



**KERJA PRAKTIK - IF184801**

**Pengembangan API Aplikasi Kinderfin : Sistem Monitoring Keuangan Kelompok Belajar dan Taman Kanak-kanak Berbasis Website**

Dinas Pendidikan Pemerintah Provinsi Jawa Timur  
Jalan Teknik Kimia, Sukolilo, Kota Surabaya, Jawa Timur, 60111  
Periode: 1 Juli 2024 - 30 Oktober 2024

**Oleh:**

Dimas Fadilah Akbar                      5025211010

**Pembimbing Jurusan**

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

**Pembimbing Lapangan**

Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2024



## **KERJA PRAKTIK – IF184801**

### **Pengembangan API Aplikasi Kinderfin: Sistem Monitoring Keuangan Kelompok Belajar dan Taman Kanak-kanak Berbasis Website**

Dinas Pendidikan Pemerintah Provinsi Jawa Timur  
Jalan Teknik Kimia, Sukolilo, Kota Surabaya, Jawa Timur, 60111  
Periode: 1 Juli 2024 - 30 Oktober 2024

Oleh:

Dimas Fadilah Akbar                      5025211010

#### **Pembimbing Jurusan**

Shitami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

#### **Pembimbing Lapangan**

Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.

DEPARTEMEN TEKNIK INFORMATIKA  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2024

*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

DAFTAR ISI.....	4
DAFTAR GAMBAR .....	8
DAFTAR TABEL.....	10
LEMBAR PENGESAHAN .....	12
KATA PENGANTAR .....	16
BAB I PENDAHULUAN .....	1
1.1.    Latar Belakang .....	1
1.2.    Tujuan .....	2
1.3.    Manfaat .....	2
1.4.    Rumusan Masalah .....	2
1.5.    Lokasi dan Waktu Kerja Praktik .....	2
1.6.    Metodologi Kerja Praktik.....	3
1.6.1.    Perumusan Masalah.....	3
1.6.2.    Studi Literatur .....	3
1.6.3.    Analisis Proses Bisnis dan Perancangan Desain Sistem..	3
1.6.4.    Implementasi Backend .....	3
1.6.5.    Pengujian dan Evaluasi .....	4
1.6.6.    Kesimpulan dan Saran.....	4
1.7.    Sistematika Laporan.....	4
1.7.1.    Bab I Pendahuluan .....	4
1.7.2.    Bab II Profil Perusahaan .....	4
1.7.3.    Bab III Tinjauan Pustaka.....	4
1.7.4.    Bab IV Analisis dan Perancangan Infrastruktur Sistem .	5

1.7.5.	Bab V Implementasi Sistem .....	5
1.7.6.	Bab VI Pengujian dan Evaluasi .....	5
1.7.7.	Bab VII Kesimpulan dan Saran .....	5
<b>BAB II PROFIL PERUSAHAAN .....</b>		<b>7</b>
2.1.	Profil Departemen Teknik Informatika ITS .....	7
2.2.	Lokasi.....	10
<b>BAB III TINJAUAN PUSTAKA .....</b>		<b>11</b>
3.1.	Pengantar Pengembangan Aplikasi Web .....	11
3.2.	Teknologi Backend .....	11
<b>BAB IV ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM .....</b>		<b>15</b>
4.1.	Definisi Umum Aplikasi .....	15
4.2.	Analisa Proses Bisnis .....	15
4.2.1.	Alur Kerja Aplikasi .....	15
4.2.2.	Otorisasi dan <i>Role</i> Aktor .....	16
4.2.3.	Database Design.....	24
4.2.4.	Arsitektur Backend.....	25
<b>BAB V IMPLEMENTASI SISTEM.....</b>		<b>29</b>
5.1.	Implementasi Modul .....	29
5.2.	Struktur Kerangka Kode Sumber .....	30
5.3.	API Implementation.....	32
<b>BAB VI PENGUJIAN DAN EVALUASI.....</b>		<b>37</b>
6.1.	Tujuan Pengujian .....	37
6.2.	Kriteria Pengujian .....	37
6.3.	Skenario Pengujian .....	37

6.4.	Evaluasi Pengujian.....	39
BAB VII KESIMPULAN DAN SARAN .....		41
7.1.	Kesimpulan .....	41
7.2.	Saran .....	41
LAMPIRAN.....		43
BIODATA PENULIS I.....		67

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR GAMBAR**

Gambar 4. 1 Alur Kerja Website Kinderfin .....	15
Gambar 4. 2 Use Case Diagram Aplikasi Kinderfin .....	21
Gambar 4. 3 Activity Diagram Fitur Daftar Ulang .....	22
Gambar 4. 4 Activity Diagram Fitur SPP, Komite, dan Ekstrakurikuler .....	22
Gambar 4. 5 Activity diagram fitur PPDB .....	23
Gambar 4. 6 Activity Diagram Fitur Rumah Tangga Sekolah .....	23
Gambar 4. 7 Activity Diagram Fitur Gaji Guru .....	23
Gambar 4. 8 Desain Antarmuka Halaman Pengeluaran Rumah Tangga Sekolah.....	24
Gambar 4. 9 Arsitektur Backemd.....	26
Gambar 4. 10 Komunikasi Antar Modul.....	27
Gambar 5. 1Modul Backend yang dikembangkan .....	29
Gambar 5. 2 Struktur Folder Program.....	30
Gambar 5. 3 Stuktur Seluruh Program .....	31



*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 4. 1 Tabel Otorisasi dan Role Aktor.....	16
Tabel 5. 1 Daftar API.....	32
Tabel 6. 1 <i>Contoh Skenario Pengujian Postman</i> .....	38
Tabel 6. 2 Evaluasi Pengujian.....	39

*[Halaman ini sengaja dikosongkan]*

**LEMBAR PENGESAHAN  
KERJA PRAKTIK**

**Pengembangan API Aplikasi Kinderfin: Sistem Monitoring  
Keuangan Kelompok Belajar dan Taman Kanak-kanak  
Berbasis Website**

Oleh:

Dimas Fadilah Akbar

5025211010

Disetujui oleh Pembimbing Kerja Praktik:

1. Shitami Chusnul Hidayati,  
S.Kom., M.Sc., Ph.D.  
NIP. 1987202012004



(Pembimbing Departemen)

2. Adhatus Solichah  
Ahmadiyah, S.Kom., M.Sc.  
NIP. 198508262015042002



(Pembimbing Lapangan)

*[Halaman ini sengaja dikosongkan]*

## **Pengembangan API Aplikasi Kinderfin: Sistem Monitoring Keuangan Kelompok Belajar dan Taman Kanak-kanak Berbasis Website**

Nama Mahasiswa : Dimas Fadilah Akbar  
NRP : 5025211010  
Departemen : Teknik Informatika FTEIC-ITS  
Pembimbing Departemen : Shitami Chusnul Hidayati, S.Kom.,  
M.Sc., Ph.D.  
Pembimbing Lapangan : Adhatas Solichah Ahmadiyah, S.Kom.,  
M.Sc.

### **ABSTRAK**

*Proyek “Pengembangan API Aplikasi Kinderfin: Sistem Monitoring Keuangan Kelompok Belajar dan Taman Kanak-kanak Berbasis Website”* dirancang untuk mempermudah pengelolaan administrasi keuangan sekolah tingkat taman kanak-kanak di Surabaya. Aplikasi ini mendukung berbagai kebutuhan keuangan, seperti pembayaran PPDB, daftar ulang, SPP, biaya komite, ekstrakurikuler, hingga pencatatan gaji guru, dan pengeluaran sekolah.

*Aplikasi dibuat dengan menggunakan framework Express.js dan database PostgreSQL dengan menerapkan arsitektur clan modular sehingga tiap modul terpisah dengan baik dan mempermudah dalam pemeliharaan aplikasi. Aplikasi backend menggunakan dockploy agar aplikasi backend dapat dijalankan dan diakses secara public. Proyek ini diharapkan menjadi solusi efektif untuk meningkatkan efisiensi keuangan sekolah*

***Kata Kunci : Express, Dockploy, Arsitektur Clean Modular, PostgreSQL***

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

Puji syukur ke hadiran Tuhan Yang Maha Esa atas limpahan rahmat, karunia, dan penyertaan-Nya sehingga kami dapat menyelesaikan laporan kerja praktik yang berjudul “*Pengembangan Backend Aplikasi Kinderfin: Sistem Monitoring Keuangan Kelompok Belajar dan Taman Kanak-kanak Berbasis Website.*” Laporan ini merupakan salah satu kewajiban akademik sebagai mahasiswa Departemen Teknik Informatika Institut Teknologi Sepuluh Nopember.

Kami menyadari bahwa laporan ini masih jauh dari sempurna, baik dalam pelaksanaan kerja praktik maupun penyusunannya. Kendati demikian, kami berharap laporan ini dapat memberikan manfaat, menambah wawasan pembaca, serta menjadi referensi yang berguna bagi pihak-pihak terkait.

Pada kesempatan ini, kami ingin menyampaikan rasa terima kasih kepada semua pihak yang telah membantu dalam pelaksanaan kerja praktik maupun penyusunan laporan ini, baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang kami atas dukungan moral dan doa yang tiada henti.
2. Ibu Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D., selaku dosen pembimbing kerja praktik dan sekaligus koordinator kerja praktik, atas bimbingannya dan arahannya yang sangat berarti.
3. Ibu Adhatus Solichah Ahmadiyah, S.Kom., M.Sc., selaku pembimbing lapangan kerja praktik berlangsung, atas ilmu dan pengalaman yang telah dibagikan.
4. Teman-teman seperjuangan yang senantiasa memberikan dukungan dan semangat dalam pelaksanaan kerja praktik ini.

Surabaya, 2 December 2024

Dimas Fadilah Akbar



*[Halaman ini sengaja dikosongkan]*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Di era digital saat ini, teknologi informasi telah menjadi salah satu pilar penting dalam pengelolaan berbagai aspek kehidupan, termasuk dalam dunia pendidikan. Banyak sekolah khususnya taman kanak-kanak (TK), menghadapi tantangan yang semakin kompleks dalam mengelola keuangan dan administrasi. Proses administrasi yang manual sering kali memakan waktu, rentan terhadap kesalahan, dan kurang efisien. Hal ini dapat mengganggu fokus utama lembaga pendidikan, yaitu memberikan pendidikan yang berkualitas kepada anak-anak.

Dengan meningkatnya kebutuhan akan sistem pengelolaan yang lebih efisien dan transparan, hadirnya aplikasi Kinderfin (*Kindergarten Financial*) menjadi solusi yang tepat. Kinderin dirancang khusus untuk memfasilitasi pengelolaan keuangan dan administrasi di lingkungan taman kanak-kanak. Aplikasi ini menawarkan berbagai fitur yang mendukung keuangan, mulai dari pembayaran SPP, komite, ekstrakurikuler, daftar ulang, dan juga pendaftaran peserta didik baru, hingga pencatatan pengeluaran dan pemasukan rumah tangga dan gaji karyawan.

Dengan Kinderfin, diharapkan proses administrasi yang sebelumnya rumit dapat disederhanakan. Aplikasi ini tidak hanya membantu pihak sekolah dalam mengelola keuangan dengan lebih baik, tetapi juga memberikan kemudahan bagi orang tua dalam melakukan proses pembayaran tagihan secara online. Selain itu, pencatatan yang terintegrasi akan meminimalisir kesalahan dan meningkatkan transparansi dalam pengelolaan dana sekolah.

Melalui laporan ini, kami akan menjelaskan lebih lanjut mengenai fitur-fitur yang ada dalam aplikasi Kinderfin, manfaat yang dapat diperoleh oleh sekolah, serta dampak positif yang diharapkan bagi pengelolaan administrasi dan keuangan di taman kanak-kanak. Dengan demikian, Kinderfin diharapkan dapat menjadi alat yang efektif dalam mendukung perkembangan pendidikan di Indonesia.

## **1.2. Tujuan**

Tujuan dari kerja praktik ini adalah untuk menyelesaikan kewajiban nilai kerja praktik sebesar 4 SKS serta membantu lembaga pendidikan, khususnya kelompok belajar dan taman kanak-kanak, dalam meningkatkan efisiensi, efektivitas, dan transparansi pengelolaan keuangan.

## **1.3. Manfaat**

Manfaat yang diperoleh dengan adanya aplikasi web Kinderfin adalah mempermudah pengelolaan keuangan di lembaga pendidikan, khususnya kelompok belajar dan taman kanak-kanak. Kinderfin dapat mengurangi risiko kesalahan pencatatan, mempercepat waktu pemrosesan transaksi, dan memudahkan akses data keuangan secara cepat dan akurat.

## **1.4. Rumusan Masalah**

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana merancang arsitektur backend yang efisien untuk mendukung berbagai fitur aplikasi Kinderfin, seperti pengelolaan pembayaran, pendaftaran peserta didik baru, dan pencatatan keuangan.
2. Bagaimana mengintegrasikan berbagai fitur transaksi keuangan dalam satu aplikasi yang efisien dan efektif?

## **1.5. Lokasi dan Waktu Kerja Praktik**

Kerja praktik ini dilakukan secara hybrid selama 4 bulan, dimulai dari tanggal 1 Juli 2024 hingga 30 Oktober 2024. Dalam satu minggu, jadwal kerja terdiri dari 4 hari *Work from Home* (WFH) dan 1 hari *Work from Office* (WFO). Pertemuan *offline* dijadwalkan setiap hari Jumat untuk memonitoring progress mingguan dan melakukan diskusi langsung dengan tim.

## **1.6. Metodologi Kerja Praktik**

Metodologi dalam pembuatan buku kerja praktik meliputi:

### **1.6.1. Perumusan Masalah**

Untuk mengetahui kebutuhan dari aplikasi web Kinderfin, kami melakukan diskusi mendalam dengan tim pengembang dan pemangku kepentingan. Dalam diskusi tersebut penulis mengidentifikasi kebutuhan spesifik dari sisi backend, termasuk pengelolaan database, keamana data, dan integrasi antar sistem. Hal ini penting untuk memastikan bahwa arsitektur backend yang dirancang dapat mendukung semua fitur yang diharapkan oleh pengguna.

### **1.6.2. Studi Literatur**

Setelah mendapatkan gambaran mengenai sistem yang diinginkan, kami melakukan studi literatur untuk memahami teknologi yang akan digunakan dalam pengembangan backend. Kami memilih untuk menggunakan Express Js dengan Typescript untuk membangun API yang efisien dan aman, serta mempelajari praktik terbaik dalam pengelolaan database dan otorisasi pengguna terhadap fitur-fitur yang dikembangkan. Tinjauan ini membantu kami dalam menentukan pendekatan yang tepat untuk pengembangan backend aplikasi Kinderfin.

### **1.6.3. Analisis Proses Bisnis dan Perancangan Desain Sistem**

Pada tahap ini, kami melakukan analisis terhadap kebutuhan sistem berdasarkan hasil diskusi dan studi literatur. Kami merancang arsitektur backend yang mencakup desain database, alur kerja API, dan integrasi dengan sistem frontend. Desain sistem ini mencakup struktur tabel untuk pengelolaand data pengguna, transaksi keuangan, dan pencatatan lainnya, serta alur bisnis untuk memastikan bahwa data dapat diproses dengan efisien dan aman.

### **1.6.4. Implementasi Backend**

Implementasi backend aplikasi Kinderfin dilakukan dengan mengambangkan Rest API menggunakan Express.js dan Typescript. Pada tahap ini, kami merealisasikan desain database dan alur bisnis yang dibuat menjadi kode yang dapat

dijalankan. Proses implementasi dimulai dengan membangun struktur database, mendefinisikan model data, serta mengembangkan endpoint API untuk berbagai fitur, seperti pendaftaran peserta didik baru, pembayaran tagihan SPP, Komite, Ekstrakurikuler, dan Daftar Ulang, serta pencatatan gaji, dan pencatatan keuangan rumah tangga.

#### **1.6.5. Pengujian dan Evaluasi**

Setelah implementasi backend selesai, kami melakukan pengujian menyeluruh terhadap API untuk memastikan semua fungsi berjalan dengan baik. Pengujian meliputi pengujian fungsional di setiap endpoint, pengujian integrasi untuk memastikan komunikasi yang baik antara frontend dan backend, serta pengujian keamanan untuk melindungi data sensitif pengguna. Evaluasi dilakukan berdasarkan hasil pengujian untuk mengidentifikasi dan memperbaiki masalah yang ditemukan.

#### **1.6.6. Kesimpulan dan Saran**

Setelah keseluruhan sistem backend aplikasi Kinderfin dievaluasi dan diperbaiki, aplikasi ini telah siap digunakan. Kami juga memberikan saran untuk pemeliharaan berkelanjutan dan pengembangan aplikasi tetap relevan dan memenuhi kebutuhan pengguna.

### **1.7. Sistematika Laporan**

#### **1.7.1. Bab I Pendahuluan**

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

#### **1.7.2. Bab II Profil Perusahaan**

Bab ini berisi gambaran umum Departemen Teknik Informatika ITS mulai dari profil, lokasi perusahaan.

#### **1.7.3. Bab III Tinjauan Pustaka**

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

- 1.7.4. Bab IV Analisis dan Perancangan Infrastruktur Sistem**  
Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.
- 1.7.5. Bab V Implementasi Sistem**  
Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi aplikasi.
- 1.7.6. Bab VI Pengujian dan Evaluasi**  
Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.
- 1.7.7. Bab VII Kesimpulan dan Saran**  
Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **PROFIL PERUSAHAAN**

#### **2.1. Profil Departemen Teknik Informatika ITS**

Departemen Teknik Informatika, yang berada di bawah Fakultas Teknologi Elektro dan Informatika Cerdas di Institut Teknologi Sepuluh Nopember (ITS), adalah lembaga pendidikan tinggi yang mengkhususkan diri pada studi Informatika. Informatika sendiri merupakan disiplin ilmu yang melibatkan praktik pemrosesan informasi dan rekayasa sistem. Disiplin ini merupakan terapan dari sains informasi yang memperhatikan interaksi antara manusia dan informasi melalui konstruksi antarmuka, teknologi, dan sistem.

Seiring dengan kemajuan teknologi komputer, kebutuhan akan pemrosesan informasi secara digital oleh individu dan organisasi terus meningkat, baik dari segi kuantitas maupun kualitas. Hal ini mendorong pentingnya pembelajaran dan penelitian di bidang Informatika, yang mencakup aspek komputasi, matematika, biologi, kognitif, serta aspek sosial. Selain itu, disiplin ini juga mempertimbangkan studi mengenai dampak sosial dari teknologi informasi, dengan tujuan untuk menghasilkan inovasi teknologi yang bermanfaat bagi masyarakat.

Departemen Teknik Informatika ITS menawarkan program studi di berbagai jenjang untuk memenuhi kebutuhan pendidikan tinggi di bidang Informatika, di antaranya:

1. Sarjana (S1): Teknik Informatika, Rekayasa Kecerdasan Artifisial, Teknik Informatika Internasional, Rekayasa Perangkat Lunak.
2. Magister (S2): Teknik Informatika.
3. Doktor (S3): Ilmu Komputer.

Sejalan dengan visi ITS yaitu menjadi perguruan tinggi dengan reputasi internasional dalam ilmu pengetahuan, teknologi, dan seni, terutama yang menunjang industri dan kelautan yang berwawasan lingkungan, maka visi Departemen



Informatika adalah menjadi inovator bidang informatika yang unggul di tingkat nasional dengan reputasi internasional, serta berperan aktif dalam upaya memajukan dan mensejahterakan bangsa. Visi Departemen Teknik Informatika (DTIF) berfokus untuk menjadi lembaga pendidikan yang unggul dan inovatif-inventif dalam informatika cerdas, mendukung transformasi digital dan berkontribusi kepada masyarakat dengan reputasi internasional.

Departemen Teknik Informatika ITS memiliki misi untuk:

1. Menyelenggarakan proses pembelajaran yang berkualitas, dan memenuhi standar nasional maupun internasional.
2. Melaksanakan penelitian yang inovatif, bermutu, dan bermanfaat.
3. Meningkatkan pemanfaatan teknologi informasi dan komunikasi untuk masyarakat.
4. Menjalin kemitraan dengan berbagai lembaga, baik di dalam maupun di luar negeri.

Departemen Teknik Informatika ITS memiliki laboratorium yang mendukung kegiatan belajar dan penelitian di berbagai bidang minat, yaitu:

1. Laboratorium Manajemen Cerdas Informasi

Laboratorium ini menawarkan bidang keahlian dalam analisis, sintesis, dan evaluasi proses bisnis serta sistem informasi pada sistem enterprise, implementasi rekayasa pengetahuan, investigasi standar dan tata kelola teknologi informasi, manajemen proyek dan sumber daya manusia, serta pengembangan solusi basis data terdistribusi dan teknologi Big Data.

2. Laboratorium Rekayasa Perangkat Lunak

Laboratorium ini menawarkan bidang keahlian yang berfokus pada pengujian perangkat lunak, manajemen proyek perangkat lunak, kemampuan mengurangi risiko kesalahan perangkat lunak, serta pengembangan perangkat lunak game.

3. Laboratorium Komputasi Berbasis Jaringan

Laboratorium ini menekankan pada kemampuan lulusan dalam membangun infrastruktur jaringan yang aman, mengembangkan sistem grid, membangun aplikasi jaringan sesuai standar, dan membangun aplikasi multimedia berbasis jaringan.

#### 4. Laboratorium Komputasi Cerdas dan Visi

Laboratorium ini menawarkan bidang keahlian yang menekankan kemampuan dalam memanipulasi dan menganalisis data citra untuk berbagai aplikasi (misalnya, biomedis, industri), menerapkan metode sistem cerdas, serta memodelkan dan mengoptimalkan sistem nyata.

#### 5. Laboratorium Teknologi Jaringan dan Keamanan Siber Cerdas

Laboratorium ini menyediakan bidang keahlian yang berfokus pada kemampuan lulusan dalam membangun berbagai arsitektur jaringan sesuai standar teknologi terkini dan menerapkan keamanan jaringan.

#### 6. Laboratorium Grafika, Interaksi, dan Game

Laboratorium ini menawarkan bidang keahlian yang menekankan kemampuan dalam desain, pengembangan, dan dokumentasi proses pembuatan game sesuai standar, termasuk pembuatan model 3D, pemrograman realitas virtual, dan pengembangan aplikasi 3D berbasis game engine.

#### 7. Laboratorium Algoritma dan Pemrograman

Laboratorium ini menawarkan bidang keahlian yang berfokus pada kemampuan untuk merancang dan menganalisis algoritma guna menyelesaikan masalah secara efektif dan efisien, mengaplikasikan model pemrograman untuk berbagai bahasa, serta memilih bahasa pemrograman yang sesuai untuk mengembangkan aplikasi berbasis kerangka kerja dan perangkat bergerak.

#### 8. Laboratorium Pemodelan dan Komputasi Terapan

Laboratorium ini mendukung riset dan kerja sama industri dalam bidang pemodelan & simulasi, peramalan sains, optimasi, serta komputasi saintifik. Mata kuliah pendukung termasuk

Matematika Diskrit, Aljabar Linier & Matriks, Teori Graf & Otomata, Komputasi Numerik, Probabilitas & Statistik, Riset Operasi, Pemodelan & Simulasi, Analisis Data Multivariat, serta mata kuliah pascasarjana terkait lainnya.

## **2.2. Lokasi**

Jalan Teknik Kimia, Gedung Departemen Teknik Informatika Kampus Institut Teknologi Sepuluh Nopember Surabaya, ITS, Sukolilo, Surabaya 60111

## **BAB III**

### **TINJAUAN PUSTAKA**

#### **3.1. Pengantar Pengembangan Aplikasi Web**

Penggunaan aplikasi web telah meningkat secara signifikan dalam berbagai bidang, termasuk individu, kelompok, organisasi, dan pemerintah, sebagai sarana untuk bertukar informasi dan mendukung tugas-tugas yang terkait dengan bisnis. Aplikasi web terdiri dari server web dan halaman web, yang biasanya ditulis dalam format HTML, PHP, dan bahasa pemrograman web lainnya, yang diakses melalui protokol HTTP. Menurut survei yang dilakukan oleh Netcraft, hingga Januari 2018, Apache merupakan aplikasi web server yang paling banyak digunakan di dunia (Muzaki et al, 2020)

Pengembangan aplikasi web mengalami transformasi signifikan seiring dengan kemajuan teknologi dan kebutuhan pengguna. Aplikasi web, yang didefinisikan sebagai aplikasi yang berjalan secara online dan diakses melalui web browser, telah menjadi bagian dari integral dari pengalaman digital sehari-hari. Dalam konteks ini, *Software as a Service* (Saas) muncul sebagai model baru yang mendukung penyediaan aplikasi *on-demand*, di mana pengguna tidak perlu melakukan instalasi atau pembaruan perangkat lunak di komputer mereka. Sebaliknya, semua perangkat lunak di hosting di server dan dapat diakses melalui Internet (Herrero et al., 2011)

#### **3.2. Teknologi Backend**

Pemilihan teknologi backend yang tepat sangat penting dalam pengembangan aplikasi web modern. Teknologi yang digunakan tidak hanya mempengaruhi performa aplikasi, tetapi juga kemudahan pengembangan, pemeliharaan, dan skalabilitas. Dalam konteks ini, beberapa teknologi yang sering digunakan adalah Node.js, Express.js, Typescript, PostgreSQL, Docker, dan Dockploy.

### 3.3. Node.js

Node.js adalah runtime JavaScript yang dibangun di atas mesin V8 milik Google Chrome. Dengan Node.js, pengembangan dapat menjalankan JavaScript di sisi server, yang memungkinkan penggunaan bahasa yang sama di seluruh stack pengembangan, dari frontend hingga backend. Node.js menjadikannya sangat efisien dalam menangani banyak koneksi secara bersamaan. Hal ini membuat Node.js ideal untuk aplikasi *realtime* seperti *chat applications*, kolaborasi daring, dan aplikasi *streaming*. Selain itu, ekosistem npm (Node Package Manager yang kaya menyediakan berbagai paket dan modul yang dapat digunakan dalam mempercepat pengembangan (Node.js, 2024).

### 3.4. Express.js

Express.js adalah *framework* web minimalis yang dibangun di atas Node.js. *Framework* ini menyediakan struktur dasar untuk membangun aplikasi web dan API dengan cepat dan efisien. Dengan Express, pengembangan dapat dengan mudah mengelola *routing*, *middleware*, dan HTTP *requests*. Fitur seperti *dynamic routing* dan dukungan untuk berbagai jenis *middleware* memungkinkan pengembang untuk menambahkan fungsionalitas tambahan dengan mudah. Express juga mendukung template engine seperti Pug dan EJS, yang memudahkan dalam menghasilkan *dynamic HTML*. Karena kesederhanaan dan fleksibilitasnya, Express telah menjadi salah satu *framework* paling populer dalam ekosistem Node.js (Express.js, 2024).

### 3.5. TypeScript

TypeScript adalah superset dari JavaScript yang menambahkan tipe statis ke dalam bahasa tersebut. Dengan menggunakan TypeScript, pengembang dapat menulis kode yang lebih terstruktur dan mudah dipelihara, karena TypeScript mendukung pengenalan tipe data dan interface. Fitur ini membantu mendeteksi kesalahan lebih awal dalam proses pengembangan, sebelum kode dieksekusi. TypeScript juga menyediakan alat bantu pengembangan seperti *autocompletion* dan *type checking*, yang meningkatkan produktivitas pengembang. Banyak proyek modern yang menggunakan TypeScript, terutama dalam kombinasi dengan Node.js dan

Express.js, untuk meningkatkan kualitas dan keamanan kode (Typescript, 2024).

### 3.6. PostgreSQL

PostgreSQL adalah sistem manajemen basis data relasional open-source yang dikenal karena kemampuannya untuk menangani data yang kompleks dan kepatuhannya terhadap standar SQL. PostgreSQL mendukung berbagai tipe data, termasuk JSONB, yang memungkinkan penyimpanan data semi-terstruktur. Fitur-fitur seperti transaksi ACID, *indexing* yang kuat, dan kemampuan untuk menjalankan query yang kompleks menjadikan PostgreSQL pilihan yang sangat baik untuk aplikasi yang memerlukan integritas data dan performa tinggi. PostgreSQL juga memiliki komunitas yang aktif dan banyak dukungan untuk berbagai bahasa pemrograman, termasuk TypeScript (PostgreSQL, 2024).

### 3.7. Docker

Docker adalah platform yang memungkinkan pengembang untuk mengemas aplikasi dan semua dependensinya ke dalam kontainer. Kontainer Docker menyediakan lingkungan yang konsisten dan terisolasi untuk menjalankan aplikasi, sehingga memudahkan pengelolaan dan penyebaran aplikasi di berbagai lingkungan, dari pengembangan hingga produksi. Dengan Docker, pengembang dapat menghindari masalah "works on my machine" yang sering terjadi ketika aplikasi berpindah dari satu lingkungan ke lingkungan lainnya. Docker juga mendukung orkestrasi dengan alat seperti Kubernetes, yang memungkinkan pengelolaan kontainer dalam skala besar (Docker, 2024).

### 3.8. Dockploy

Dockploy adalah alat yang dirancang untuk menyederhanakan proses penyebaran aplikasi berbasis Docker. Dengan Dockploy, pengembang dapat dengan mudah mengotomatiskan penyebaran aplikasi ke lingkungan produksi. Dockploy memungkinkan integrasi yang mulus dengan pipeline CI/CD, sehingga memudahkan pengelolaan versi aplikasi dan proses pengujian otomatis. Selain itu, Dockploy menyediakan fitur untuk memantau dan mengelola kontainer yang sedang berjalan, memastikan bahwa aplikasi tetap berjalan dengan baik dan dapat diakses oleh pengguna (Dockploy, 2024).

*[Halaman ini sengaja dikosongkan]*

## BAB IV ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM

### 4.1. Definisi Umum Aplikasi

Secara umum, Kinderfin merupakan sistem monitoring keuangan untuk staff, guru, serta orang tua wali murid kelompok belajar dan taman kanak-kanak berbasis website. Sistem ini memiliki delapan *view* yang meliputi *view* login, *view* pendaftaran orang tua, *view dashboard*, *view* PPDB, *view* daftar ulang, *view* SPP, *view* gaji guru, dan *view* pengeluaran rumah tangga sekolah. Sementara pengguna aplikasi Kinderfin terbagi menjadi lima jenis role, yaitu admin, bendahara, sekretaris, guru, dan wali murid.

### 4.2. Analisa Proses Bisnis

Pada bagian ini akan dijelaskan mengenai proses bisnis yang telah dirancang pada aplikasi Kinderfin.

#### 4.2.1. Alur Kerja Aplikasi



Gambar 4. 1 Alur Kerja Website Kinderfin



Website Kinderfin memulai alur kerjanya dengan mengidentifikasi *role* pengguna yang telah login lalu menampilkan dashboard dengan menu-menu sesuai *role* pengguna. Kemudian pengguna dapat memilih salah satu menu dari *dashboard*. Aplikasi Kinderfin akan menampilkan data sesuai menu yang dipilih oleh pengguna. Sesuai otorisasi yang dimiliki, pengguna dapat melakukan operasi-operasi pada data yang ditampilkan, seperti operasi *read*, *update*, *delete*, dan *create*.

Melalui pembuatan *request* API dari website Kinderfin, server Kinderfin akan melakukan operasi pada database sesuai *request* dari website. Sehingga data milik sekolah tetap sinkron ketika diakses oleh pengguna atau perangkat yang berbeda.

#### 4.2.2. Otorisasi dan Role Aktor

Terdapat lima jenis aktor, yaitu admin, sekretaris, bendahara, guru, dan orang tua. *Role* admin adalah jenis *role* yang merupakan *super user*, yaitu pengguna yang dapat mengakses semua fitur. Otorisasi dan *role* pengguna tersebut dapat dilihat di Tabel 4.1

Tabel 4. 1 Tabel Otorisasi dan Role Aktor

Role	Otorisasi	Deskripsi
Admin	Melakukan operasi <i>create</i> , <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman gaji guru.	Melihat semua riwayat pembayaran gaji guru, membuat entri pembayaran gaji baru, memperbarui data riwayat pembayaran gaji, dan menghapus data pembayaran gaji.

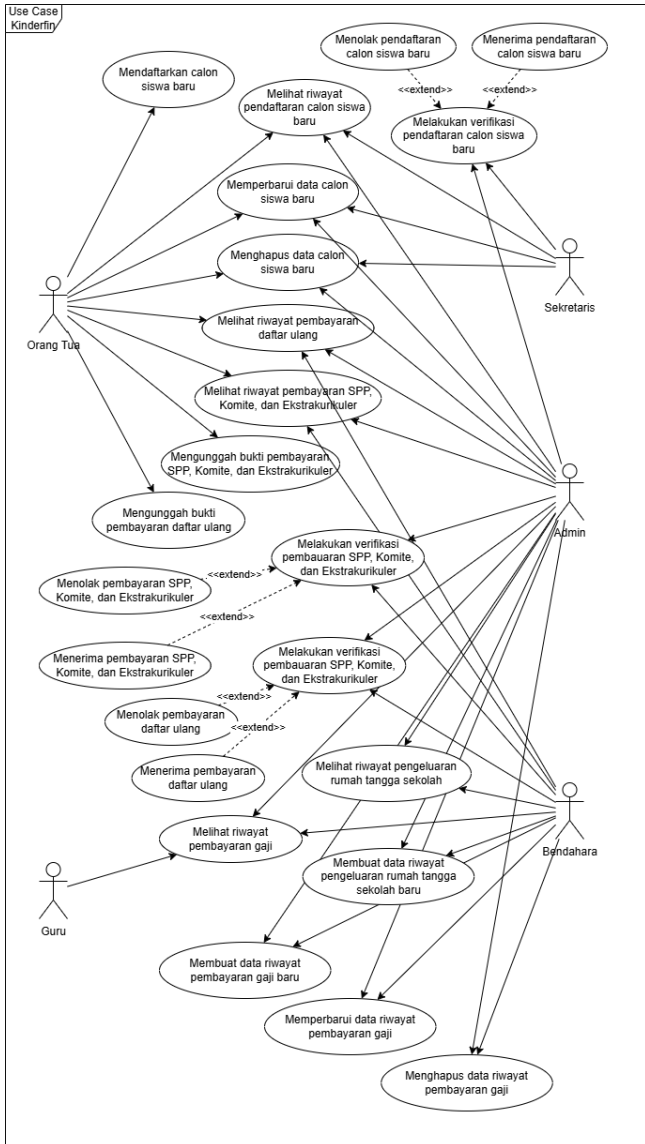
Role	Otorisasi	Deskripsi
	Melakukan operasi <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman PPDB.	Melihat semua riwayat pendaftaran calon siswa baru, menghapus data riwayat pendaftaran calon siswa baru, melakukan verifikasi pendaftaran calon siswa baru, serta meninggalkan catatan untuk orang tua yang membuat entri pendaftaran calon siswa baru.
	Melakukan operasi <i>create</i> , <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman daftar ulang.	Melihat semua riwayat tagihan daftar ulang, membuat tagihan daftar ulang baru, melakukan pengaturan diskon pembayaran daftar ulang, melihat semua riwayat pembayaran daftar ulang, dan melakukan verifikasi pembayaran daftar ulang.
	Melakukan operasi <i>create</i> , <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman SPP, komite, dan ekstrakurikuler	Melihat semua riwayat tagihan SPP, membuat tagihan SPP baru, melakukan pengaturan diskon pembayaran SPP, melihat semua riwayat pembayaran SPP, dan melakukan verifikasi pembayaran SPP.
	Melakukan operasi <i>create</i> , <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman pengeluaran rumah tangga sekolah.	Melihat semua riwayat pengeluaran rumah tangga sekolah, membuat entri data pengeluaran rumah tangga sekolah baru, memperbarui data, serta menghapus data riwayat pengeluaran rumah tangga sekolah.

Role	Otorisasi	Deskripsi
Sekretaris	Melakukan operasi <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman PPDB.	Melihat semua riwayat pendaftaran calon siswa baru, menghapus data riwayat pendaftaran calon siswa baru, melakukan verifikasi pendaftaran calon siswa baru, serta meninggalkan catatan untuk orang tua yang membuat entri pendaftaran calon siswa baru.
Bendahara	Melakukan operasi <i>create</i> , <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman gaji guru.	Melihat semua riwayat pembayaran gaji guru, membuat entri pembayaran gaji baru, memperbarui data riwayat pembayaran gaji, dan menghapus data pembayaran gaji.
	Melakukan operasi <i>create</i> , <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman daftar ulang.	Melihat semua riwayat tagihan daftar ulang, membuat tagihan daftar ulang baru, melakukan pengaturan diskon pembayaran daftar ulang, melihat semua riwayat pembayaran daftar ulang, dan melakukan verifikasi pembayaran daftar ulang.
	Melakukan operasi <i>create</i> , <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman SPP, komite, dan ekstrakurikuler	Melihat semua riwayat tagihan SPP, membuat tagihan SPP baru, melakukan pengaturan diskon pembayaran SPP, melihat semua riwayat pembayaran SPP, dan melakukan verifikasi pembayaran SPP.

Role	Otorisasi	Deskripsi
	Melakukan operasi <i>create</i> , <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman pengeluaran rumah tangga sekolah.	Melihat semua riwayat pengeluaran rumah tangga sekolah, membuat entri data pengeluaran rumah tangga sekolah baru, memperbaiki data, serta menghapus data riwayat pengeluaran rumah tangga sekolah.
Guru	Melakukan operasi <i>read</i> pada halaman gaji guru.	Melihat riwayat pembayaran gaji sesuai pengguna guru yang sedang <i>login</i> .
Orang tua/wali murid	Melakukan operasi <i>read</i> , <i>update</i> , dan <i>delete</i> pada halaman PPDB.	Melihat riwayat pendaftaran calon siswa baru yang pernah dibuat, menghapus atau memperbaiki data riwayat pendaftaran calon siswa baru yang belum diverifikasi, serta membuat data pendaftaran calon siswa baru
	Melakukan operasi <i>create</i> dan <i>read</i> pada halaman daftar ulang.	Melihat riwayat pembayaran daftar ulang, mengunggah bukti pembayaran sebagai data pembayaran daftar ulang baru, melihat rekap pembayaran daftar ulang per anak, serta melihat informasi pembayaran daftar ulang.

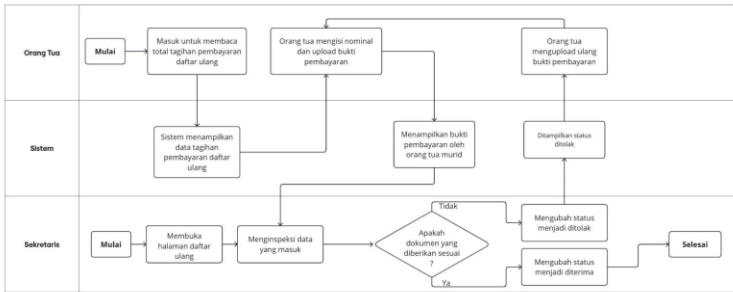
Role	Otorisasi	Deskripsi
	Melakukan operasi <i>create</i> dan <i>read</i> pada halaman SPP, komite, dan ekstrakurikuler	Melihat riwayat pembayaran SPP, mengunggah bukti pembayaran sebagai data pembayaran SPP baru, melihat rekap pembayaran SPP per anak, serta melihat informasi pembayaran SPP.

Gambar 4.2 merupakan use case diagram untuk aplikasi Kinderfin.



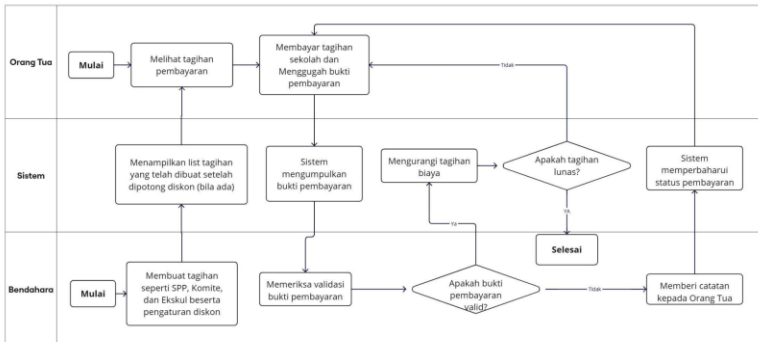
Gambar 4. 2 Use Case Diagram Aplikasi Kinderfin

Gambar 4.3 merupakan *activity diagram* fitur daftar ulang.



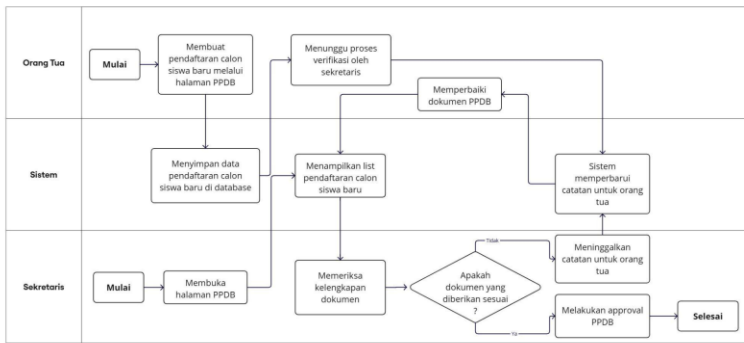
Gambar 4. 3 Activity Diagram Fitur Daftar Ulang

Gambar 4.4 merupakan *activity diagram* fitur SPP, Komite, dan Ekstrakurikuler.



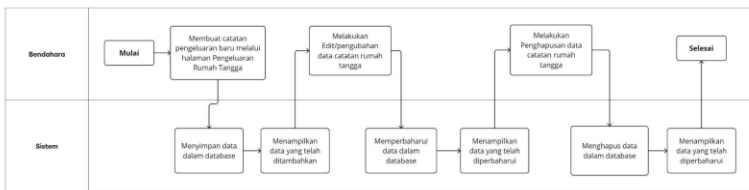
Gambar 4. 4 Activity Diagram Fitur SPP, Komite, dan Ekstrakurikuler

Gambar 4.5 merupakan *activity diagram* fitur PPDB.



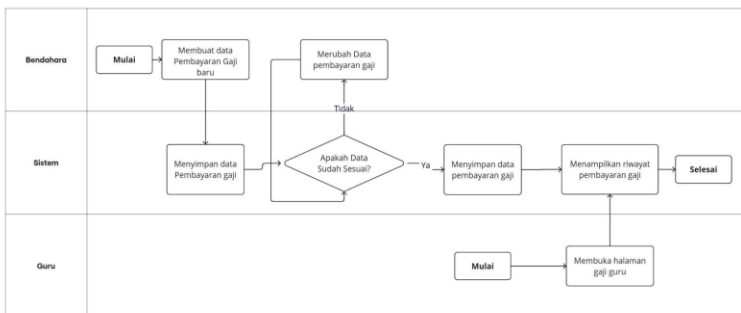
Gambar 4. 5 Activity diagram fitur PPDB

Gambar 4.6 merupakan *activity diagram* fitur pengeluaran rumah tangga sekolah.



Gambar 4. 6 Activity Diagram Fitur Rumah Tangga Sekolah

Gambar 4.7 merupakan *activity diagram* fitur gaji guru.

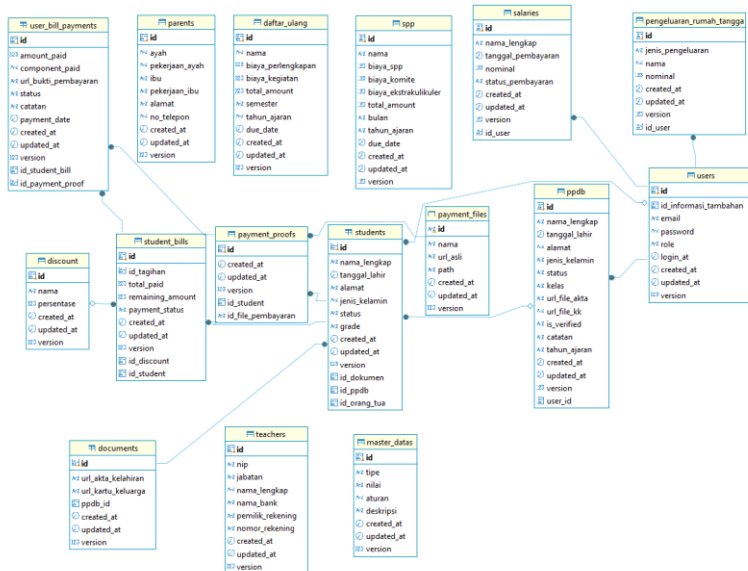


Gambar 4. 7 Activity Diagram Fitur Gaji Guru



## 4.2.3. Database Design

Berikut merupakan database design yang diimplementasikan ke sistem



Gambar 4. 8 Desain Antarmuka Halaman Pengeluaran Rumah Tangga Sekolah

Database ini dirancang untuk sistem manajemen data di sebuah institusi pendidikan, seperti sekolah. Secara keseluruhan, database ini memiliki beberapa tabel yang saling terhubung dengan tujuan mengelola berbagai aspek administrasi, termasuk pembayaran, data siswa, data orang tua, dokumen, serta informasi keuangan lainnya. Berikut adalah penjelasan umum mengenai struktur database yang terlihat pada diagram:

### 4.2.4.1. Tabel Utama

1. **users**: Tabel ini menyimpan data pengguna sistem, seperti informasi login, peran (role), dan tanggal pembuatan akun. Tabel ini berperan sebagai pusat autentikasi dan otorisasi.
2. **ppdb**: Menyimpan data pendaftaran siswa baru, termasuk nama lengkap, alamat, tanggal lahir, jenis kelamin, status verifikasi, dan tahun ajaran.

3. **students**: Berisi data siswa yang sudah terdaftar, meliputi nama lengkap, tanggal lahir, kelas, tingkat (grade), serta informasi orang tua.
4. **parents**: Menyimpan informasi tentang orang tua siswa, seperti pekerjaan ayah dan ibu, serta alamat rumah.

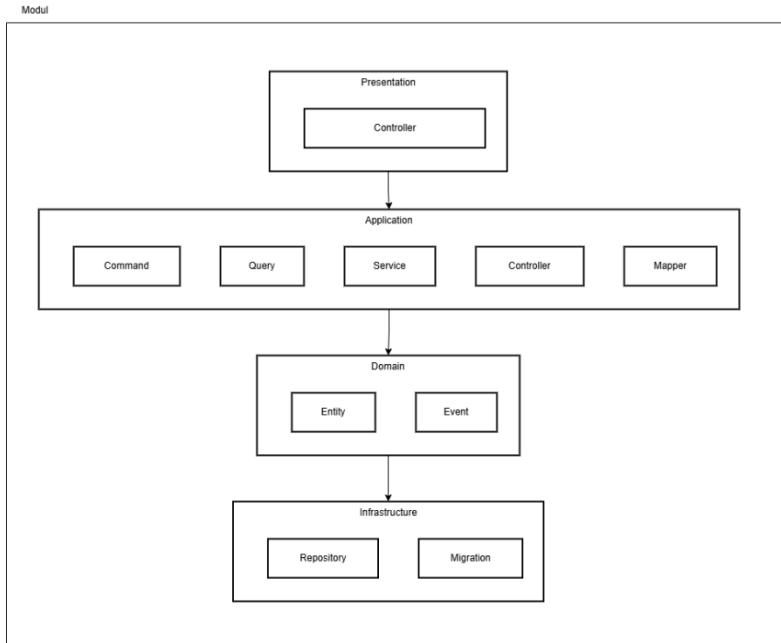
#### 4.2.4.2. Tabel Pembayaran dan Keuangan

1. **student\_bills**: Digunakan untuk menyimpan data tagihan siswa, seperti jumlah tagihan total, jumlah yang tersisa, status pembayaran, serta referensi ke diskon.
2. **user\_bill\_payments**: Mencatat transaksi pembayaran tagihan siswa, termasuk jumlah yang dibayar, bukti pembayaran, tanggal pembayaran, serta status transaksi.
3. **payment\_proofs**: Menyimpan bukti pembayaran, seperti file terkait pembayaran tertentu.
4. **payment\_files**: Menyediakan informasi mengenai file terkait pembayaran lainnya, seperti URL file asli.
5. **spp**: Digunakan untuk mencatat detail pembayaran SPP siswa, termasuk biaya komite, ekstrakurikuler, dan total nominal.
6. **discount**: Menyimpan informasi mengenai potongan harga yang diberikan, seperti persentase dan nama diskon.

Struktur ini menunjukkan perancangan yang terstruktur dengan fokus pada pengelolaan data yang terpusat dan terintegrasi, memastikan kemudahan akses dan akurasi data antarbagian dalam institusi.

#### 4.2.4. Arsitektur Backend

Arsitektur backend yang digunakan pada aplikasi Kinderfin adalah *Modular Clean Architecture* seperti pada Gambar x.x



Gambar 4. 9 Arsitektur Backemd

Pada arsitektur yang terdapat pada Gambar x.x dibagi menjadi beberapa lapisan yaitu *Presentation Layer*, *Application Layer*, *Domain Layer*, dan *Infrastructure Layer*. Berikut adalah masing-masing kegunaannya:

1. *Presentation Layer*: bertanggung jawab untuk menangani permintaan HTTP, memvalidasi data masukan, dan memanggil logika lapisan aplikasi yang sesuai untuk melakukan operasi yang diminta. Lapisan ini juga menangani pengecualian dan mengirimkan respons yang sesuai kembali ke klien.
2. *Application Layer*: bertanggung jawab sebagai jembatan antara *Presentation Layer* dan *Domain Layer*. Lapisan ini berisi logika bisnis utama dan mengoordinasikan interaksi antara berbagai komponen dalam sistem.
3. *Domain Layer*: Lapisan ini mendefinisikan model-model data domain dan juga event-event yang terjadi pada modul tersebut serta mendefinisikan *interface service* dan *repository* yang digunakan.

4. *Infrastructure Layer*: Lapisan ini bertanggung jawab dalam melakukan koneksi dan juga manajemen migrasi ke dalam database.



Gambar 4. 10 Komunikasi Antar Modul

Pada arsitektur tersebut interaksi antar modulnya mengandalkan *Event Emitter*, sehingga dalam melakukan aplikasi perlu mendefinisikan *event* terlebih dahulu sebagai identifier dalam melakukan komunikasi. Dalam komunikasi terdapat dua peran yaitu *subscriber* sebagai penerima data dan juga *publisher* sebagai pengirim datanya.

*[Halaman ini sengaja dikosongkan]*

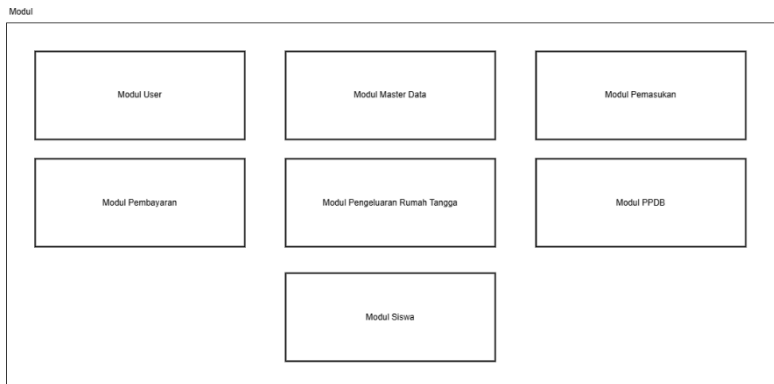
## **BAB V**

### **IMPLEMENTASI SISTEM**

Bab ini membahas tentang implementasi dari sistem yang kami buat. Implementasi ini akan dibagi ke dalam beberapa bagian sesuai fitur yang dikembangkan.

#### **5.1. Implementasi Modul**

Pada aplikasi ini dikembangkan beberapa modul seperti pada Gambar x.x



Gambar 5. 1Modul Backend yang dikembangkan

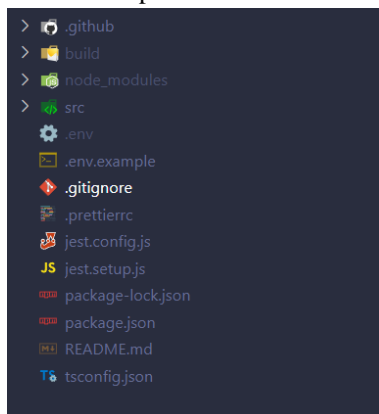
Berikut adalah fungsi dari modul-modul tersebut:

1. Modul User: Modul ini bertanggung jawab dalam manajemen user seperti autentikasi pengguna, delete akun pengguna, melihat profile pengguna, dan menambahkan pengguna baru.
2. Modul Master Data: Modul ini menangani data master atau data referensial yang digunakan di seluruh aplikasi.
3. Modul Pemasukan: Modul ini bertanggung jawab dalam melakukan pencatatan pemasukan keuangan dari sekolah terkait.
4. Modul Pembayaran: Modul ini bertanggung jawab dalam memproses segala bentuk pembayaran seperti pembayaran tagihan spp, komite, ekstrakurikuler, dan daftar ulang.

5. Modul Pengeluaran Rumah Tangga: Modul ini bertanggung jawab dalam melakukan pencatatan pengeluaran rumah tangga dari sekolah terkait.
6. Modul PPDB: Modul ini bertanggung jawab dalam melakukan proses pendaftaran peserta didik baru.
7. Modul Siswa: Modul ini berfungsi untuk melakukan manajemen siswa.

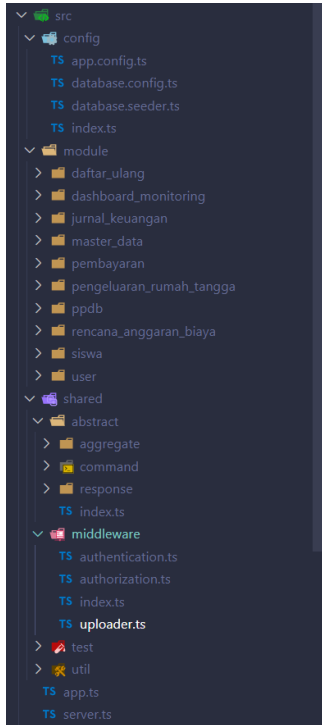
## 5.2. Struktur Kerangka Kode Sumber

Kode sumber menggunakan Express.js yang digunakan sebagai kerangka kerja pembuatan aplikasi *backend*. Berikut adalah gambaran mengenai struktur kode sumber aplikasi *backend* Kinderfin.



Gambar 5. 2 Struktur Folder Program

Pada struktur kerangka kerja dari kode sumber diisi oleh beberapa file konfigurasi dan folder utama dalam mengembangkan aplikasi *backend*. Untuk kode sumber dari aplikasi *backend* Kinderfin diletakkan pada folder *src*. Berikut adalah isi dari folder *src*.



Gambar 5. 3 Stuktur Seluruh Program

Berikut adalah penjelasan masing-masing folder yang ada pada folder *src*:

1. **config**: menyimpan dan mengelola pengaturan-pengaturan aplikasi agar dapat diakses dan digunakan secara konsisten di seluruh bagian aplikasi. Dengan memisahkan konfigurasi ke dalam file-file terpisah, aplikasi menjadi lebih mudah untuk dikonfigurasi, dikelola, dan dipelihara.
2. **module**: menyimpan modul-modul yang diperlukan dalam aplikasi.
3. **shared**: menyimpan komponen-komponen, utilitas, dan model-model umum yang dapat digunakan bersama di seluruh aplikasi.



### 5.3. API Implementation

Pada Aplikasi *Backend* Kinderfin terdapat beberapa API fungsional yang dikembangkan seperti yang ditunjukkan pada Tabel 5.1

Tabel 5. 1 Daftar API

No	Module	Endpoint	Metode	Deskripsi
1	Users	/api/v1/users/login	POST	Autentikasi user
		/api/v1/users/logout	POST	Keluar dari aplikasi
		/api/v1/users/guru	POST	Mendaftarkan pengguna baru dengan <i>role</i> sebagai <i>guru</i>
		/api/v1/users/orang-tua	POST	Mendaftarkan pengguna baru dengan <i>role</i> sebagai orang tua
		/api/v1/users	GET	Mendapatkan semua pengguna
2	Payment	/api/v1/payments/add-spp	POST	Menambahkan tagihan SPP
		/api/v1/payments/discount	POST	Menambahkan data <i>discount</i>
		/api/v1/payments/spp?student=	POST	Melakukan pembyaran SPP
		/api/v1/payments/orang_tua/bill	GET	Mendapatkan seluruh tagihan orang tua
		/api/v1/payments/orang_tua/history	GET	Mendapatkan history pembayaran orang tua
		/api/v1/payments/spp	GET	Mendapatkan seluruh data SPP
		/api/v1/payments/spp/history	GET	Mendapatkan seluruh data <i>history</i> pembayaran SPP

No	Module	Endpoint	Metode	Deskripsi
		/api/v1/payments/discount	GET	Mendapatkan seluruh data <i>discount</i> .
		/api/v1/payments/discount/{id}	PUT	Mengubah data <i>discount</i> berdasarkan id.
		/api/v1/payments/discount/{id}	DELETE	Menghapus data <i>discount</i> berdasarkan id.
		/api/v1/payments/spp/{id}	PUT	Mengubah data SPP berdasarkan id.
		/api/v1/payments/spp/{id}	DELETE	Menghapus data SPP berdasarkan id.
		/api/v1/payments/spp/user-bill/status/{id}	POST	Melakukan persetujuan atau penolakan pembayaran tagihan SPP
		/api/v1/payments/spp/student-bill/status/{id}	POST	Mengubah status tagihan SPP
		/api/v1/payments/spp/student-bills	GET	Mendapatkan semua data tagihan SPP
		/api/v1/payments/spp/history	DELETE	Menghapus seluruh data <i>history</i> pembayaran tagihan SPP
		/api/v1/payments/spp/history/{id}	DELETE	Menghapus data <i>history</i> pembayaran tagihan SPP berdasarkan id
3	Student	/api/v1/students/add	POST	Menambahkan data siswa
		/api/v1/students/all	GET	Mendapatkan seluruh data siswa
4	PPDB	/api/v1/ppdb	GET	Mendapatkan seluruh data PPDB

No	Module	Endpoint	Metode	Deskripsi
		/api/v1/ppdb/{id}	GET	Mendapatkan data PPDB berdasarkan
		/api/v1/ppdb/orang-tua	GET	Mendapatkan data PPDB berdasarkan orang tua
		/api/v1/ppdb/{id}	PUT	Mengedit data PPDB berdasarkan id
		/api/v1/ppdb/	DELETE	Menghapus seluruh data PPDB
		/api/v1/ppdb/verifikasi/	PUT	Melakukan verifikasi data PPDB
		/api/v1/master-datas/jenis-pembayaran	GET	Mendapatkan jenis pembayaran
6	Salaries	/api/v1/gaji	GET	Mendapatkan semua data gaji
		/api/v1/gaji/{id}	GET	Mendapatkan data gaji berdasarkan id
		/api/v1/gaji	POST	Menambahkan data gaji
		/api/v1/gaji/{id}	DELETE	Menghapus data gaji berdasarkan id
		/api/v1/gaji/{id}	PUT	Mengubah data gaji berdasarkan id
7	Pengeluaran Rumah Tangga	/api/v1/pengeluaran-rumah-tangga	GET	Mendapatkan semua data pengeluaran rumah tangga
		/api/v1/pengeluaran-rumah-tangga/{id}	GET	Mendapatkan data pengeluaran rumah tangga berdasarkan id

No	Module	Endpoint	Metode	Deskripsi
		/api/v1/pengeluaran-rumah-tangga/add	POST	Menambahkan data pengeluaran rumah tangga berdasarkan id
		/api/v1/pengeluaran-rumah-tangga/{id}	PUT	Mengubah data pengeluaran rumah tangga berdasarkan id
No	Module	Endpoint	Metode	Deskripsi
		/api/v1/pengeluaran-rumah-tangga/{id}	DELETE	Menghapus data pengeluaran rumah tangga berdasarkan id
8	Master data	/api/v1/master-datas/Bank	POST	Menambahkan master data Bank
		/api/v1/master-datas/Bank	GET	Mendapatkan master data Bank
		/api/v1/master-datas/Bank/{id}	PUT	Mengedit master data Bank berdasarkan id
		/api/v1/master-datas/Bank/{id}	DELETE	Menghapus master data Bank berdasarkan id
7	Pemasukan	/api/v1/pemasukan	GET	Mendapatkan semua data pemasukan
		/api/v1/pemasukan/{id}	GET	Mendapatkan data pemasukan berdasarkan id
		/api/v1/pemasukan/add	POST	Menambahkan data pemasukan berdasarkan id
		/api/v1/pemasukan/{id}	PUT	Mengubah data pemasukan berdasarkan id
		/api/v1/pemasukan/{id}	DELETE	Menghapus data pemasukan berdasarkan id

Terdapat sekitar delapan modul dengan 50 endpoint fungsional yang dikembangkan dan telah diintegrasikan dengan *frontend* website Kinderfin.

## **BAB VI**

### **PENGUJIAN DAN EVALUASI**

Bab ini menjelaskan tahap uji coba terhadap Aplikasi Kinderfin. Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi arsitektur dengan analisis dan perancangan arsitektur.

#### **6.1. Tujuan Pengujian**

Pengujian API *testing* dilakukan terhadap aplikasi *Backend* Kinderfin memiliki beberapa tujuan utama yaitu sebagai berikut:

- a. Memastikan bahwa *endpoint* yang ada pada aplikasi *backend* benar-benar berfungsi dengan baik.
- b. Mengidentifikasi performa aplikasi *backend* dan kesalahan yang mungkin terjadi.

#### **6.2. Kriteria Pengujian**

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut :

1. Kemampuan API untuk mengakses fitur sesuai otorisasi pengguna yang *login*.
2. Kemampuan API untuk dapat mengakses *endpoint* sesuai dengan *role* yang dimiliki pengguna.
3. Kemampuan API untuk menerima input dari pengguna sesuai jenis input (teks, angka, password, dan *selection*) dan melakukan validasi input.
4. Kemampuan API untuk menampilkan data dari database secara akurat.
5. Kemampuan API untuk membuat data baru.
6. Kemampuan API untuk memperbarui data.
7. Kemampuan API untuk menghapus data.
8. Kemampuan API untuk menerima dan mengunggah *file* dari pengguna dengan tipe *file* yang sesuai.

#### **6.3. Skenario Pengujian**

Skenario pengujian dilakukan dengan membuat HTTP *Request* dengan menggunakan postman dengan input yang disesuaikan

berdasarkan *endpoint*-nya dan hasil yang diharapkan harus sesuai dengan yang didefinisikan di program. Pada Tabel 6.1 merupakan contoh skenario pengujian yang diterapkan pada salah satu fitur yaitu menambahkan data orang tua.

Tabel 6. 1 *Contoh Skenario Pengujian Postman*

No	Skenario	Contoh Masukan	Hasil
1	Email telah terdaftar	{ "ayah": "Andika", "ibu": "Istri Andika", "pekerjaan_ayah": "Karyawan BUMN", "pekerjaan_ibu": "Ibu Rumah Tangga", "alamat": "Surabaya", "no_telepon": "0831223013912", "email": "andika24@gmail.com", "password": "Andikalaksana1!", "role": "Orang Tua" }	{ "status": "failed", "error": "Email telah terdaftar" }
2	Email belum terdaftar	{ "ayah": "Andika", "ibu": "Istri Andika", "pekerjaan_ayah": "Karyawan BUMN", "pekerjaan_ibu": "Ibu Rumah Tangga", "alamat": "Surabaya", "no_telepon": "0831223013912", "email": "andika20@gmail.com", "password": "Andikalaksana1!", "role": "Orang Tua" }	{ "status": "success", "message": "Berhasil menambahkan data baru" }

No	Skenario	Contoh Masukan	Hasil
3	Input email kosong	{ "ayah": "Andika", "ibu": "Istri Andika", "pekerjaan_ayah": "Karyawan BUMN", "pekerjaan_ibu": "Ibu Rumah Tangga", "alamat": "Surabaya", "no_telepon": "0831223013912", "password": "Andikalaksana1!", "role": "Orang Tua" }	{ "status": "failed", "error": "Input email diperlukan" }

Untuk tiap-tiap *endpoint* akan dilakukan pengujian yang sama yaitu menguji apabila masukan yang wajib tapi dikosongkan, apabila masukan data dengan *unique key* sudah ada di database, dan juga apabila tidak ada kesalahan sama sekali pada data.

#### 6.4. Evaluasi Pengujian

Hasil pengujian dilakukan dengan mengamati perilaku aplikasi *backend* Kinderfin berdasarkan berbagai skenario uji coba yang telah dirancang. Pengujian ini bertujuan untuk memastikan bahwa aplikasi *backend* Kinderfin berfungsi sesuai dengan spesifikasi yang diharapkan dan memenuhi kebutuhan pengguna. Tabel 6.2 di bawah ini menjelaskan hasil uji coba terhadap aplikasi yang telah dikembangkan.

Tabel 6. 2 Evaluasi Pengujian

No.	Kriteria Pengujian	Hasil Pengujian	Status
1.	Kemampuan API untuk mengakses fitur sesuai otorisasi pengguna yang <i>login</i>	Aplikasi berhasil mengimplementasikan middleware otorisasi berdasarkan <i>role</i> pengguna yang <i>login</i> .	Sesuai



No.	Kriteria Pengujian	Hasil Pengujian	Status
2.	Kemampuan API untuk dapat mengakses <i>endpoint</i> sesuai dengan <i>role</i> yang dimiliki pengguna	Pengguna dapat menggunakan <i>endpoint</i> yang sesuai dengan <i>role</i> yang telah ditentukan.	Sesuai
3.	Kemampuan API untuk menerima input sesuai jenis (teks, angka, dll.).	API dapat menerima input sesuai tipe dan memberikan pesan error jika salah.	Sesuai
4.	Kemampuan API untuk menampilkan data dari database secara akurat.	Data dari database ditampilkan dengan akurat.	Sesuai
5.	Kemampuan API untuk membuat data baru.	Data baru berhasil dibuat dan ditampilkan.	Sesuai
6.	Kemampuan API untuk memperbarui data.	Data berhasil diperbarui.	Sesuai
7.	Kemampuan API untuk menghapus data.	Data berhasil dihapus.	Sesuai
8.	Kemampuan API untuk menerima dan mengunggah file dengan tipe yang sesuai.	File terunggah sesuai tipe file yang diterima dan tersimpan di server.	Sesuai

Berdasarkan hasil pengujian di atas dapat disimpulkan bahwa aplikasi Kinderfin telah memenuhi seluruh kriteria pengujian yang direncanakan dan setiap fitur utama dalam aplikasi berfungsi sesuai dengan harapan sehingga memastikan pengguna dapat berinteraksi dengan sistem secara optimal dan tanpa kendala berarti.

## **BAB VII**

### **KESIMPULAN DAN SARAN**

#### **7.1. Kesimpulan**

Kesimpulan yang didapat setelah melakukan perancangan dan pengembangan sistem aplikasi Kinderfin pada kegiatan kerja praktik ini adalah sebagai berikut:

- a. Aplikasi Kinderfin berhasil dikembangkan sesuai dengan spesifikasi dan kebutuhan yang telah dirancang untuk mendukung pengelolaan keuangan di kelompok belajar dan taman kanak-kanak.
- b. Implementasi sistem berbasis otorisasi pada aplikasi Kinderfin memungkinkan pengguna mengakses fitur sesuai peran masing-masing serta meningkatkan keamanan dan kenyamanan dalam penggunaan aplikasi.
- c. Hasil pengujian menunjukkan bahwa aplikasi Kinderfin telah memenuhi seluruh kriteria pengujian dan berfungsi sesuai harapan sehingga operasional aplikasi berjalan dengan lancar tanpa kendala berarti bagi pengguna.
- d. Aplikasi Kinderfin menjadikan pengelolaan keuangan di kelompok belajar dan taman kanak-kanak lebih terstruktur, efisien, dan transparan serta meningkatkan akurasi dan kemudahan akses data bagi pengguna.

#### **7.2. Saran**

Saran untuk perancangan sistem aplikasi Kinderfin adalah sebagai berikut :

- a. Dalam mengembangkan API backend sebaiknya dibuatkan *unit testing* terlebih dahulu untuk melakukan pengujian terhadap fungsionalitas dari aplikasi yang dibuat.
- b. Dapat menggunakan beberapa *tools* untuk melakukan pengujian terhadap performa aplikasi backend dengan menggunakan JMeter atau Apache Benchmark.

*[Halaman ini sengaja dikosongkan]*

# LAMPIRAN

## 1. Dokumentasi Postman Modul User

### 1. Login

Kinderrline API / Local / Users / Login / [Success] Orang Tua Save Share ...

**POST** `((url))/users/login` Try

Params Headers **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON** Beauty

```
1 {
2   "email": "bud@gmail.com",
3   "password": "OrangtuaPassi1"
4 }
5
6 // |
7 // "email": "rootuser@gmail.com",
8 // "password": "Rootpassi1"
9 // }
```

Body Headers (20) Status Code 200 OK

Pretty Raw Preview **JSON** Close Search

```
1 {
2   "status": "success",
3   "message": "User berhasil login",
4   "data": {
5     "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IHYwNCIsIm51IjoiIn0",
6     "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IHYwNCIsIm51IjoiIn0",
7     "role": "Orang Tua",
8     "username": "budi"
9   }
}
```

### 2. Add Teacher

Kinderrline API / Local / Users / Add Teachers / **Add Teachers** Save

**POST** `http://localhost:9001/api/v1/users/guru`

Params Headers **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON** Close Search

```
1 {
2   "nip": "010203040506070809",
3   "jabatan": "Sekretaris",
4   "nama_lengkap": "Dimas Fadilah Akbar",
5   "email": "dimas@gmail.com",
6   "password": "SekretarisPassi1",
7   "role": "Sekretaris",
8   "nama_bank": "BNI",
9   "pemilik_rekening": "Dimas Fadilah Akbar",
10  "nomor_rekening": "123456789"
}
```

Body Headers (19) Status Code 201 Create

Pretty Raw Preview **JSON** Close Search

```
1 {
2   "status": "success",
3   "message": "Berhasil menambahkan data baru"
}
```

### 3. Add Parents

Kinderfine API / Local / Users / Add Parents / [Success]

Save

POST `{{dev_url}}/api/v1/users/orang-tua`

Params Headers Body

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL  JSON

```
12 {
13   {
14     "ayah": "budiman",
15     "ibu": "sitiwoMan",
16     "pekerjaan_ayah": "pns",
17     "pekerjaan_ibu": "pns",
18     "alamat": "surabaya",
19     "no_telepon": "081223013912",
20     "email": "budiman@gmail.com",
21     "password": "OrangtuaPass12",
22     "role": "Orang Tua"
23   }
}
```

Body Headers (19)

Status Code 201 Created

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil menambahkan data baru"
4 }
```

### 4. Get All Users

Kinderfine API / Local / Users / getAllUsers / [Success]

Save

GET `{{dev_url}}/api/v1/users`

Params Headers Body

Query Params

Key	Value	Description
-----	-------	-------------

Body Headers (19)

Status Code 200 OK

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan semua data",
4   "data": [
5     {
6       "id": "4874cb6e-ab8f-4de1-9ec8-c618b8b9628",
7       "id_informasi_tambahan": "272d4fe1-64a8-48cb-a1d9-6e96960e48e",
8       "email": "bud1@gmail.com",
9       "role": "Orang Tua",
10      "login_at": "2024-10-05T11:21:08.427Z"
11    },
12    {
13      "id": "eccbb79f-3e54-43d7-b52d-c2534d645e28",
14      "id_informasi_tambahan": "ebc9a12f-2280-48bb-9d76-0de8c1bf59be",
15      "email": "budiman@gmail.com",
16      "role": "Orang Tua",
17      "login_at": null
18    }
19  ]
}
```

## 2. Payment SPP

### 1. Menambahkan Tagihan SPP

Kinderline API / Local / Payments SPP / Add Bill SPP (Bendahara) / [Dev] Success

Save

POST `{{dev_url}}/payments/add-spp`

Params Headers **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON** ▾

```
1 {
2   "nama": "SPP Bulan Agustus 2024",
3   "biaya_spp": "200000",
4   "biaya_komite": "20000",
5   "biaya_ekstrakurikuler": "20000",
6   "bulan": "Februari",
7   "tahun_ajaran": "2025",
8   "due_date": "2025-09-13"
9 }
```

Body Headers (25) Status Code 201 Create

Pretty Raw Preview **JSON** ▾

```
1 {
2   "status": "success",
3   "message": "Berhasil menambahkan data baru"
4 }
```

### 2. Menambahkan Diskon

Kinderline API / Local / Payments SPP / Add Discount (Bendahara) / [Dev] Success

Save

POST `{{dev_url}}/payments/discount`

Params Headers **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON** ▾

```
1 {
2   "nama": "Yatin",
3   "persentase": 10
4 }
```

Body Headers (25) Status Code 201 Create

Pretty Raw Preview **JSON** ▾

```
1 {
2   "status": "success",
3   "message": "Berhasil menambahkan data baru"
4 }
```

### 3. Membayar Tagihan SPP oleh Orang Tua

Kinderfine API / Local / Payments SPP / Settle SPP Bill (Orang Tua) / [DEV] Success

Save

POST `[[dev_url]]/payments/spp?student=73ca8e5a-aa5a-4947-adf2-c980f9e53e18&spp=a4a245b8-8f74-4745-bc02-dac6cfce86f`

Params Headers Body

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

Key	Value	Description
<input checked="" type="checkbox"/> amount_paid	Text 10000	
<input checked="" type="checkbox"/> component_paid	Text SPP	
<input checked="" type="checkbox"/> bukti_pembayaran	File <code>FC7C06BB-0CDB-4C6B-97A7-7A5BAOC...</code>	
Key	Value	Description

Body Headers (25)

Status Code 201 Create

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil menambahkan data baru"
4 }
```

### 4. Mendapatkan Tagihan SPP Orang Tua

Kinderfine API / Local / Payments SPP / Get SPP Bill (Orang Tua) / [New Success]

Save

GET `[[url]]/payments/orang_tua/bill`

Params Headers (1) Body

Query Params

Key	Value	Description
Key	Value	Description

Body Headers (19)

Status Code 200 OK

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan semua data",
4   "data": [
5     {
6       "id_bill": "99c76588-9fae-4ce8-b9af-8e7e40252e1c",
7       "id_student": "2ae83853-4257-43f2-93d8-c1683bdb2a9e",
8       "id_tagihan": "5a905d02-aecc-4a62-93e4-4f99f89d172",
9       "id_discount": "67fce9b4-e097-4f11-acd7-bb6808da601",
10      "total_paid": 0,
11      "remaining_amount": 0,
12      "payment_status": "BELUM LUNAS",
13      "nama_siswa": "Andika Laksana",
14      "nama_tagihan": "SPP Bulan Juli 2024",
15      "biaya_spp": 200000,
16      "biaya_komite": 20000,
17      "biaya_ekstrakurikuler": 20000,
18      "due_date": "2025-09-13T08:00:00.000Z"
19    }
20  ],
21 }
```

## 5. Mendapatkan Riwayat Pembayaran Tagihan SPP Orang Tua

Kinderline API / Local / Payments SPP / Get History Payment (Orang Tua) / [Dev] Success Save Share

GET `{{dev_url}}/payments/orang_tua/history` Try

Params Headers (1) Body

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Headers (25) Status Code 200 OK

Pretty Raw Preview JSON Copy

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan semua data",
4   "data": [
5     {
6       "id_user_bill_payment": "2e3123d8-bb4e-4baa-9cec-3e6d71a066e9",
7       "amount_paid": 12000,
8       "component_paid": "SPP",
9       "url_bukti_pembayaran": "https://ik.imagekit.io/2hjvbo17c/0412390d-3738-41ac-9283-fac4802d7df_scheme-ci-cd_draio_34beYn6Z8.png",
10      "payment_date": "2024-10-06T14:16:00.208Z",
11      "id_student_bill": "3bd17868-2ff1-4763-8d3e-0e82bcfae2d2",
12      "id_payment_proof": "8e6d24e6d-0346-4366-9d92-63e6f6e59b27",
13      "nama_tagihan": "SPP Bulan Januari 2025",
14      "nama_siswa": "Fathin Muhasbi",
15      "id_orang_tua": "d9fa0876d-fe9b-4cf9-8033-38a0d1426528",
16      "orang_tua": "ayah"
17    }
18  ]
19 }
```

## 6. Mendapatkan Semua Tagihan SPP Bendahara

Kinderline API / Local / Payments SPP / Get All SPP Bill (Bendahara, Sekretaris) / [Dev] Success Save

GET `{{dev_url}}/payments/spp`

Params Headers (1) Body

Query Params

Key	Value	Description
Key	Value	Description

Body Headers (25) Status Code 200 OK

Pretty Raw Preview JSON Copy

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan data",
4   "data": [
5     {
6       "id": "63fe7c09-e8ae-46c0-b97b-3ab5c72b2977",
7       "nama": "SPP Bulan Agustus 2026",
8       "biaya_spp": 100000,
9       "biaya_komite": 50000,
10      "biaya_ekstrakurikuler": 40000,
11      "total_amount": 190000,
12      "bulan": "Agustus",
13      "tahun_ajaran": "2026",
14      "due_date": "2026-08-31T00:00:00.000Z"
15    }
16  ]
17 }
```



## 7. Mendapatkan Semua Riwayat Pembayaran

Kinderrfine API / Local / Payments SPP / Get All SPP Bill (Bendahara, Sekretaris) / [Dev] Success

Save

GET

Params Headers (1) Body

Query Params

Key	Value	Description
Key	Value	Description

Body Headers (25)

Status Code 200 OK

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan data",
4   "data": [
5     {
6       "id": "63fe7c99-e8ae-46c9-b978-3ab5c72b2977",
7       "nama": "SPP Bulan Agustus 2026",
8       "biaya_spp": 100000,
9       "biaya_komite": 50000,
10      "biaya_ekstrakurikuler": 40000,
11      "total_amount": 190000,
12      "bulan": "Agustus",
13      "tahun_ajaran": "2026",
14      "due_date": "2026-08-31T00:00:00.000Z"
15    }
16  ],
17 }
```

## 8. Mendapatkan Semua Diskon

Kinderrfine API / Local / Payments SPP / Get All Discount (Bendahara) / [Dev] Success

Save

GET

Params Headers (1) Body

Query Params

Key	Value	Description
Key	Value	Description

Body Headers (25)

Status Code 200 OK

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan data",
4   "data": [
5     {
6       "id": "b4d737af-d813-4b87-bd38-03bb069188e",
7       "nama": "Yatin",
8       "persentase": 10
9     },
10    {
11      "id": "5fe6387d-3196-4310-897d-d3882faa29c1",
12      "nama": "Dhuafa",
13      "persentase": 10
14    }
15  ],
16 }
```

## 9. Mengubah Data Diskon

Kinderfine API / Local / Payments SPP / Edit Discount (Bendahara) / [Dev] Success

**PUT** ▼ `{{dev_url}}/payments/discount/5fe6387d-3196-4310-897d-d3882faa29c1`

Params Headers **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON** ▼

```
1 {
2   "nama": "Dhuafa",
3   "peisentase": 15
4 }
```

**Body** Headers (25) Statu

Pretty Raw Preview **JSON** ▼

```
1 {
2   "status": "success",
3   "message": "Berhasil memperbarui data"
4 }
```

## 10. Menghapus Data Diskon Berdasarkan ID

Kinderfine API / Local / Payments SPP / Delete Discount (Bendahara) / [Dev] Success

**DELETE** ▼ `{{dev_url}}/payments/discount/39b7aca3-e715-4287-9e4d-e4f21fc3f3f1`

Params Headers **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

This request does not have a body

**Body** Headers (25)

Pretty Raw Preview **JSON** ▼

```
1 {
2   "status": "success",
3   "message": "Berhasil menghapus data"
4 }
```

## 11. Mengubah Data SPP

Kinderfine API / Local / Payments SPP / Edit SPP (Bendahara) / [Dev] Success

PUT

Params Headers (1) **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON**

```
23 {
24   "nama": "SPP Bulan Mei 2025",
25   "biaya_spp": 200000,
26   "biaya_komite": 50000,
27   "biaya_ekstrakurikuler": 40000,
28   "bulan": "Januari",
29   "tahun_ajaran": "2025",
30   "due_date": "2025-04-30T00:00:00.000Z"
31 }
```

**Body** Headers (25)

Pretty Raw Preview **JSON**

```
1 {
2   "status": "success",
3   "message": "Berhasil memperbarui data"
4 }
```

## 12. Menghapus Data SPP

Kinderfine API / Local / Payments SPP / Delete SPP (Bendahara) / [Dev] Success [@](#)

DELETE

Params Headers (1) **Body**

Query Params

Key	Value	Description
Key	Value	Description

**Body** Headers (25)

Pretty Raw Preview **JSON**

```
1 {
2   "status": "success",
3   "message": "Berhasil menghapus data"
4 }
```

### 13. Mengubah Status Pembayaran

Kinderfine API / Local / Payments SPP / User Bill Payment Set Status / [DEV] Success


POST ⌵ | {{{dev\_url}}}/payments/spp/user-bill/status/d35f39d9-c127-4c05-877e-972ce59d5bfa

Params   Headers (1)   **Body** ●

none    form-data    x-www-form-urlencoded    raw    binary    GraphQL   JSON ⌵

```
1  {
2    "status": "REJECTED",
3    "catatan": "Gambar tidak sesuai, tolong kirim ulang"
4  }
```

Body   Headers (25)

Pretty   Raw   Preview   JSON ⌵   

```
1  {
2    "status": "success",
3    "message": "Berhasil memperbarui data"
4  }
```

## 14. Mendapatkan Semua Tagihan SPP

Kinderfine API / Local / Payments SPP / Get All Student Bill (Bendahara, Sekretaris) / [Success]

GET `{{dev_url}}/payments/spp/student-bills`

Params Headers Body

Query Params

Key	Value	Description
Key	Value	Description

Body Headers (19) Status Code

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan semua data",
4   "data": [
5     {
6       "id_bill": "99c76588-9fae-4ce0-b9af-8e7e40252e1c",
7       "id_student": "2ae83653-4257-43f2-93d8-c1603bdb2a9e",
8       "id_tagihan": "5a9d5dd2-aecc-4a62-93e4-4f99f89dd172",
9       "id_discount": "67fce9b4-e097-4f11-acd7-bb6b808da601",
10      "total_paid": 0,
11      "remaining_amount": 0,
12      "payment_status": "BELUM LUNAS",
13      "nama_siswa": "Andika Laksana",
14      "nama_tagihan": "SPP Bulan Juli 2024",
15      "biaya_spp": 200000,
16      "biaya_komite": 20000,
17      "biaya_ekstrakurikuler": 20000,
18      "due_date": "2025-09-13T00:00:00.000Z"
19    }
20  ],
21 }
```

## 15. Menghapus Semua Riwayat Pembayaran SPP

DELETE `{{dev_url}}/payments/spp/history`

Params Headers Body

Query Params

Key	Value
Key	Value

Body Headers (19)

Pretty Raw Preview JSON

```
1 {}
2   "status": "success",
3   "message": "Berhasil menghapus data"
4 }
```

## 16. Menghapus Data Riwayat Pembayaran SPP Berdasarkan ID

Kinderfine API / Local / Payments SPP / Delete History Payment by Id (Admin) / [Dev] Success


**DELETE** ▼ `{{dev_url}}/payments/spp/history/afa04dc6-6bef-425f-8bca-93f865d4571f`

Params Headers (1) Body

Query Params

Key	Value	De
Key	Value	De

Body Headers (26)

Pretty Raw Preview JSON ▼ 

```
1
2
3
4
{"status": "success",
 "message": "Berhasil menghapus data"}
```





### 3. Dokumentasi Postman Modul User

#### 1. Menambahkan Data Siswa


POST ▼ | `{{dev_uri}}/students/add`

Params Headers **Body** ●

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

Key		Value
<input checked="" type="checkbox"/> nama_lengkap	Text <span>▼</span>	Fathin Muhashibi
<input checked="" type="checkbox"/> tanggal_lahir	Text <span>▼</span>	2012-08-10
<input checked="" type="checkbox"/> alamat	Text <span>▼</span>	Surabaya
<input checked="" type="checkbox"/> jenis_kelamin	Text <span>▼</span>	Laki-laki
<input checked="" type="checkbox"/> akta_kelahiran	File <span>▼</span>	 _MG_0630.JPG 
<input checked="" type="checkbox"/> kartu_keluarga	File <span>▼</span>	 _MG_0650.JPG 
Key	Text <span>▼</span>	Value

Body Headers (25)

Pretty Raw Preview JSON ▼ 

```
1 {
2   "status": "success",
3   "message": "Berhasil menambahkan data baru"
4 }
```

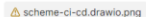

## 4. Dokumentasi Postman Modul PPDB

### 1. Mendaftarkan PPDB


POST `{{url}}/ppdb/add`

Params Headers **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

Key	Value	Description
<input checked="" type="checkbox"/> nama_lengkap	Text andika akmal	
<input checked="" type="checkbox"/> tanggal_lahir	Text 2021-08-11	
<input checked="" type="checkbox"/> alamat	Text surabaya	
<input checked="" type="checkbox"/> jenis_kelamin	Text Perempuan	
<input checked="" type="checkbox"/> akta_kelahiran	File 	
<input checked="" type="checkbox"/> kartu_keluarga	File 	
<input checked="" type="checkbox"/> status	Text Yatim	
<input checked="" type="checkbox"/> kelas	Text bawah	

Body Headers (19) Status Code 201 Creat



Pretty Raw Preview JSON 


```

1 {
2   "status": "success",
3   "message": "Berhasil menambahkan data baru"
4 }

```

### 2. Mendapatkan Semua Data PPDB


Kindertine API / Local / PPDB / Get All PPDB Data / **Success**  


GET `{{dev:url}}/ppdb` 

Params Headers Body

Query Params

Key	Value	Description	Bulk
Key	Value	Description	

Body Headers (19) Status Code 200 OK 

Pretty Raw Preview JSON 



```

1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan data",
4   "data": [
5     {
6       "id": "df6022f5-f506-4eb0-8fc3-0b2ff60bb26e",
7       "nama_lengkap": "malawa",
8       "tanggal_lahir": "2012-08-10T00:00:00.000Z",
9       "alamat": "jember",
10      "jenis_kelamin": "Perempuan",
11      "status": "Yatim",
12      "kelas": "tk",
13      "url_file_akta": "https://ik_imagekit.io/2hjvbo17c/7938be9-cb39-469b-ab99-e574a6706e17_scheme-ci-cd.drawio_cr-Pe1yH4f.png",
14      "url_file_kk": "https://ik_imagekit.io/2hjvbo17c/lv7c237fd-5a41-483c-86e5-390d083a3575_scheme-ci-cd.drawio_mCgBm8-6.png",
15      "user_id": "4074c6e-ab0f-4de1-9ecb-c610b8bd9620"
16     }
17   ]
18 }

```



### 3. Mendapatkan Data PPDB Berdasarkan ID

Kinderline API / Local / PPDB / Get PPDB Data By ID / [Success]  


GET

Params Headers Body

Query Params



Key	Value	Description	...
Key	Value	Description	

Body Headers (19) Status Code 200 OK

Pretty Raw Preview JSON 

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan data",
4   "data": {
5     "id": "970d822c-9d2b-407b-b65f-1d8995f3cd82",
6     "nama_lengkap": "andika akmal",
7     "tanggal_lahir": "2021-08-11T08:00:00.000Z",
8     "alamat": "surabaya",
9     "jenis_kelamin": "Perempuan",
10    "status": "Yatin",
11    "kelas": "bawah",
12    "url_file_akta": "https://ik.imagekit.io/2h1vbo17c/2fc1846b-1bff-47f8-b677-255542af56d7_scheme-ci-cd_drawio_KhxpWUed0.png",
13    "url_file_kk": "https://ik.imagekit.io/2h1vbo17c/617752f9-4mec-4333-b666-58daac8d70e3_scheme-ci-cd_drawio_FdzG4C9gT.png",
14    "user_id": "4074cb6e-ab8f-4de1-9ec8-c610b8bd9620"
15  }
16 }
```

### 4. Mendapatkan Data PPDB Berdasarkan ID Orang Tua

Kinderline API / Local / PPDB / Get PPDB Data By User ID / orang tua / [Success]  


GET

Params Headers Body

Query Params

Key	Value	Description	...
Key	Value	Description	

Body Headers (19) Status Code 200 OK

Pretty Raw Preview JSON 

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan data",
4   "data": [
5     {
6       "id": "df6022f5-f506-4eb0-8fc3-8b2ff68bb26e",
7       "nama_lengkap": "ualawe",
8       "tanggal_lahir": "2012-08-10T08:00:00.000Z",
9       "alamat": "jenbei",
10      "jenis_kelamin": "Perempuan",
11      "status": "Yatin",
12      "kelas": "tk",
13      "url_file_akta": "https://ik.imagekit.io/2h1vbo17c/7938be09-cb38-469b-ab99-e574a6786e17_scheme-ci-cd_drawio_CR-Pe1yH4f.png",
14      "url_file_kk": "https://ik.imagekit.io/2h1vbo17c/b7c237fb-5a41-403c-86e5-390d883a3575_scheme-ci-cd_drawio_mMc8M0-g.png",
15      "user_id": "4074cb6e-ab8f-4de1-9ec8-c610b8bd9620"
16     }
17   ],
18 }
```

## 5. Mengubah Data PPDB Berdasarkan ID

Kinderfine API / Local / PPDB / Update / [Success]

**PUT**  `{{url}}` /ppdb/970d822c-9d2b-40f7-b65f-1d8995f3cd82

Params Headers **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

Key	Value	De
<input checked="" type="checkbox"/> nama_lengkap	Text Updated nama	
Key	Text Value	De

Body Headers (19)

Pretty Raw Preview JSON

```
1
2 "status": "success",
3 "message": "Berhasil memperbarui data"
4
```

## 6. Menghapus Data PPDB Berdasarkan ID

Kinderfine API / Local / PPDB / Delete PPDB data By ID / [Success]

**DELETE**  `{{url}}` /ppdb/df6022f5-f506-4eb0-8fc3-8b2ff68bb26e

Params Headers Body

Query Params

Key	Value
Key	Value

Body Headers (19)

Pretty Raw Preview JSON

```
1
2 "status": "success",
3 "message": "Berhasil menghapus data"
4
```

## 5. Dokumentasi Postman Modul Gaji

### 1. Mendapatkan Semua Data Gaji

GET `/(dev_url)/gaji`

Params Headers Body

Query Params

Key	Value	Description
Key	Value	Description

Body Headers (19)

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan semua data",
4   "data": [
5     {
6       "id": "9eba52fb-2be3-4983-86b1-f71acd460565",
7       "nama_lengkap": "John Doe",
8       "tanggal_pembayaran": "2024-08-21T05:00:00.000Z",
9       "nominal": 1500000,
10      "status_pembayaran": "Lunas",
11      "id_user": "5a53d571-f85b-4373-8935-bc7eefab74f6"
12    },
13    {
14      "id": "43751337-a51d-4a9d-8055-8d4025c7eb0d",
15      "nama_lengkap": "John Doe",
16      "tanggal_pembayaran": "2024-08-21T05:00:00.000Z",
17      "nominal": 1500000,
18      "status_pembayaran": "Lunas",
19      "id_user": "4e5508fd-979d-47ad-a56b-e9a604d02f1f"
20    }
21  ],
22 }
```

### 2. Mendapatkan Data Gaji Berdasarkan ID

GET `/(dev_url)/gaji?user_id=5a53d571-f85b-4373-8935-bc7eefab74f6`

Params Headers Body

Query Params

Key	Value	Description
<input checked="" type="checkbox"/> Key	Value	Description
<input checked="" type="checkbox"/> user_id	5a53d571-f85b-4373-8935-bc7eefab74f6	
Key	Value	Description

Body Headers (19) Status Code 200

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan semua data",
4   "data": [
5     {
6       "id": "9eba52fb-2be3-4983-86b1-f71acd460565",
7       "nama_lengkap": "John Doe",
8       "tanggal_pembayaran": "2024-08-21T05:00:00.000Z",
9       "nominal": 1500000,
10      "status_pembayaran": "Lunas",
11      "id_user": "5a53d571-f85b-4373-8935-bc7eefab74f6"
12    }
13  ],
14 }
```

## 6. Dokumentasi Postman Modul Pengeluaran Rumah Tangga

### 1. Mendapatkan Semua Pengeluaran Rumah Tangga

GET `{{dev_url}}/pengeluaran-rumah-tangga`

Params Headers Body

Query Params

Key	Value	Description
-----	-------	-------------

Body Headers (19) Str

Pretty Raw Preview JSON `⌵` `⌵`

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan semua data",
4   "data": [
5     {
6       "id": "b1133150-9707-4eb6-bba2-ed2286d2a076",
7       "jenis_pengeluaran": "Transportasi",
8       "nama": "Ongkos Bus",
9       "nominal": 15000,
10      "id_user": "5a53d571-f85b-4373-8935-bc7eefab74fe"
11    },
12    {
13      "id": "a253b043-a15f-4f34-b0c0-06480c990ec5",
14      "jenis_pengeluaran": "Transportasi",
15      "nama": "Ongkos Bus",
16      "nominal": 15000,
17      "id_user": "5a53d571-f85b-4373-8935-bc7eefab74fe"
18    },
19    {
20      "id": "c42333c4-b08e-47fe-8687-437526de5815",
21      "jenis_pengeluaran": "Transportasi",
22      "nama": "Ongkos Bus",
23      "nominal": 15000,
24      "id_user": "5a53d571-f85b-4373-8935-bc7eefab74fe"
25    }
26  ]
27 }
```

### 2. Mendapatkan Pengeluaran Rumah Tangga Berdasarkan User ID

Kinderfine API / Local / Pengeluaran Rumah Tangga / getAllPengeluaranRTbyId / **getAllPengeluaranRTbyId**

GET `{{dev_url}}/pengeluaran-rumah-tangga?user_id=5a53d571-f85b-4373-8935-bc7eefab74fe`

Params Headers Body

Query Params

Key	Value	Description
<input checked="" type="checkbox"/> user_id	5a53d571-f85b-4373-8935-bc7eefab74fe	

Body Headers (19) Status Code

Pretty Raw Preview JSON `⌵` `⌵`

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan semua data",
4   "data": [
5     {
6       "id": "b1133150-9707-4eb6-bba2-ed2286d2a076",
7       "jenis_pengeluaran": "Transportasi",
8       "nama": "Ongkos Bus",
9       "nominal": 15000,
10      "id_user": "5a53d571-f85b-4373-8935-bc7eefab74fe"
11    },
12    {
13      "id": "a253b043-a15f-4f34-b0c0-06480c990ec5",
14      "jenis_pengeluaran": "Transportasi",
15      "nama": "Ongkos Bus",
16      "nominal": 15000,
17      "id_user": "5a53d571-f85b-4373-8935-bc7eefab74fe"
18    }
19  ]
20 }
```

### 3. Menambahkan Data Pengeluaran Rumah Tangga

Kunderfine API / Local / Pengeluaran Rumah Tangga / inputPengeluaranRT / New Request

POST

Params Headers **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

Key	Value	Description
<input checked="" type="checkbox"/> jenis_pengeluaran	Bahan sehari-hari	
<input checked="" type="checkbox"/> nama	Makan	
<input checked="" type="checkbox"/> nominal	100000	
<input checked="" type="checkbox"/> user_id	8b6ffabb-e9bb-4479-b3d1-96ca08ff3075	

Body Headers (19)

Pretty Raw Preview JSON

```
1 {"status": "success",
2  "message": "Berhasil menambahkan data baru"
3 }
4
```

### 4. Mengubah Data Pengeluaran Rumah Tangga

Kunderfine API / Local / Pengeluaran Rumah Tangga / updatePengeluaranRT / New Request

PUT

Params Headers **Body**

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL

<input checked="" type="checkbox"/> nama	Minum	
<input checked="" type="checkbox"/> nominal	180000	
<input checked="" type="checkbox"/> user_id	5a53d571-f85b-4373-8935-bc7eefab74f6	

Body Headers (19)

Pretty Raw Preview JSON

```
1 {"status": "success",
2  "message": "Berhasil memperbarui data"
3 }
4
```

## 5. Menghapus Data Pengeluaran Rumah Tangga Berdasarkan ID

Kinderfine API / Local / Pengeluaran Rumah Tangga / deletePengeluaranRT / deletePengeluaranRT

**DELETE** ▼ | `{{dev_url}}/pengeluaran-rumah-tangga/3f3f0595-4e76-48b8-aeb7-8e794ac987dd`

Params Headers Body

Query Params

Key	Value	De
Key	Value	De

Body Headers (19)

Pretty Raw Preview JSON ▼ ☰

```
1  {
2    "status": "success",
3    "message": "Berhasil menghapus data"
4  }
```

## 7. Dokumentasi Postman Modul Master Data

### 1. Mendaftarkan Master Data

Kinderfine API / Local / Master Data / Register Bank Master Data / [200] OK

**POST** ▼ | `{{url}}/master-datas/Bank`

Params Headers **Body** ●

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON** ▼

```
1  {
2    "tipe": "Bank",
3    "nilai": "BCA",
4    "aturan": "10",
5    "deskripsi": "Total digit bank BCA"
6  }
```

Body Headers (19)

Pretty Raw Preview JSON ▼ ☰

```
1  {
2    "status": "success",
3    "message": "Berhasil menambahkan data baru"
4  }
```

## 2. Mendapatkan Semua Master Data

Kinderfine API / Local / Master Data / Get All Master Data / [200] OK

GET `{{url}}/master-datas/Bank`

Params Headers Body

Query Params

Key	Value	Description
Key	Value	Description

Body Headers (19) Statu

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil mendapatkan semua data",
4   "data": [
5     {
6       "id": "22609f4b-d971-4a7e-b9c2-3fd86cf57393",
7       "tipe": "Bank",
8       "nilai": "BNI",
9       "aturan": "10",
10      "deskripsi": "10"
11    },
12    {
13      "id": "ec2b022e-5d14-4fbb-9b2d-51918a6fe737",
14      "tipe": "Bank",
15      "nilai": "BCA",
16      "aturan": "10",
17      "deskripsi": "Total digit bank BCA"
18    }
19  ]
}
```

## 3. Mengubah Data Master Data

Kinderfine API / Local / Master Data / Update Master Data / [200] OK

PUT `{{url}}/master-datas/Bank/ec2b022e-5d14-4fbb-9b2d-51918a6fe737`

Params Headers Body ●

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON**

1 {
2 "nilai": "BRI",
3 "aturan": "15",
4 "deskripsi": "Total Digit Bank BRI"
5 }

Body Headers (19)

Pretty Raw Preview JSON

```
1 {
2   "status": "success",
3   "message": "Berhasil memperbarui data"
4 }
```

## 4. Menghapus Data Master Data

Kinderfine API / Local / Master Data / Delete Master Data / [200] OK

**DELETE** ▼ `{{uri}}/master-datas/Bank/ec2b022e-5d14-4fbb-9b2d-51918a6fe737`


---

Params Headers Body

Query Params

Key	Value
Key	Value

Body Headers (19)

Pretty Raw Preview JSON ▼ 

```
1 {
2   "status": "success",
3   "message": "Berhasil menghapus data"
4 }
```



*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- Docker. (2024). *Docker Docs*. Diakses pada 10 November 2024 di <https://docs.docker.com/>
- Dockploy. (2024). *Dockploy Documentation*. Diakses pada 10 November 2024 di <https://docs.dokploy.com/docs/core>
- Express.js. (2024). *Express - Node.js web application framework*. Diakses pada 10 November 2024 di <https://expressjs.com/>
- Herrero, J. L. (2011). *2011 7th International Conference on Next Generation Web Services Practices*. IEEE.
- Muzaki, R. A. (2020). *2020 International Workshop on Big Data and Information Security (IWBIS) : Faculty of Computer Science, Universitas Indonesia, Depok, October 17th-18th, 2020*. IEEE.
- Node.js. (2024). *Index / Node.js v23.3.0 Documentation*. Diakses pada 10 November 2024 di <https://nodejs.org/docs/latest/api/>
- PostgreSQL. (2024). *PostgreSQL: Documentation*. Diakses pada 10 November 2024 di <https://www.postgresql.org/docs/>
- Typescript. (2024). *TypeScript: The starting point for learning TypeScript*. Diakses pada 10 November 2024 di <https://www.typescriptlang.org/docs/>

*[Halaman ini sengaja dikosongkan]*

## **BIODATA PENULIS I**

Nama : Dimas Fadilah Akbar  
Tempat, Tanggal Lahir : Surabaya, 11 Agustus 2024  
Jenis Kelamin : Laki-laki  
Telepon : +6281334034303  
Email : 5025211010@student.its.ac.id

### **AKADEMIS**

Kuliah : Departemen Teknik Informatika –  
FTEIC , ITS  
Angkatan : 2021  
Semester : 7 (Tujuh)