaksi sistem dan pengguna kedalam use case diagram serta memvalidasi sistem yang dikembangkan akan sesuai dengan harapan pengguna, mengatasi masalah, dan memberikan solusi yang tepat untuk mencapai tujuan penelitian.

### 3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang memiliki proses atau layanan yang terdapat pada sistem yang dibuat berdasarkan bagaimana sistem harus bereaksi pada acion tertentu dan bagaimana proses sistem bekerja pada situasi tertentu. Kebutuhan fungsional memiliki gambaran fitur-fitur yang dapat memenuhi kebutuhan user. Pada penggalian kebutuhan pada penelitian ini peneliti berkesempatan mewawancarai pihak tata usaha ntuk mendapatkan kebutuhan dari pihak sekolah dan pada tahapan ini juga pembangunan API aplikasi SIAKAD memiliki 2 kebutuhan fungsional berdasarkan kebutuhan setiap stakeholder.

**Table 3.1** Kebutuhan Fungsional Admin

<b>Kebutuhan Fungsional</b>	<b>Keterangan</b>
Manajemen Surat	Dapat melakukan manajemen data keluar surat oleh staff tata usaha serta dapat melakukan rekap hasil data surat keluar
Manajemen jadwal kegiatan	Dapat melakukan manajemen data jadwal kegiatan sekolah oleh staff tata usaha sarta dapat melakukan rekap hasil jadwal pelajaran
Manajemen data siswa	Dapat melakukan manajemen data siswa oleh staff tata usaha
Manajemen data guru dan karyawan	Dapat melakukan manajemen data guru dan karyawan oleh staf tata usaha
Verifikasi Surat Dinas	Dapat melakukan verifikasi surat tugas bagi guru yang mengajukan surat dinas

**Table 3.2** Kebutuhan Fungsional Guru

<b>Kebutuhan Fungsional</b>	<b>Keterangan</b>
Login Guru	Dapat mengakses sistem untuk user masuk kedalam sistem

Permohonan surat	Dapat melakukan permohonan surat dinas
Rekap jadwal kegiatan sekolah	Dapat melihat hasil rekapan jadwal pelajaran
Manajemen Data Penilaian mata pelajaran	Dapat membuat penilaian mata pelajaran terhadap siswa

### 3.3.2 Kebutuhan Non-Fungsional

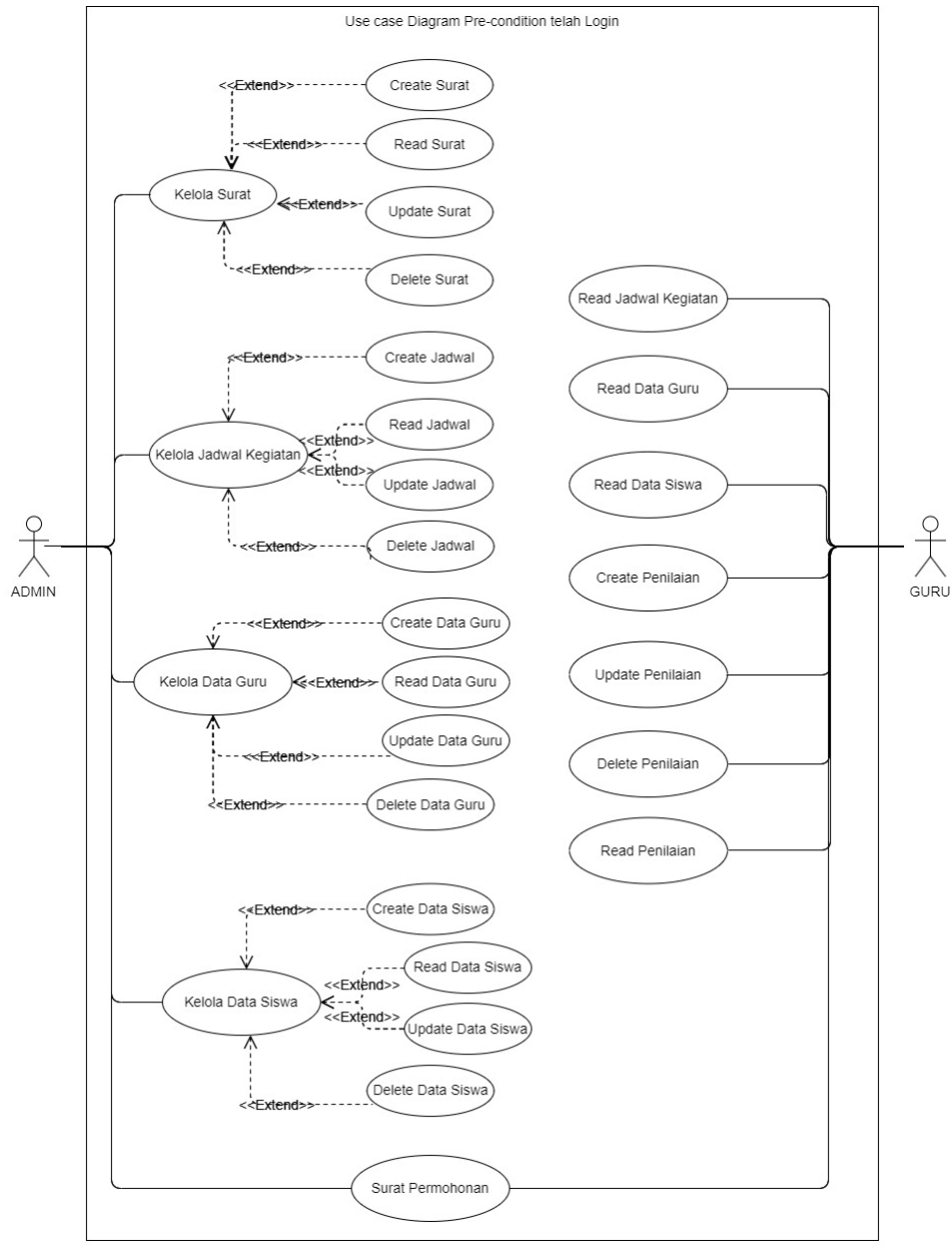
Kebutuhan Non-fungsional merupakan kebutuhan perilaku yang ada pada sistem serta menjadi batasan layanan atau fungsi. Berikut aspek non fungsional pada pembangunan API aplikasi SIAKAD.

**Table 3.3** Kebutuhan Non-Fungsional

Kebutuhan Non-Fungsional	Keterangan
User	Sistem dapat dijalankan di beberapa software web browser antara lain Google Chrome, Microsoft Edge serta Mozilla Firefox
Software	Mendukung keamanan dengan fitur login bagi user
Hardware	Perangkat komputer dengan koneksi internet

### 3.3.3 Use Case Diagram

Dalam tahapan pembangunan API aplikasi SIAKAD terdapat dua aktor yang dapat mengakses yaitu staf tata usaha sebagai admin yang memiliki peran mengelola sistem termasuk manajemen surat, jadwal kegiatan, data guru dan data siswa. sedangkan guru dapat membuat surat permohonan dinas, melihat data siswa dan guru serta membuat penilaian mata pelajaran, mendapatkan hasil rekapan jadwal pelajaran dan melihat data guru dan karyawan serta data siswa. Pada fase ini juga akan divisualisasikan gambaran sistem yang akan dibangun menggunakan *usecase diagram*.



**Gambar 3.2 Use Case SIAKAD**

Berdasarkan Gambar 3.2 maka diketahui kebutuhan fungsional dari setiap aktor dalam aplikasi Siakad yang akan dirancang. Use case diagram tersebut memiliki pre-condition yaitu telah login kedalam sistem pada setiap aktornya.

### 3.3.4 Use Case Description

*Use Case Description* merupakan deskripsi detail tentang suatu spesifikasi perangkat lunak yang menggambarkan secara terperinci bagaimana suatu sistem akan digunakan oleh stakeholder atau pengguna dalam suatu sistem.

**Table 3.4** Aktor Deskripsi

Stakeholder	Deskripsi
Admin	Admin merupakan staff tata usaha atau aktor yang terdaftar dalam sistem untuk mengelola aplikasi Sistem Informasi Akademik
Guru	Guru merupakan aktor terdaftar kedalam sistem yang memiliki peran terbatas yang hanya dapat melihat data-data setiap fitur seperti jadwal kegiatan, data siswa, data guru, mengambil rekapan jadwal kegiatan, dapat membuat surat permohonan dinas dan membuat penilaian terhadap mata pelajaran.

Berdasarkan use case diagram Sistem Informasi Akademik mempunyai 2 aktor dalam sistem yaitu admin dan guru. Berikut ini penjelasan dari masing-masing aktor pada use case deskripsi.

**Table 3.5** Use Case Description

Aktor	Use Case	Deskripsi
Admin	Manage Surat	Pada fitur ini admin dapat mengelola data surat keluar. Seperti dapat menambahkan (create), melihat (read), mengubah (update) dan menghapus (delete) data surat keluar
	Manage Jadwal Kegiatan	Pada fitur ini admin dapat mengelola data jadwal kegiatan. Seperti dapat menambahkan (create), melihat (read), mengubah (update) dan menghapus (delete) data jadwal kegiatan serta dapat secara otomatis mengenerate hasil rekapan data jadwal kegiatan kedalam excel
	Manage Data Siswa	Pada fitur ini admin dapat mengelola data siswa Seperti dapat menambahkan (create), melihat (read), mengubah (update) dan menghapus (delete) data siswa
	Manage Data Guru Dan Karyawan	Pada fitur ini admin dapat mengelola data guru dan karyawan Seperti dapat menambahkan

		(create), melihat (read), mengubah (update) dan menghapus (delete) data guru dan karyawan
Guru	View Jadwal Kegiatan Sekolah	Pada fitur ini, guru hanya bisa melihat secara otomatis data jadwal kegiatan sekolah berubah setelah admin menambahkan data kegiatan sekolah dan guru bisa mendownload hasil rekapan jadwal kegiatan sekolah
	View Data Siswa	Pada fitur ini, guru hanya bisa melihat secara otomatis data siswa berubah setelah admin menambahkan data siswa
	View Data Guru Dan Karyawan	Pada fitur ini, guru hanya bisa melihat secara otomatis data guru dan karyawan setelah admin menambahkan data guru dan karyawan
	Create Surat Permohonan Dinas	Pada fitur ini, guru dapat membuat surat permohonan dinas dan dapat mendownload kembali ketika admin telah menyetujuiinya
	Create Penilaian mata pelajaran	Pada fitur ini, guru dapat membuat penilaian mata pelajaran terhadap siswa
	Manage Penilaian mata pelajaran	Pada fitur ini, guru dapat membuat penilaian mata pelajaran terhadap siswa

### 3.4 Perancangan API

Tahap ini penulis mendesain arsitektur sistem untuk mendapat gambaran web service Pada tahap perancangan API dalam penelitian ini, tujuan yang ingin dicapai adalah merancang sebuah sistem yang memenuhi persyaratan fungsional dan non fungsional yang telah ditentukan. Tujuan ini mencakup merancang alur kerja, desain web service, dan desain basis data dari aplikasi sistem informasi akademik (SIKAD)[11]. Pada fase dan menggunakan Entity Relationship Diagram (ERD) untuk perancangan basis data serta memvisualisasikan data yang saling terkait satu sama lain dan saling berinteraksi didalam sistem.

### 3.5 Pembuatan API

Tahap ini adalah mengimplementasikan desain sistem yang telah dirancang sebelumnya menjadi sebuah sistem yang berfungsi. Pada pengembangannya metode yang digunakan adalah metode Test-Driven Development (TDD). Dengan menggunakan metode TDD proses penggabungan unit menjadi lebih mudah dan dipastikan bekerja dengan semestinya sebelum diintegrasikan satu sama lain, serta memudahkan untuk menemukan unit penyebab Bug atau eror dalam suatu sistem[8][17]. Pada metode TDD dalam mengembangkan sebuah sistem terdapat lima tahap yaitu Write a test, Run a test, Writing functional code, Re-run the test, dan Refactoring.

- a. *Write a test* adalah pengujian unit untuk setiap fitur yang akan dibuat, pengujian unit pada sistem ini akan menggunakan *tools Jest*. Pengujian unit yang akan dibuat meliputi fitur-fitur yang akan disediakan sistem. Pada saat membuat skenario pengujian membutuhkan deskripsi yang jelas tentang fitur-fitur tersebut, dari spesifikasi hingga kebutuhan yang dimulai dari input hingga output yang diharapkan.
- b. *Run a test* yaitu setelah pengujian untuk setiap fitur, pastikan ketika menjalankan pengujian fitur yang telah dibuat hasilnya gagal atau tidak memenuhi hasil yang diharapkan. Pada tahap ini biasa disebut fase merah karena kode fungsional belum dibuat.
- c. *Writing functional code* pada tahap ini menulis kode program yang menghasilkan kode sumber berdasarkan rancangan yang telah dibuat pada fase sebelumnya. Tujuan utama fase ini adalah membuat code program yang dapat lulus pengujian.
- d. *Re-run the test* pada tahap ini memastikan kode program yang dibuat, mulai dari semua fitur yang tersedia telah lulus pengujian. Jika tidak maka akan dilakukan iterasi dalam pembuatan kode program hingga semuanya lulus pengujian. Dengan melakukan iterasi secara berkala maka dapat memantau kinerja sistem sambil menghindari kesalahan atau bug pada kode program setelah melakukan perubahan.
- e. Refactoring pada tahap ini dilakukan perbaikan pada kode program yang masih berantakan, duplikat, dan tidak efisien dengan melakukan *iterasi* pengujian sambil memastikan perubahan yang dilakukan menghasilkan sesuatu yang serupa sebelum dilakukan *refactoring*.



### 3.6 Pengujian

Tahap ini merupakan tahap akhir dari penelitian. Laporan yang ditulis merupakan seluruh hasil dari, analisis dan pengujian, serta kesimpulan dari penelitian yang sudah dilakukan.

