

D4TI-P3-SKPL-1194045-1194066-2021

**DOKUMEN PEMBANGUNAN PERANGKAT LUNAK**

**IMPLEMENTASI WEB SERVICE REST PADA  
SISTEM INFORMASI PENGGAJIAN KARYAWAN**

**Versi: <Draft-tgl/Final>**

untuk:


**Politeknik Pos Indonesia**

Dipersiapkan oleh:

Fahriza Rizky Amalia / 1194045  
Salsabila Vebi Natasya / 1194066

D4 Teknik Informatika – Politeknik Pos Indonesia

Bandung – 2021

	<b>Program Studi D4 Teknik Informatika Politeknik Pos Indonesia</b>	<b>Nomor Dokumen</b>		<b>Halaman</b>
		<i>D4TI-P3- SKPL-1194045- 1194066- 2021</i>		<i>1/...</i>
		<b>Revisi</b>	<i>0</i>	<i>Tgl: &lt;tgl&gt;</i>

## DAFTAR PERUBAHAN

Revisi	Deskripsi
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	
<b>G</b>	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

### Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

--	--	--	--

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>5</b>
<b>Daftar Gambar.....</b>	<b>9</b>
<b>1. Pendahuluan.....</b>	<b>12</b>
1.1. Tujuan Penulisan Dokumen .....	12
1.2. Lingkup masalah .....	12
1.3. Aturan penomoran .....	12
1.4. Referensi.....	13
1.5. Deskripsi Umum Dokumen .....	13
<b>2. Kebutuhan Perangkat Lunak.....</b>	<b>14</b>
2.1. Deskripsi Umum Sistem.....	14
2.2. Fitur Utama Perangkat Lunak .....	15
2.2.1. Kebutuhan Fungsional .....	15
2.2.2. Kebutuhan Non Fungsional .....	16
2.3. Model Use Case.....	17
2.3.1. Diagram Use Case .....	17
2.3.2. Definisi Actor.....	18
2.3.3. Definisi Use Case .....	18
2.3.4. Skenario Use Case.....	20
2.3.4.1.1. Activity Diagram UseCase Kelola Data Karyawan .....	21
2.3.4.2.1. Activity Diagram UseCase Kelola Data Posisi.....	22
2.3.4.3.1. Activity Diagram UseCase Kelola Data Department .....	24
2.3.4.4.1. Activity Diagram UseCase Kelola Data Cuti .....	26
2.3.4.5.1. Activity Diagram UseCase Kelola Data Lmebur .....	28
2.3.4.6.1. Activity Diagram UseCase Kelola Data Gaji .....	30
2.3.4.7.1. Activity Diagram UseCase Lihat Data Karyawan .....	32
2.3.4.8.1. Activity Diagram UseCase Lihat Data Cuti.....	32
2.3.4.9.1. Activity Diagram UseCase Lihat Data Lembur .....	33
2.4. Spesifikasi Tambahan .....	34
2.5. Glossary .....	34
<b>3. Model Analisis.....</b>	<b>35</b>
3.1. Realisasi Use Case Tahap Analisis.....	35

3.2.	Diagram Kelas Keseluruhan .....	40
3.3.	Kelas Analisis.....	40
3.4.	Paket Analisis.....	41
3.4.1.	Identifikasi Paket Analisis.....	41
3.4.2.	Identifikasi Kelas Analisis tiap paket.....	41
3.5.	Deskripsi Arsitektur.....	42
3.6.	Proses Bisnis.....	43
3.7.	Pedoman Perancangan .....	43
4.	Model Perancangan.....	44
4.1.	Realisasi Use Case Tahap Perancangan.....	44
4.1.1.	Use Case Kelola Data Karyawan.....	44
4.1.2.	Use Case Kelola Data Posisi .....	46
4.1.3.	Use Case Kelola Data Department .....	48
4.1.4.	Use Case Kelola Data cuti .....	50
4.1.5.	Use Case Kelola Data Lembur.....	52
4.1.6.	Use Case Lihat Gaji .....	54
4.1.7.	Use Case Lihat Data Karyawan.....	55
4.1.8.	Use Case Lihat Data Cuti .....	56
4.1.9.	Use Case Lihat Data Lembur.....	57
4.1.10.	Use Case Kelola Data Gaji .....	59
4.2.	Perancangan Detail Elemen Logical View .....	61
4.2.1.	Stereotyped Class Karyawan Server.....	61
4.2.2.	Stereotyped Class Karyawan Client.....	61
4.2.3.	Stereotyped Class TampilKaryawanClient, TambahKaryawanClient, EditKaryawanClient .....	62
4.2.4.	Stereotyped Class Posisi Server .....	62
4.2.5.	Stereotyped Class Posisi Client.....	63
4.2.6.	Stereotyped Class TampilPosisiClient, TambahPosisiClient, EditPosisiClient.....	63
4.3.	Perancangan Detail Kelas.....	64
4.3.1.	Kelas KaryawanControl.....	64
4.3.2.	Kelas Karyawan .....	64
4.3.3.	Kelas PosisiControl .....	65

4.3.4.	Kelas Posisi .....	66
4.3.5.	Kelas DepartmentControl .....	67
4.3.6.	KelasDepartment .....	67
4.3.7.	Kelas GajiControl .....	68
4.3.8.	KelasGaji .....	69
4.3.9.	KelasCuticontrol .....	69
4.3.10.	KelasCutu .....	70
4.3.11.	Kelas LemburControl .....	71
4.3.12.	KelasLembur .....	71
4.4.	Diagram Keseluruhan .....	72
4.5.	Diagram Statechart .....	74
4.6.	Perancangan Antar Muka .....	75
4.6.1.	Login .....	75
4.6.2.	Home .....	76
4.6.3.	Form Tambah Data .....	77
4.6.4.	Form Update Data .....	78
4.6.5.	Halaman Tampil Data .....	79
4.7.	Perancangan Representasi presistensi Kelas .....	80
4.7.1.	Conceptual Data Model (CDM) .....	80
4.7.2.	Physical Data Model (PDM) .....	81
4.8.	Coding standard dan Naming Convention .....	82
4.8.1.	Coding Standard .....	82
4.8.2.	Naming Convention .....	82
4.10.	Deployment diagram .....	83
5.	Implementasi .....	84
5.1.	Lingkungan Implementasi .....	84
5.1.1.	Kebutuhan Perangkat Keras .....	84
5.1.2.	Kebutuhan Perangkat Lunak .....	84
5.2.	Implementasi Kelas .....	84
5.3.	Implementasi Elemen WAE ( Component View) .....	85
5.4.	Impelementasi Antarmuka .....	86
5.5.	File Lain .....	86
6.	Pengujian .....	87

<b>6.1. Rencana dan Prosedur Pengujian .....</b>	<b>87</b>
<b>6.1.1. Rencana Pengujian .....</b>	<b>87</b>
<b>6.2. Kasus Uji .....</b>	<b>88</b>
<b>6.2.1. Pengujian Use Case Login.....</b>	<b>88</b>
<b>6.2.2. Pengujian Use Case Kelola Data Karyawan .....</b>	<b>88</b>
<b>6.2.3. Pengujian Use Case Kelola Data Posisi.....</b>	<b>89</b>
<b>6.2.4. Pengujian Use Case Kelola Data Department.....</b>	<b>89</b>
<b>6.2.5. Pengujian Use Case Kelola Data Cuti .....</b>	<b>90</b>
<b>6.2.6. Pengujian Use Case Kelola Data Lembur .....</b>	<b>90</b>
<b>6.2.7. Pengujian Use Case Kelola Data Gaji.....</b>	<b>91</b>
<b>6.2.8. Pengujian Fungsi Web Service .....</b>	<b>92</b>
<b>7. Lampiran.....</b>	<b>96</b>
<b>7.1. Pembagian Tugas .....</b>	<b>96</b>
<b>7.2. Dokumentasi Web Service.....</b>	<b>96</b>



## Daftar Gambar

Gambar 2. 1 UseCase Diagram.....	18
Gambar 2. 2 Activity Diagram UseCase Kelola Karyawan .....	21
Gambar 2. 3 Activity Diagram UseCase Kelola Posisi.....	23
Gambar 2. 4 Activity Diagram UseCase Kelola Department.....	23
Gambar 2. 5 Activity Diagram UseCase Kelola cuti.....	23
Gambar 2. 6 Activity Diagram UseCase Kelola Lembur.....	23
Gambar 2. 7 Activity Diagram UseCase Kelola Gaji .....	31
Gambar 2. 8 Activity Diagram UseCase Lihat Data Karyawan .....	32
Gambar 2. 9 Activity Diagram UseCase Lihat Data Cuti.....	33
Gambar 2. 10 Activity Diagram UseCase Lihat Data Lembur .....	33
Gambar 3. 1 Realisasi UseCase U1 .....	35
Gambar 3. 2 Realisasi UseCase U2 .....	36
Gambar 3. 3 Realisasi UseCase U3 .....	37
Gambar 3. 4 Realisasi UseCase U4 .....	37
Gambar 3. 5 Realisasi UseCase U5 .....	38
Gambar 3. 2 Realisasi UseCase U6 .....	39
Gambar 3. 7 Diagram Kelas Analisis .....	40
Gambar 3. 8 Arsitektur Sistem.....	42
Gambar 4. 1 Squence Diagram Kelola Karyawan .....	45
Gambar 4. 2 Diagram Kelas Perancangan Karyawan.....	45
Gambar 4. 3 Squence Diagram Kelola Posisi.....	47

Gambar 4. 4 Diagram Kelas Perancangan Posisi.....	47
Gambar 4. 5 Squence Diagram Kelola Department.....	49
Gambar 4. 6 Diagram Kelas Perancangan Department .....	49
Gambar 4. 7 Squence Diagram Kelola Cuti.....	51
Gambar 4. 8 Diagram Kelas Perancangan Cuti .....	52
Gambar 4. 9 Squence Diagram Kelola Lembur.....	53
Gambar 4. 10 Diagram Kelas Perancangan Lembur .....	53
Gambar 4.11 Squence Diaram Lihat Gaji.....	54
Gambar 4. 12 Diagram Kelas Perancangan Lihat Gaji.....	55
Gambar 4. 13 Squence Diagram Lihat Karyawan .....	56
Gambar 4. 14 Diagram Kelas Perancangan Lihat Karyawan .....	56
Gambar 4. 15 Squence Diagram Lihat Cuti.....	57
Gambar 4. 16 Diagram Kelas Perancangan Lihat Cuti.....	57
Gambar 4. 17 Squence Diagram Lihat Lembur .....	58
Gambar 4. 18 Diagram Kelas Perancangan Lihat Lembur .....	58
Gambar 4. 19 Squence Diagram Kelola Gaji.....	60
Gambar 4. 20 Diagram Kelas Perancangan Gaji .....	60
Gambar 4. 21 Diagram Kelas Keseluruhan Perancangan .....	73
Gambar 4. 22 Diagram Statechart.....	74
Gambar 4. 23 AntarMuka Login.....	75
Gambar 4. 24 AntarMuka Home .....	76
Gambar 4. 25 AntarMuka Form Tambah Data .....	77
Gambar 4.26 AntarMuka Form Update .....	78
Gambar 4. 27 AntarMuka Tampil Data .....	79
Gambar 4. 28 Conseptual Data Model.....	80
Gambar 4. 29 Physical Data Model .....	81
Gambar 4. 30 Deployment Diagram.....	83
Gambar 7. 1 Get Cuti.....	92
Gambar 7. 2 Get Cuti By id .....	92
Gambar 7. 3 Post Cuti.....	93

Gambar 7. 4 Put Cuti .....	94
Gambar 7. 5 Delete Cuti .....	94
Gambar 7. 6 Get Lembur .....	95
Gambar 7. 7 Get Lembur By id .....	95
Gambar 7. 8 Post Lembur .....	95
Gambar 7. 9 Put Lembur.....	96
Gambar 7. 10 Delete Lembur.....	96
Gambar 7. 11 Get Karyawan .....	97
Gambar 7. 12 Get Karyawan By id.....	97
Gambar 7. 13 Post Karyawan .....	98
Gambar 7. 14 Put Karyawan .....	99
Gambar 7. 15 Delete Karyawan.....	99
Gambar 7. 16 Get Posisi .....	100
Gambar 7. 17 Get Posisi By id.....	100
Gambar 7. 18 Post Posisi .....	100
Gambar 7. 19 Put Posisi.....	101
Gambar 7. 20 Delete Posisi.....	101
Gambar 7. 21 Get Gaji .....	102
Gambar 7. 22 Get Gaji By id .....	102
Gambar 7. 23 Post Gaji.....	103
Gambar 7. 24 Put Gaji .....	104
Gambar 7. 25 Delete Gaji .....	104
Gambar 7. 26 Get Department.....	105
Gambar 7. 27 Get Department By id .....	105
Gambar 7. 28 Post Department.....	106
Gambar 7. 29 Put Department .....	106
Gambar 7. 30 Delete Department .....	107
Gambar 7. 36 Get User .....	107
Gambar 7.37 Get User By id.....	108
Gambar 7.38 Post User .....	108

Gambar 7. 39 Put User.....	109
Gambar 7. 40 Delete User.....	109

## **1. Pendahuluan**

### ***1.1. Tujuan Penulisan Dokumen***

Dokumen pembangunan Perangkat lunak (SKPL) ini dibuat dengan tujuan untuk digunakan sebagai acuan dalam pembuatan perangkat lunak. Dokumen ini berisi tentang apa saja kebutuhan yang harus dipersiapkan agar perangkat lunak dapat dikembangkan. Dalam dokumen ini juga mendefinisikan Batasan perancangan perangkat lunak yang akan dikerjakan. Dimana system informasi ini diharapkan bisa membantu dalam pendataan gaji pokok, lembur, dan cuti yang pasti akan lebih efisien dan aman.

Dokumen ini juga digunakan oleh pengembang perangkat lunak sebagai acuan secara teknis yang mungkin berguna bagi pengembangan system informasi ini pada tahap selanjutnya.

### ***1.2. Lingkup masalah***

Perangkat lunak System Informasi Penggajian Karyawan dikembangkan dengan tujuan untuk :

1. Menampilkan informasi tentang penggajian karyawan
2. Melakukan perencanaan dalam pengolahan data gaji karyawan yang berupa gaji pokok, upah lembur dan cuti dalam suatu perusahaan
3. Merubah system penggajian karyawan konvensional ke modern dengan memanfaatkan salah satu web service yaitu REST

### ***1.3. Aturan penomoran***

Setiap kebutuhan perangkat lunak yang ada didalam dokumen ini akan diberi penomoran yang sesuai dengan formay berikut ini :

SKPL-JK-XX.Y, dengan penjelesan :

- JK adalah singkatan dari Jenis Kebutuhan dan bisa diganti dengan F jika kebutuhan fungsional dan NF jika untuk kebutuhan Non fungsional.
- XX adalah nomor kebutuhan fungsi
- Y merupakan nomor kebutuhan fungsi turunan dari XX.

#### ***1.4. Referensi***

Referensi-referensi yang digunakan dalam pembuatan dokumen dan dalam pengembangan perangkat lunak adalah :

- a. Soares, J. D. C. L. Perapan Metode Sistem Rekomendasi Hibrida Pada Sistem Pemandu Lokasi Wisata di Timor-Leste. Dalam penelitian ini membuat aplikasi system pemandu lokasi wisata di Timor-Leste yang dibuat laporannya dalam bentuk SKPL.
- b. Gustina, R., & Leidiyana, H. (2020). Sistem Informasi Penggajian Karyawan Berbasis Web Menggunakan Framework Laravel. Dalam penelitian ini membuat aplikasi system informasi penggajian karyawan yang dibuat dengan framework Laravel. Dengan dibangunnya suatu system informasi penggajian diharapkan dapat mempermudah perusahaan dalam memproses gaji karyawan dengan cepat dan tepat.
- c. Pratama, A. F., Indriana, I. H., & Matondang, N. H. (2021). Perancangan Sistem Penggajian Pengajar dengan Menggunakan Framework Angular dan Codeigniter (Studi Kasus MABIT Nurul Fikri). Penelitian ini menggunakan Mysql sebagai basis data, menggunakan framework codeigniter untuk pembuatan Rest api sebagai Backend dan menggunakan angular dalam pembuatan Frontend yang menggunakan Rest api sebagai layanan transfer data.

#### ***1.5. Deskripsi Umum Dokumen***

Pada dokumen SKPL ini secara umum terdiri dari 6 bagian utama. Pada bagian bagian pertama terdapat pendahuluan. Yang berisi tentang tujuan pembuatan dokumen, lingkup masalah dalam pembuatan dokumen, aturan penomoran yang digunakan dalam dokumen, referensi yang digunakan baik dalam pembuatan dokumen maupun dalam pengembangan perangkat lunak itu sendiri.

Pada bagian kedua terdapat kebutuhan perangkat lunak. Yang berisi tentang deskripsi umum system yang menjelaskan system secara keseluruhan, fitur utama perangkat lunak dan kebutuhan fungsional dan non fungsional system informasi penggajian, model Use case yang terdiri dari diagram Use case yang berguna untuk menjelaskan alur system yang akan dibuat, definisi actor, definisi Use case, skenario Use case.

Pada bagian ketiga terdapat model analisis. Yang berisi tentang relisasi Use Case ke tahap analisis, diagram kelas keseluruhan, kelas analisis, paket analisis yang terdiri dari identifikasi paket analysis dan identifikasi kelas analisis tiap paket, deskripsi arsitektur, pedoman perancangan.

Pada bagian keempat terdapat model perancangan. Yang berisi realisasi Use case pada ke tahap perancangan yang terdiri dari identifikasi elemen WAE -logical view, identifikasi kelas perancangan, sequence diagram, diagram kelas perancangan. Perancangan detil elemen logical view yang berisi stereotyped class, perancangan detil kelas, diagram kelas keseluruhan, algoritma/query, diagram statechart, perancangan antarmuka, perancangan representasi persistensi kelas, coding standard dan naming convention dan deployment diagram.

Kemudian pada bagian kelima adalah bagian implementasi. Yang berisi lingkungan implementasi, implementasi kelas, implementasi elemen WAE, implementasi antarmuka. Yang terakhir pada bagian keenam ada pengujian. Yang berisi tentang rencana dan prosedur pengujian, kasus uji, evaluasi pengujian.

## **2. Kebutuhan Perangkat Lunak**

### ***2.1.Deskripsi Umum Sistem***

Sistem informasi penggajian karyawan merupakan perangkat lunak yang dikembangkan dengan implementasi Web Service Rest, data yang dibutuhkan berupa informasi tentang data department perusahaan, data posisi karyawan, data gaji karyawan, data cuti karyawan, data lembur karyawan dan data karyawan berdasarkan atribut yang dibutuhkan dalam penggajian karyawan. Sistem informasi ini diharapkan dapat membantu karyawan dalam memperoleh informasi penggajian dengan mudah, cepat dan akurat.

Fungsi – fungsi yang terdapat dalam sistem informasi ini antara lain : department, posisi, gaji, cuti, lembur dan karyawan. Dimana fungsi department adalah proses untuk menampilkan informasi department. Fungsi posisi adalah proses untuk menampilkan informasi posisi. Fungsi karyawan adalah proses menampilkan data karyawan. Fungsi gaji adalah proses menampilkan informasi gaji karyawan. Fungsi cuti adalah proses menampilkan informasi cuti karyawan. Dan fungsi lembur adalah proses menampilkan informasi lembur karyawan.

Sistem informasi penggajian karyawan ini berjalan dengan berbasis web. Sedangkan untuk lingkungan pemrogramannya menampilkan data dengan format JSON dan untuk databasenya menggunakan Mysql.

## **2.2. Tujuan**

Tujuan dari perancangan system informasi penggajian karyawan diharapkan menjadi solusi yang dapat digunakan untuk mempermudah dalam proses perhitungan gaji dan pengolahan data karyawan. Sehingga dalam pembuatan laporan dapat dilakukan keakuratan data serta efisiensi waktu serta dapat membantu dalam keamanan data.

## **2.3. Fitur Utama Perangkat Lunak**

### **2.3.1. Kebutuhan Fungsional**

Kebutuhan fungsional pada sistem informasi penggajian karyawan adalah sebagai berikut :

#### **1. Fungsi Department (SKPL-JF-01)**

Fungsi department adalah fungsi yang digunakan untuk menampilkan dan juga mengedit, menambah serta menghapus informasi department perusahaan.

#### **2. Fungsi Posisi (SKPL-JF-02)**

Fungsi posisi adalah fungsi yang digunakan untuk menampilkan dan juga mengedit, menambah serta menghapus informasi posisi karyawan.

#### **3. Fungsi Karyawan (SKPL-JF-03)**

Fungsi karyawan adalah fungsi yang digunakan untuk menampilkan dan

juga mengedit, menambah serta menghapus informasi karyawan.

4. Fungsi Cuti (**SKPL-JF-04**)

Fungsi cuti adalah fungsi yang digunakan untuk menampilkan dan juga mengedit, menambah serta menghapus informasi cuti karyawan.

5. Fungsi Lembur (**SKPL-JF-05**)

Fungsi lembur adalah fungsi yang digunakan untuk menampilkan dan juga mengedit, menambah serta menghapus informasi lembur karyawan.

6. Fungsi Gaji (**SKPL-JF-06**)

Fungsi gaji adalah fungsi yang digunakan untuk menampilkan dan juga mengedit, menambah serta menghapus informasi karyawan.

7. Fungsi Admin (**SKPL-JF-07**)

Fungsi admin adalah fungsi yang digunakan untuk mengelolah semua data yang ada di system.

**2.3.2. Kebutuhan Non Fungsional**

1. Kebutuhan Non-fungsional Perangkat keras (**SKPL-JNF-01**)

Dalam proses merancang dan membangun sistem informasi penggajian karywan diperlukan beberapa perangkat yang digunakan untuk membangun sistem informasi tersebut.

a. Kebutuhan minimal hardware dalam menjalankan system (**SKPL-JNF-01-01**)

- Laptop dengan spesifikasi berikut :
  1. Processor 1GHz
  2. Memory (RAM) dengan kapasitas 1 GB
  3. Hardisk dengan ruang penyimpanan 100 GB
  4. VGA Intel HD Graphic
- Koneksi internet 1 Mbps

b. Kebutuhan perangkat lunak (**SKPL-JNF-01-02**)

- Bahasa pemrograman : PHP
- Database : MySQL
- WebBrowser : Google Chrome, Mozilla Firefox

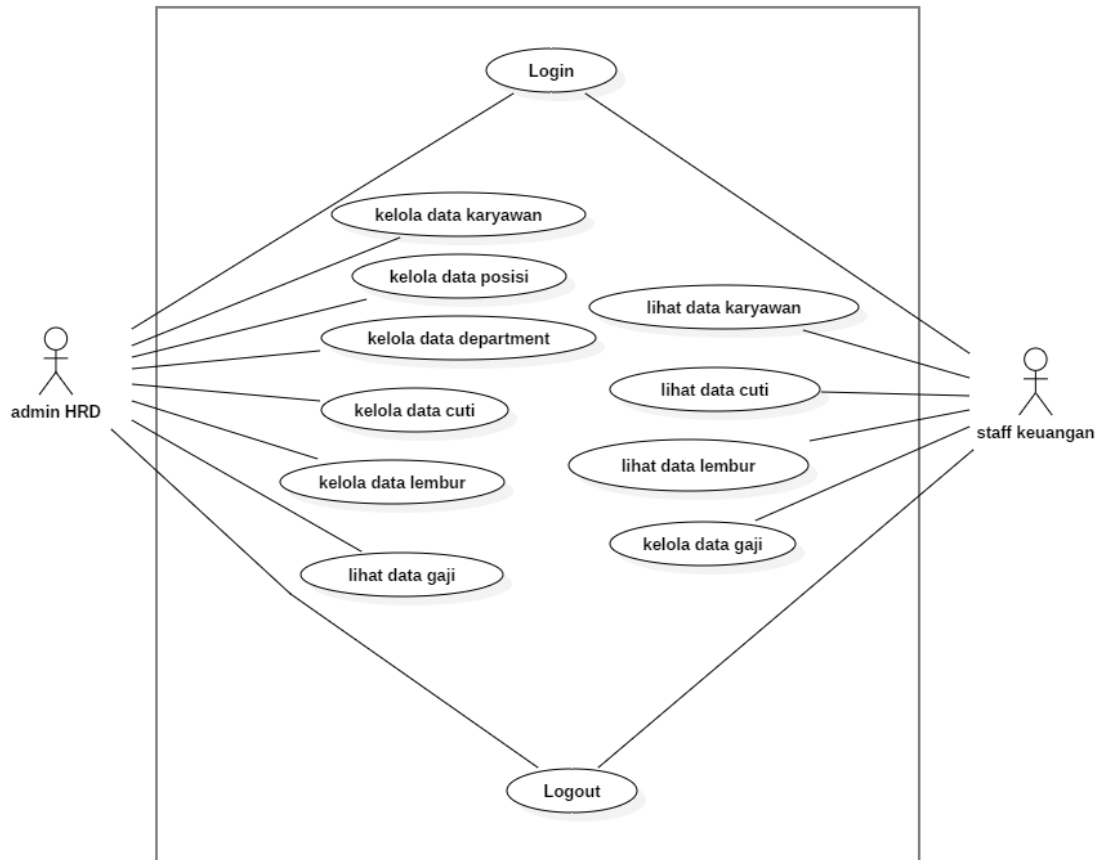


- Software :
  1. Microsoft Office Word 2013, digunakan untuk menyusun laporan dan dokumentasi.
  2. Postman, digunakan untuk menguji API yang tersedia di REST web service.
  3. Staruml, digunakan untuk merancang Usecase diagram, Activity diagram, Sequence diagram.

## ***2.4.Model Use Case***

### ***2.4.1. Diagram Use Case***

Dari hasil penjabaran pada kebutuhan fungsional, maka dibuat pemodelan system menggunakan diagram UML. Berikut ini adalah pemodelan yang Digambarkan dengan *Use Case Diagram* yang menggambarkan actor dengan kebutuhan fungsional system informasi penggajian karyawan :



Gambar 2.1. UseCase Diagram

#### 2.4.2. Definisi Actor

Pada system penggajian karyawan ini hanya terdapat satu actor yang memiliki wewenang untuk mengakses system ini, berikut adalah penjelasannya :

Tabel 2.1. Definisi Aktor

No	Actor	Deskripsi
1	Staff Keuangan	Actor pada pada role ini memiliki wewenang untuk mengelola data gaji yang ada pada system penggajian karyawan ini.
2	HRD	Actor pada role ini memiliki wewenang untuk mengelola data posisi, department, cuti, lembur.

#### 2.4.3. Definisi Use Case

Table 2.2. Definisi Use Case

No	Use Case	Deskripsi
----	----------	-----------

U1	Login	<b>Deskripsi singkat :</b> actor melakukan login dengan username dan password <b>Aktor :</b> admin personalia dan staff keuangan <b>Trigger :</b> actor memasukkan username dan password
U2	Kelola Data karyawan	<b>Deskripsi singkat :</b> actor dapat melakukan tambah, edit, hapus data karyawan <b>Aktor :</b> admin personalia <b>Trigger :</b> actor memilih menu data karyawan
U3	Kelola data posisi	<b>Deskripsi singkat :</b> actor dapat melakukan tambah, edit, hapus data posisi <b>Aktor :</b> admin personalia <b>Trigger :</b> actor memilih menu data posisi
U4	Kelola data department	<b>Deskripsi singkat :</b> actor dapat melakukan tambah, edit, hapus data department <b>Aktor :</b> admin personalia <b>Trigger :</b> actor memilih menu data department
U5	Kelola data cuti	<b>Deskripsi singkat :</b> actor dapat melakukan tambah, edit, hapus data cuti <b>Aktor :</b> admin personalia <b>Trigger :</b> actor memilih menu data cuti
U6	Kelola data lembur	<b>Deskripsi singkat :</b> actor dapat melakukan tambah, edit, hapus data lembur <b>Aktor :</b> admin personalia <b>Trigger :</b> actor memilih menu data lembur
U7	Lihat data gaji	<b>Deskripsi singkat :</b> actor melihat daftar data gaji <b>Aktor :</b> admin personalia <b>Trigger :</b> actor memilih menu data gaji
U8	Lihat data karyawan	<b>Deskripsi singkat :</b> actor melihat daftar data karyawan <b>Aktor :</b> staff keuangan <b>Trigger :</b> actor memilih menu data karyawan
U9	Lihat data cuti	<b>Deskripsi singkat :</b> actor melihat daftar data cuti <b>Aktor :</b> staff keuangan <b>Trigger :</b> actor memilih menu data cuti
U10	Lihat data lembur	<b>Deskripsi singkat :</b> actor melihat daftar data lembur <b>Aktor :</b> staff keuangan

		<b>Trigger</b> : actor memilih menu data lembur
U11	Kelola data gaji	<b>Deskripsi singkat</b> : actor dapat melakukan tambah, edit, hapus data lembur <b>Aktor</b> : staff keuangan <b>Trigger</b> : actor memilih menu data gaji

#### 2.4.4. Skenario Use Case

##### 2.4.4.1. Skenario Use Case Kelola Data Karyawan

Nama Use Case : Kelola data karyawan

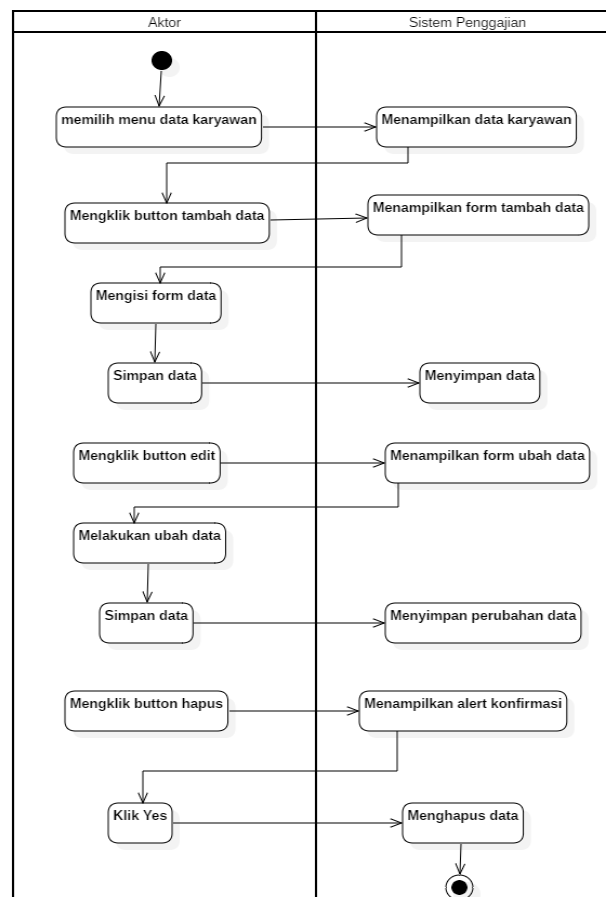
Skenario :

*Table 2.3. Skenario Use Case Kelola Data Karyawan*

Aksi Aktor	Reaksi System
<b>Skenario Normal : Tambah Data Karyawan</b>	
1. Memilih menu data karyawan	
	2. Menampilkan data karyawan
3. Mengklik tambah data	
	4. Menampilkan form tambah data
5. Mengisi form data	
6. Simpan data	
	7. Menyimpan data yang sudah dibuat
<b>Skenario Normal : Edit Data Karyawan</b>	
1. Memilih menu data karyawan	
	2. Menampilkan data karyawan
3. Mengklik tombol ubah data	
	4. Menampilkan form ubah data
5. Melakukan perubahan data	
6. Simpan data	
	7. Menyimpan data yang sudah di ubah.
<b>Skenario Normal : Hapus Data karyawan</b>	
1. Memilih menu data karyawan	
	2. Menampilkan data karyawan

3. Mengklik tombol hapus pada data yang ingin dihapus	
	4. Menampilkan alert konfirmasi
5. Klik yes	
	6. Menghapus data

#### 2.4.4.1.1. Activity Diagram UseCase Kelola Data Karyawan



Gambar 2.2. Activity diagram UseCase Kelola Karyawan

#### 2.4.4.2. Skenario Use Case Kelola Data Posisi

Nama Use Case : Kelola Data Posisi

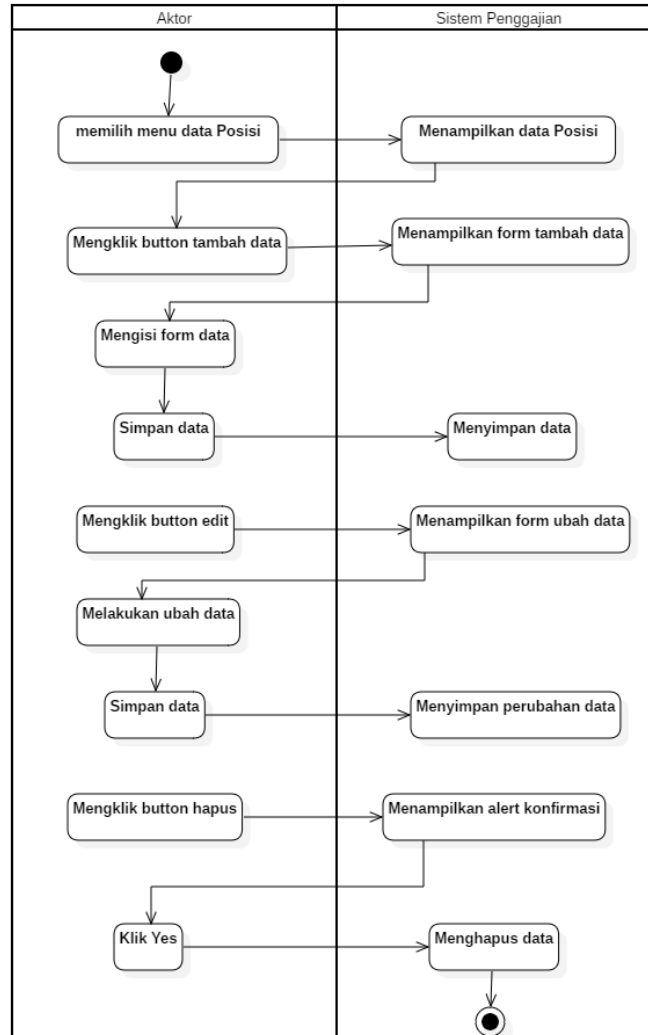
Skenario :

Table 2.4. Skenario UseCase Kelola Data Posisi

Aksi Aktor	Reaksi System
------------	---------------

<b>Skenario Normal : Tambah Data Posisi</b>	
1. Memilih menu data posisi	
	2. Menampilkan data posisi
3. Mengklik tambah data	
	4. Menampilkan form tambah data
5. Mengisi form data	
6. Simpan data	
	7. Menyimpan data yang sudah dibuat
<b>Skenario Normal : Edit Data Posisi</b>	
1. Memilih menu data posisi	
	2. Menampilkan data posisi
3. Mengklik tombol ubah data	
	4. Menampilkan form ubah data
5. Melakukan perubahan data	
6. Simpan data	
	7. Menyimpan data yang sudah di ubah.
<b>Skenario Normal : Hapus Data Posisi</b>	
1. Memilih menu data posisi	
	2. Menampilkan data posisi
3. Mengklik tombol hapus pada data yang ingin dihapus	
	4. Menampilkan alert konfirmasi
5. Klik yes	
	6. Menghapus data

#### 2.4.4.2.1. Activity Diagram UseCase Kelola Data Posisi



Gambar 2.3. Activity Diagram Kelola Data Posisi

#### 2.4.4.3. Skenario Use Case Kelola Data Department

Nama Use Case : Kelola Data Department

Skenario :

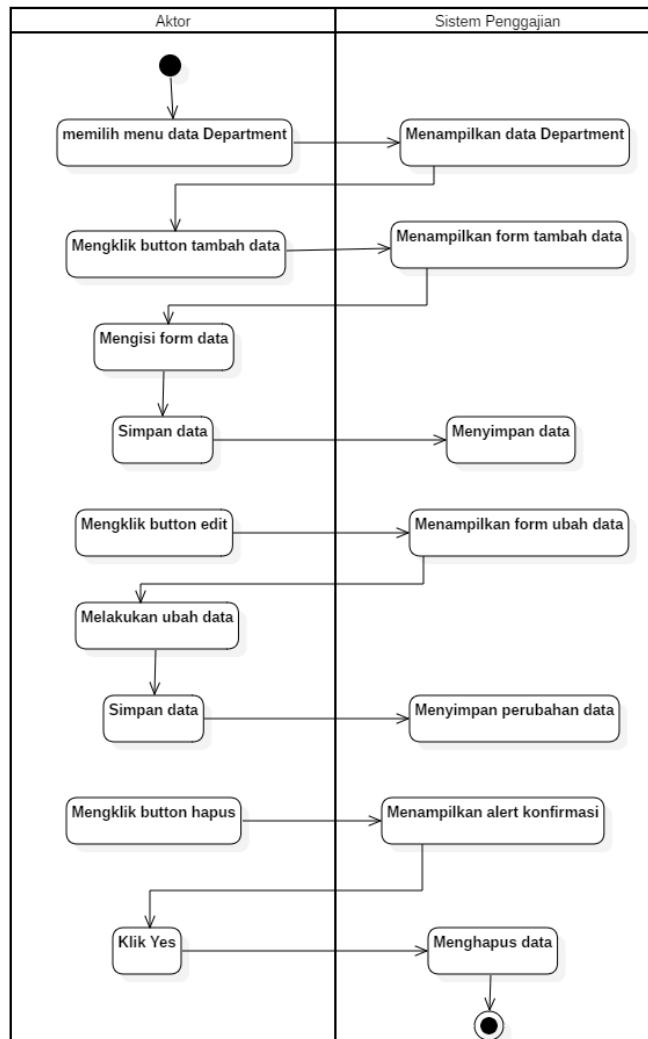
Table 2.5. Skenario UseCase Kelola Data Department

Aksi Aktor	Reaksi System
<b>Skenario Normal : Tambah Data Department</b>	
1. Memilih menu data department	
	2. Menampilkan data department
3. Mengklik tambah data	

	4. Menampilkan form tambah data
5. Mengisi form data	
6. Simpan data	
	7. Menyimpan data yang sudah dibuat
<b>Skenario Normal : Edit Data Department</b>	
1. Memilih menu data department	
	2. Menampilkan data department
3. Mengklik tombol ubah data	
	4. Menampilkan form ubah data
5. Melakukan perubahan data	
6. Simpan data	
	7. Menyimpan data yang sudah di ubah.
<b>Skenario Normal : Hapus Data department</b>	
1. Memilih menu data department	
	2. Menampilkan data department
3. Mengklik tombol hapus pada data yang ingin dihapus	
	4. Menampilkan alert konfirmasi
5. Klik yes	
	6. Menghapus data

#### 2.4.4.3.1. Activity Diagram UseCase Kelola Data Department





Gambar 2.4. Activity Diagram Kelola Data Department

#### 2.4.4.4. Skenario Use Case Kelola Data Cuti

Nama : Use Case Kelola data cuti

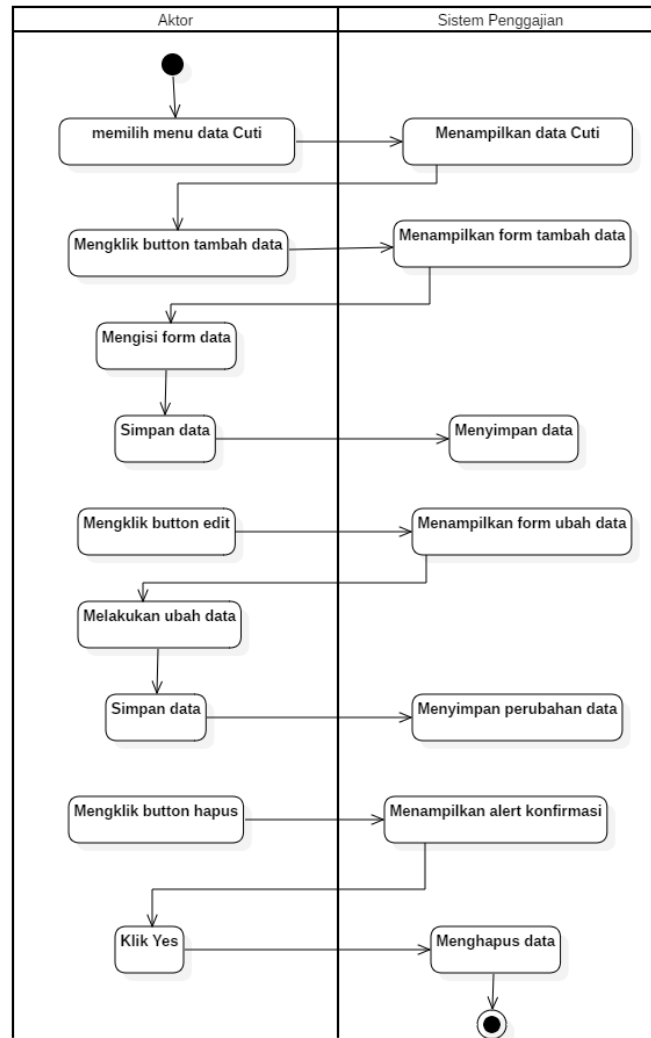
Skenario :

Table 2.6. Skenario Use Case Kelola Data Cuti

Aksi Aktor	Reaksi System
<b>Skenario Normal : Tambah Data Cuti</b>	
1. Memilih menu data cuti	
	2. Menampilkan data cuti
3. Mengklik tambah data	
	4. Menampilkan form tambah data

5. Mengisi form data	
6. Simpan data	
	7. Menyimpan data yang sudah dibuat
<b>Skenario Normal : Edit Data Cuti</b>	
1. Memilih menu data cuti	
	2. Menampilkan data cuti
3. Mengklik tombol ubah data	
	4. Menampilkan form ubah data
5. Melakukan perubahan data	
6. Simpan data	
	7. Menyimpan data yang sudah di ubah.
<b>Skenario Normal : Hapus Data Cuti</b>	
1. Memilih menu data cuti	
	2. Menampilkan data karyawan
3. Mengklik tombol hapus pada data yang ingin dihapus	
	4. Menampilkan alert konfirmasi
5. Klik yes	
	6. Menghapus data

#### 2.4.4.4.1. Activity Diagram UseCase Kelola Data Cuti



Gambar 2.5. Activity Diagram Kelola Data Cuti

#### 2.4.4.5. Skenario Use Case Kelola Data Lembur

Nama Use Case : Kelola Data Lembur

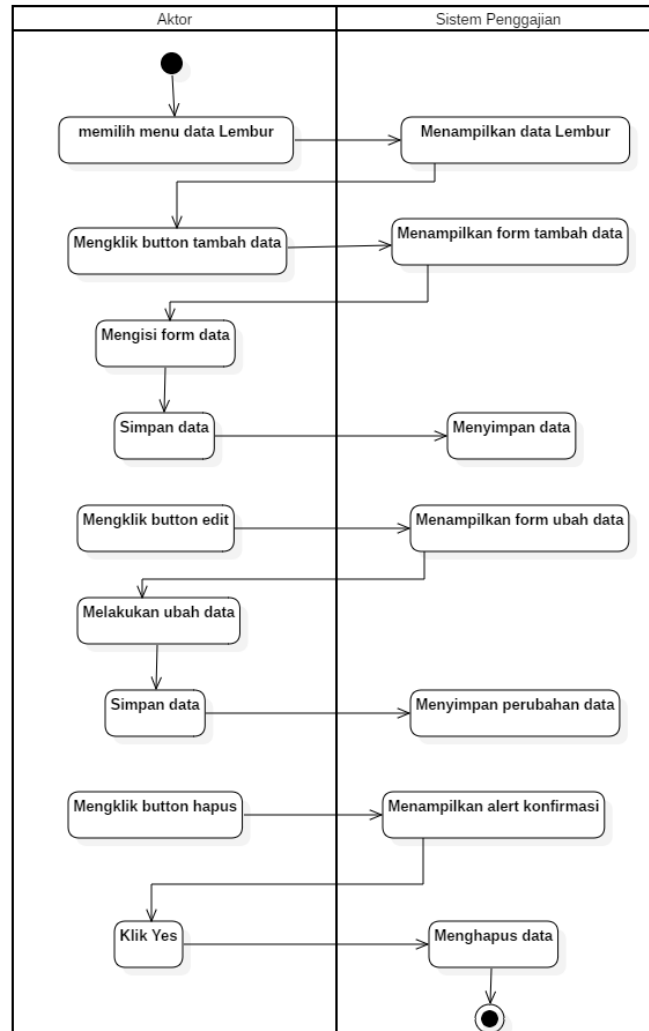
Skenario

Table 2.7. Skenario Use Case Data Lembur

Aksi Aktor	Reaksi System
<b>Skenario Normal : Tambah Data Lembur</b>	
1. Memilih menu data Lembur	2. Menampilkan data lembur
3. Klik “tambah data”	

	4. Menampilkan form tambah data
5. Mengisi form data	
6. Simpan data	
	7. Menyimpan data yang sudah dibuat
<b>Skenario Normal :Edit Data Lembur</b>	
1. Memilih menu data lembur	
	2. Menampilkan data lembur
3. Klik button ubah	
	4. Menampilkan form ubah data
5. Melakukan perubahan data	
6. Simpan data	
	7. Menyimpan data yang sudah di ubah
<b>Skenario Normal :Hapus Data Lembur</b>	
1. Memilih menu data lembur	
	2. Menampilkan data lembur
3. Mengklik tombol hapus pada data yang ingin dihapus	
	4. Menampilkan alert konfirmasi
5. Klik yes	
	6. Menghapus data

#### 2.4.4.5.1. Activity Diagram UseCase Kelola Data Lmebur



Gambar 2.6. Activity Diagram Kelola Data Lembur

#### 2.4.4.6. Skenario Use Case Kelola Data Gaji

Nama Use Case : Kelola Data Gaji

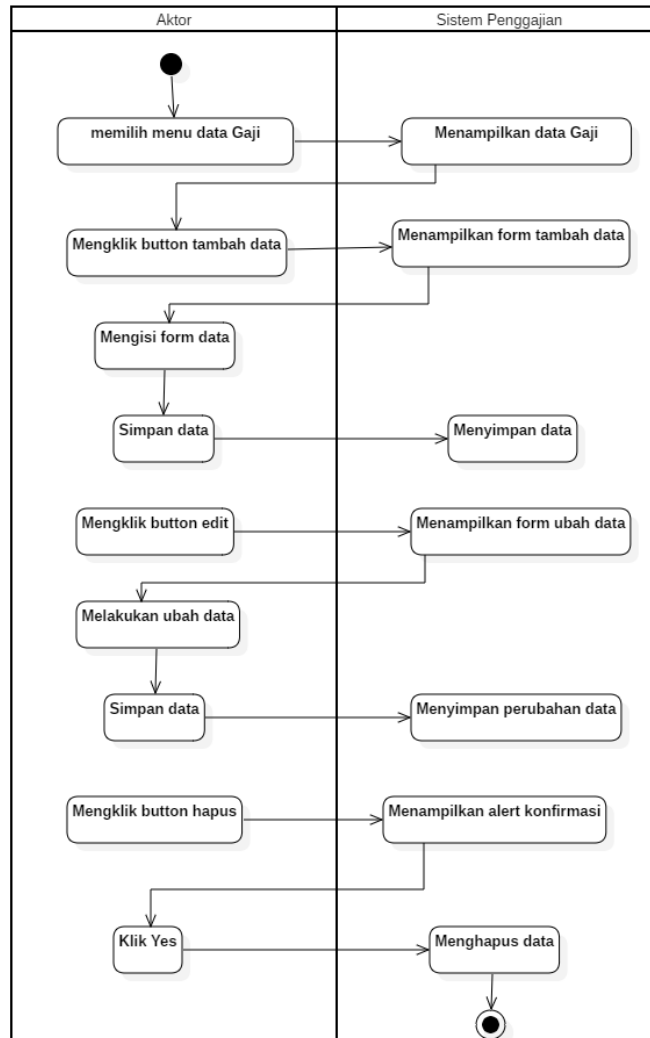
Skenario :

Table 2.8. Skenario Use Case Kelola Data Gaji

Skenario Aktor	Reaksi system
<b>Skenario Normal : Tambah Data Gaji</b>	
1. Memilih menu data gaji	
	2. Menampilkan list data gaji
3. Klik “Tambah Data”	
	4. Menampilkan form tambah data

5. Mengisi form data	
6. Klik button simpan	
	7. Menyimpan data yang sudah dibuat
<b>Skenario Normal : Edit Data Gaji</b>	
1. Memilih menu data Gaji	
	2. Menampilkan list data gaji
3. Klik button ubah	
	4. Menampilkan form ubah data
5. Melakukan perubahan data	
6. Simpan data	
	7. Menyimpan data yang sudah diubah
<b>Skenario Normal :Hapus Data Gaji</b>	
1. Memilih menu data gaji	
	2. Menampilkan data gaji
3. Klik button hapus pada data yang ingin dihapus	
	4. Menampilkan alert konfirmasi
5. Klik yes	
	6. Menghapus data

#### 2.4.4.6.1. Activity Diagram UseCase Kelola Data Gaji



Gambar 2.7. Activity Diagram Kelola Data Gaji

#### 2.4.4.7. Skenario Use Case Lihat Data Karyawan

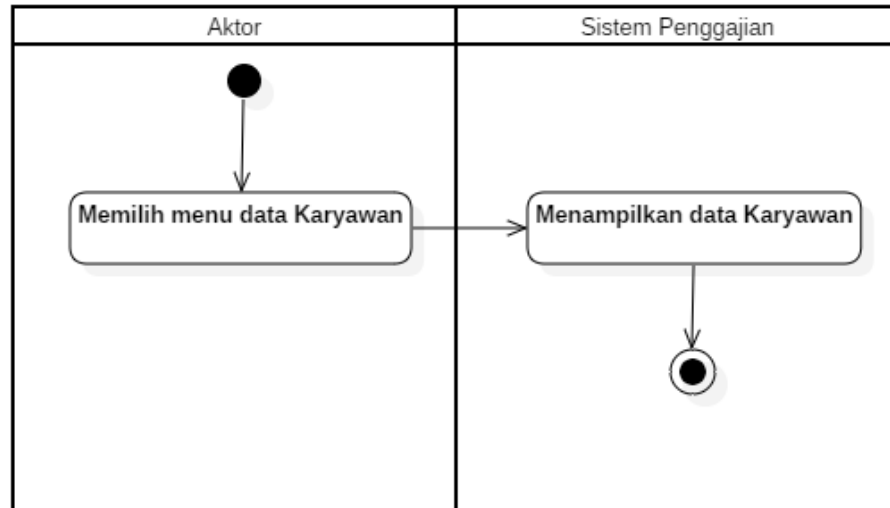
Nama Use Case : Lihat Data Karyawan

Skenario :

Table 2.9. Skenario Lihat Data Karyawan

Aksi Aktor	Reaksi System
<b>Skenario Normal : Lihat Data Karyawan</b>	
1. Memilih menu data Karyawan	
	2. Menampilkan data Karyawan

#### 2.4.4.7.1. Activity Diagram UseCase Lihat Data Karyawan



Gambar 2.8. Activity diagram Lihat Data Karyawan

#### 2.4.4.8. Skenario Use Case Lihat Data Cuti

Nama Use Case : Lihat Data Cuti

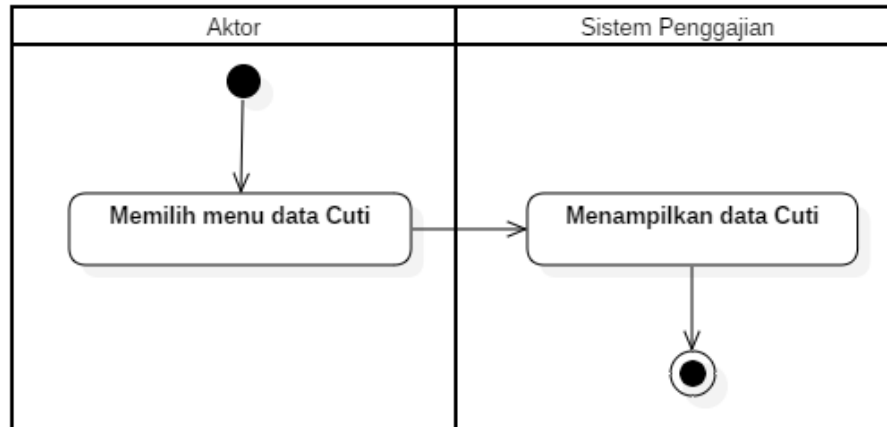
Skenario :

Table 2.10. Skenario Use Case Lihat Data Cuti

Skenario Aktor	Reaksi System
<b>Skenario Normal : Lihat Data Cuti</b>	
1. Memilih menu data cuti	
	2. Menampilkan list data cuti

#### 2.4.4.8.1. Activity Diagram UseCase Lihat Data Cuti





*Gambar 2.9. Activity Diagram Lihat Data Cuti*

#### 2.3.4.9.Skenario Use Case Lihat Data Lembur

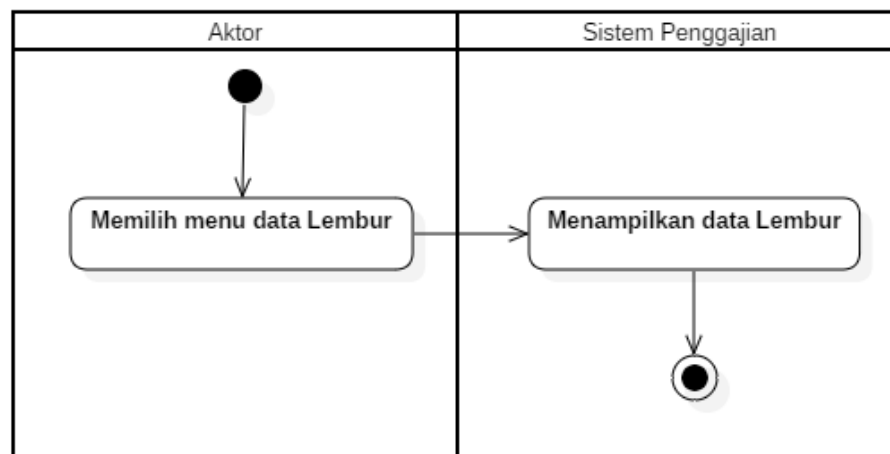
Nama Use Case : Lihat Data Lembur

Skenario :

*Table 2.11. Skenario Use Case Lihat Data Lembur*

Skenario Aktor	Reaksi System
<b>Skenario Normal : Lihat Data Lembur</b>	
1. Memilih menu data lembur	
	2. Menampilkan list data lembur

##### 2.3.4.9.1. Activity Diagram UseCase Lihat Data Lembur



*Gambar 2.10. Activity Diagram Use Case Lihat Data Lembur*

## 2.5. Spesifikasi Tambahan

Berikut adalah atribut system perangkat lunak dan performansi :

Table 2.12. Spesifikasi Tambahan

No SKPL	Kebutuhan	Tuntutan Kebutuhan
SKPL-JNF-02	Waktu tanggap	Tidak lebih dari 10 detik
SKPL-JNF-03	Ketersediaan	Harus lengkap dan Dapat mengelola (menambah, mengupdate, menghapus data)
SKPL-JNF-04	Keamanan	System menyediakan form login sehingga terdapat username dan password yang terdaftar pada database.
SKPL-JNF-05	Format Menu	System penggajian menggunakan format menu yang dibuat sederhana dan user friendly.
SKPL-JNF-06	Warna Aplikasi	System penggajian ini menggunakan tampilan yang memiliki warna dark.

## 2.6. Glossary

Table 2.13. Glossary

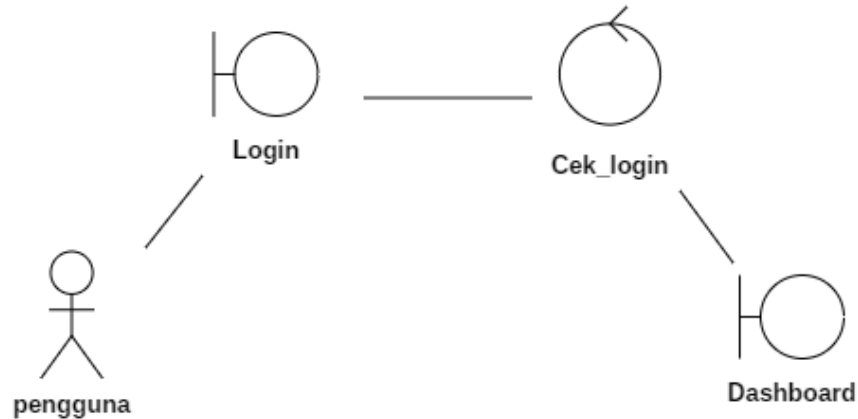
Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
Web Service	Merupakan suatu system perangkat lunak yang didesain untuk mendukung interaksi mesin ke mesin.
PHP	Merupakan bahasa penulisan script open-source yang biasanya digunakan dalam pemrograman atau pengembangan website
REST	Merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web.

### 3. Model Analisis

#### 3.1. Realisasi Use Case Tahap Analisis

##### a) Realisasi Use Case U1

Berikut adalah bentuk realisasi tahap analisis dari usecase U1 yaitu Login :



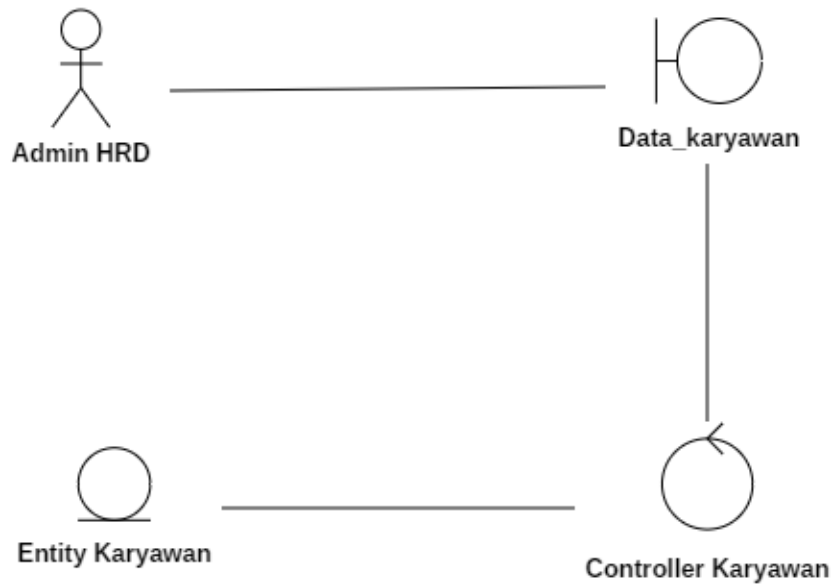
*Gambar 3.1. Realisasi UseCase U1*

Keterangan :

1. Pengguna masuk ke halaman login dan selanjutnya menginputkan username dan password.
2. Kelas control Cek\_login akan melakukan verifikasi dan autentifikasi hak akses apakah pengguna sebagai Admin HRD atau Staff Keuangan.
3. Pengguna masuk ke dalam dashboard sesuai dengan hak akses.

### b) Realisasi Use Case U2

Berikut adalah bentuk realisasi tahap analisis dari usecase U2 yaitu Kelola Data Karyawan:

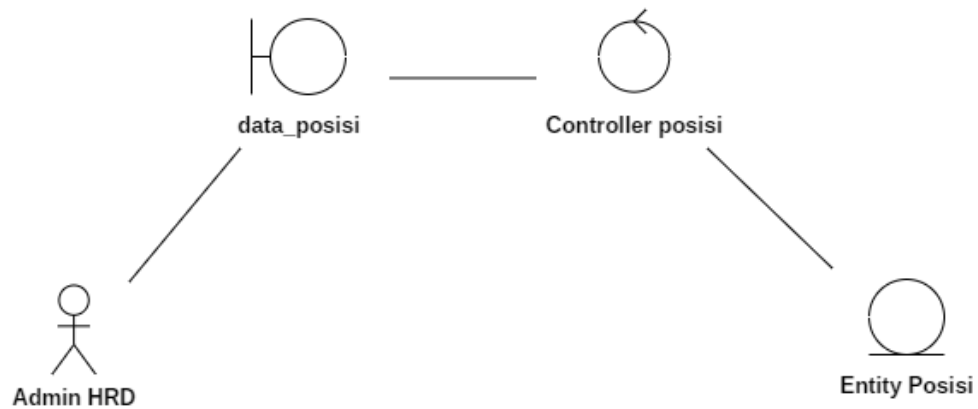


*Gambar 3.2. Realisasi UseCase U2*

Keterangan :

1. Admin HRD memilih menu data karyawan untuk melakukan pengelolaan data yaitu : Add, Update, Delete.
2. Controller akan melakukan verifikasi sesuai dengan permintaan, contohnya seperti verifikasi tambah data apakah sudah sesuai.
3. Kemudian data masuk ke database data karyawan.

### c) Realisasi Use Case U3

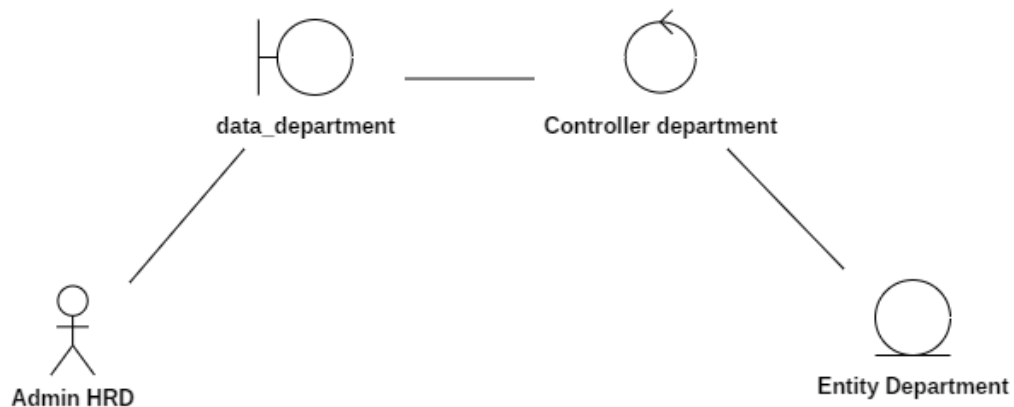


*Gambar 3.3. Realisasi Use Case U3*

Keterangan :

1. Admin HRD memilih menu data posisi untuk melakukan pengelolaan data yaitu : Add, Update, Delete.
2. Controller akan melakukan verifikasi sesuai dengan permintaan, contohnya seperti verifikasi tambah data apakah sudah sesuai.
3. Kemudian data masuk ke database data posisi.

**d) Realisasi Use Case U4**

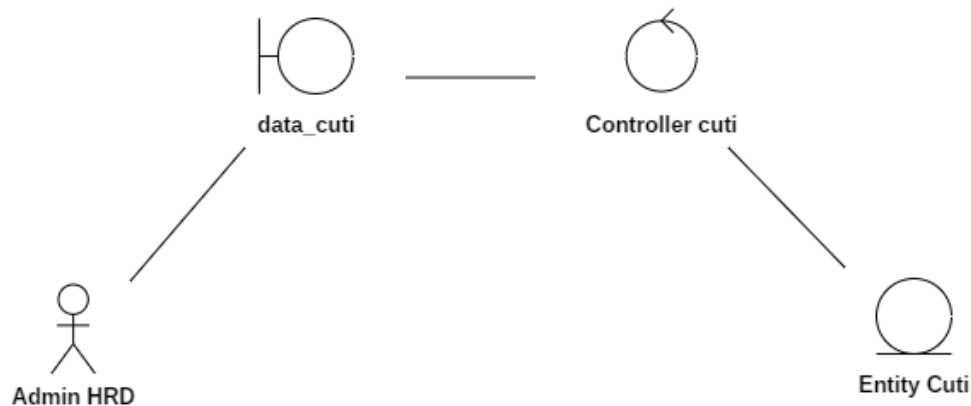


*Gambar 3.4. Realisasi UseCase U4*

Keterangan :

1. Admin HRD memilih menu data Department untuk melakukan pengelolaan data yaitu : Add, Update, Delete.
2. Controller akan melakukan verifikasi sesuai dengan permintaan, contohnya seperti verifikasi tambah data apakah sudah sesuai.
3. Kemudian data masuk ke database data department.

**e) Realisasi Use Case U5**

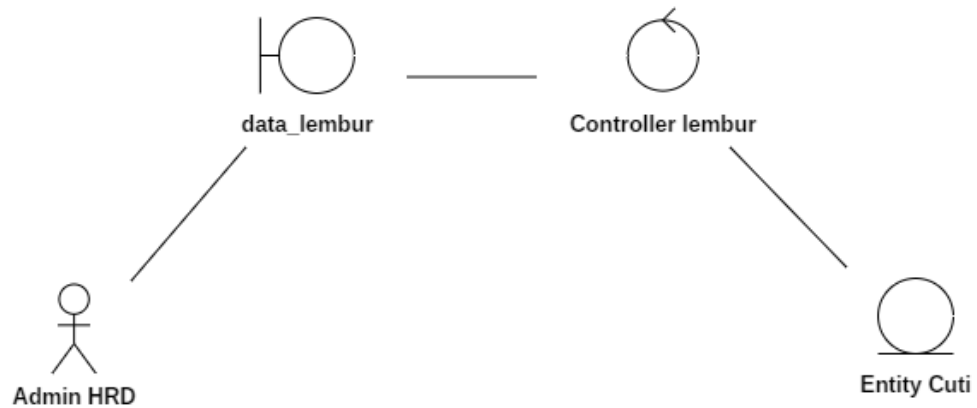


*Gambar 3.5. Realisasi UseCase U5*

Keterangan :

1. Admin HRD memilih menu data cuti untuk melakukan pengelolaan data yaitu : Add, Update, Delete.
2. Controller akan melakukan verifikasi sesuai dengan permintaan, contohnya seperti verifikasi tambah data apakah sudah sesuai.
3. Kemudian data masuk ke database data cuti

**f) Realisasi Use Case U6**

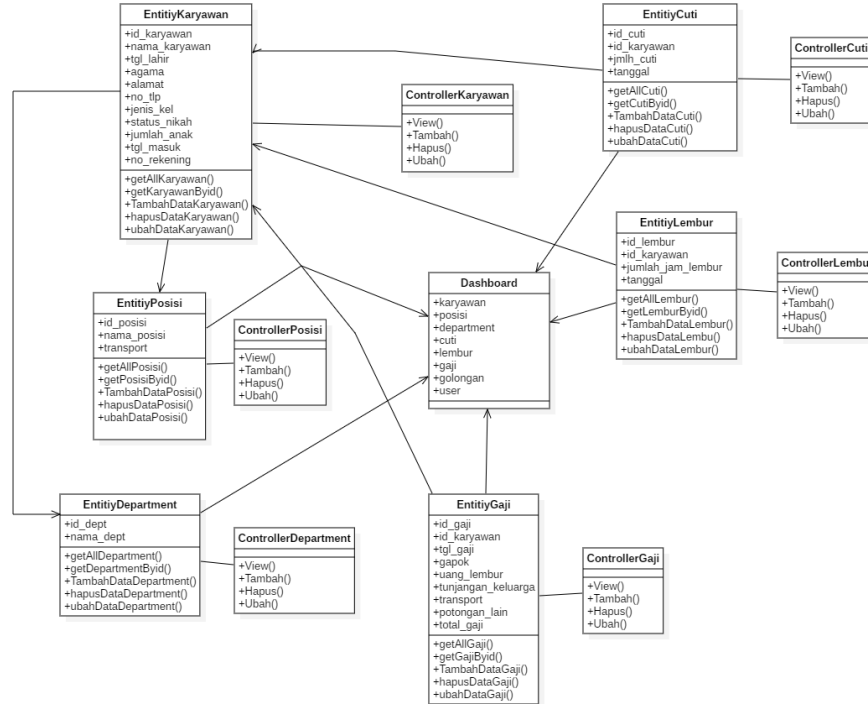


*Gambar 3.6. Realisasi UseCase U6*

Keterangan :

1. Admin HRD memilih menu data lembur untuk melakukan pengelolaan data yaitu : Add, Update, Delete.
2. Controller akan melakukan verifikasi sesuai dengan permintaan, contohnya seperti verifikasi tambah data apakah sudah sesuai.
3. Kemudian data masuk ke database data lembur.

### 3.2. Diagram Kelas Keseluruhan



Gambar 3.7. Diagram kelas Analisis

### 3.3. Kelas Analisis

Table 3.1. Kelas Analisis

No	Nama Kelas	Jenis
1	Login	Boundary
2	Cek_Login	Control
3	Dashboard	Boundary
4	Data_Karyawan	Boundary
5	Controller Karyawan	Control
6	Entitykaryawan	Entity
7	DataPosisi	Boundary
8	ControllerPosisi	Control
9	EntityPosisi	Entity
10	Datadepartment	Boundary
11	ControllerDepartment	Control
12	EntityDepartment	Entity
13	DataCuti	Boundary
14	ControllerCuti	Control
15	EntityCuti	Entity
16	DataLembur	Boundary



17	ControllerLembur	Control
18	EntityLembur	Entity
19	DataGaji	Boundary
20	ControllerGaji	Control
21	EntityGaji	Entity

### 3.4. Paket Analisis

#### 3.4.1. Identifikasi Paket Analisis

Table 3.2. Identifikasi Paket Analisis

No	Nama Paket	Use Case Terkait	Keterangan
1	Verifikasi User	1. Login	Paket yang digunakan untuk melakukan login ke dalam system dengan menginputkan username dan password.
2.	Pengelolaan Data penggajian	1. Lihat data cuti 2. Lihat data lembur 3. Lihat data karyawan 4. Kelola data gaji	Paket yang digunakan untuk mengelola data gaji setiap bulannya.
3	Pengelolaan data pendukung penggajian	1. Kelola Data Karyawan 2. Kelola data cuti 3. Kelola data lembur	Paket yang digunakan untuk mengelola data pendukung penggajian
4	Kelola Data Posisi	1. Kelola data posisi	Paket yang digunakan untuk mengelola data posisi
5	Kelola Data Department	1. Kelola data department	Paket yang digunakan untuk mengelola data department

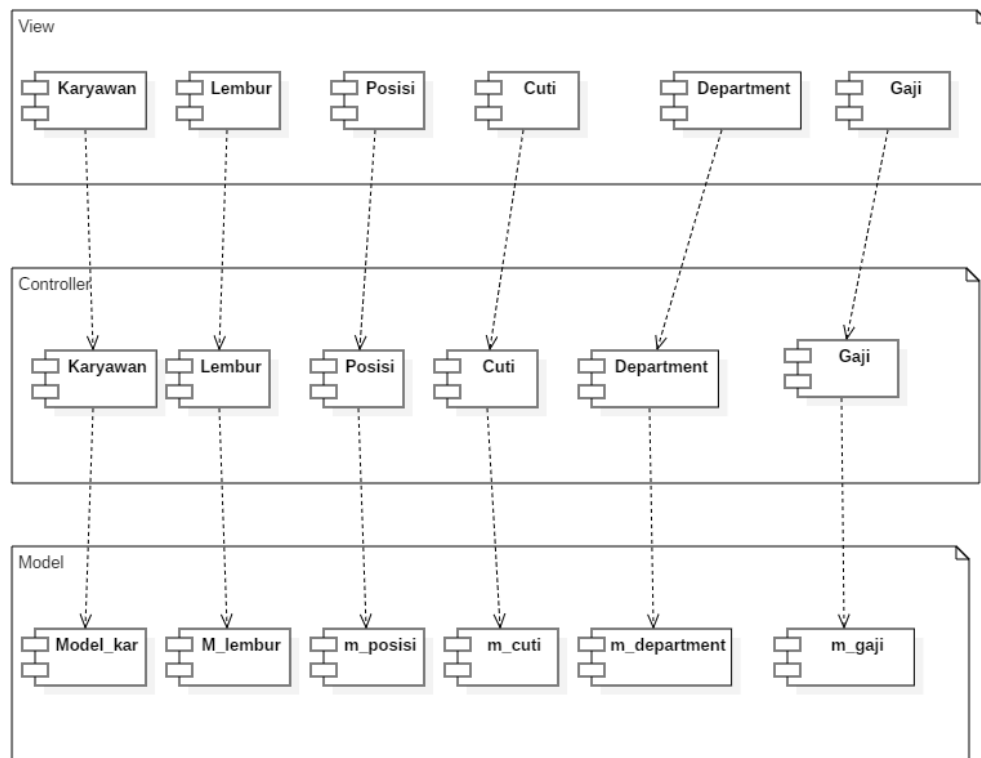
#### 3.4.2. Identifikasi Kelas Analisis tiap paket

Table 3.3. Identifikasi Kelas Analisis Tiap Paket

No	Nama Paket	Nama Kelas Analisis	Jenis Kelas (Boundary, Control, Entity)
1	Verifikasi Admin	1. Form login 2. Cek login 3. Proses verifikasi	1. Boundary 2. Control 3. Entity
2	Pengelolaan Data penggajian	1. Data cuti 2. Data lembur 3. Data karyawan	1. Boundary 2. Boundary 3. Boundary

		4. Data gaji 5. Controller gaji 6. Proses input, edit, hapus gaji	4. Boundary 5. Control 6. Entity
3	Pengelolaan data pendukung penggajian	1. Data karyawan 2. Data cuti 3. Data lembur 4. Form ubah dan tambah 5. Controller karyawan 6. Controller cuti 7. Controller lembur 8. Proses input, edit, hapus data	1. Boundary 2. Boundary 3. Boundary 4. Boundary 5. Control 6. Control 7. Control 8. Entity

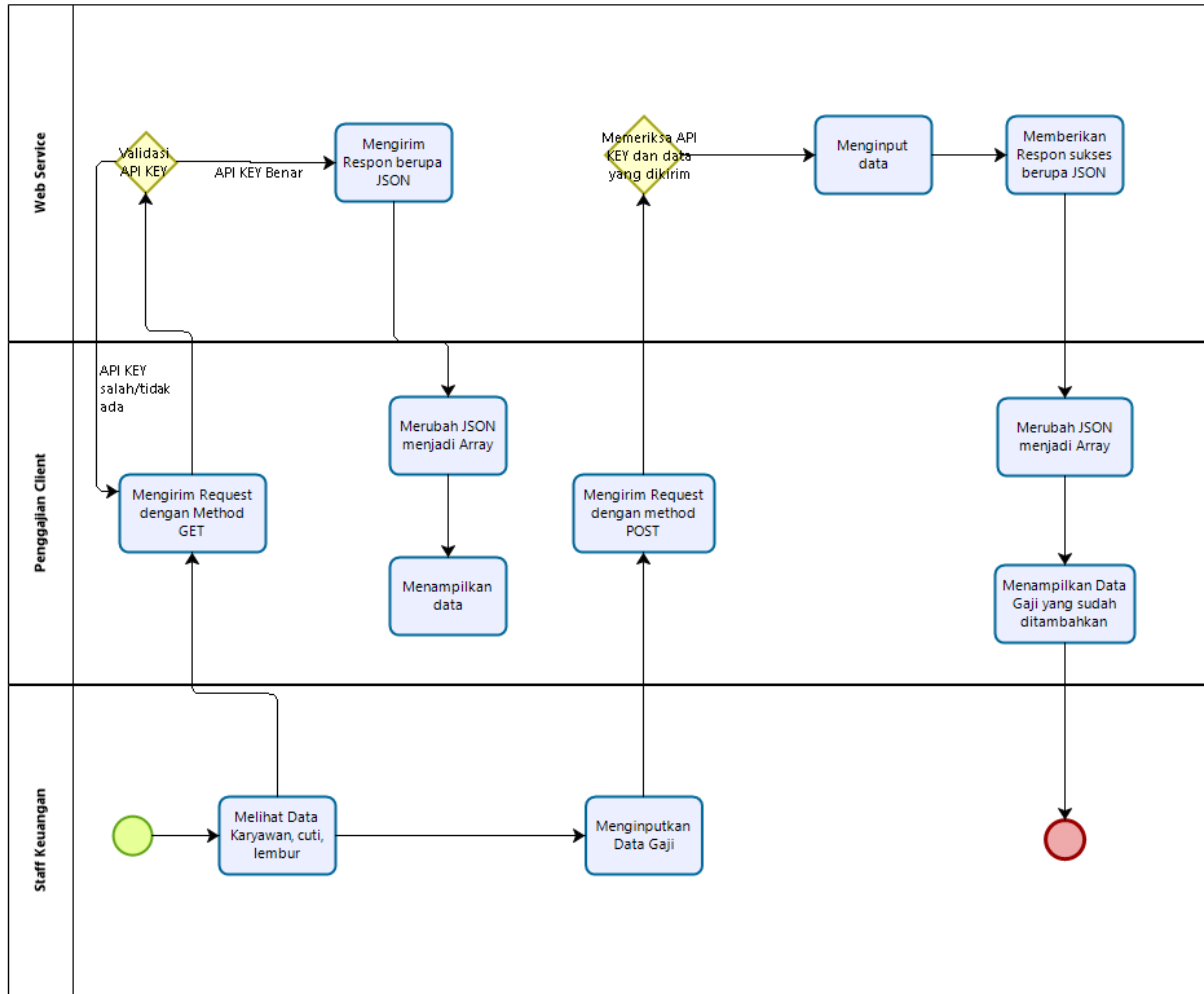
### 3.5.Deskripsi Arsitektur



Gambar 3.8. Arsitektur Sistem

Arsitektur Sistem Penggajian ini menggunakan architectural style layered dengan architectural pattern MVC. Dalam hal ini, Model merupakan layer terbawah yang berhubungan langsung dengan Web Service API Server, Controller bertanggung jawab

### 3.6. Proses Bisnis



### 3.7. Pedoman Perancangan

System informasi penggajian karyawan akan dibangun dengan sebuah framework dan Bahasa pemrograman sebagai berikut :

1. API akan dibangun dengan menggunakan Bahasa pemrograman PHP dan framework Codeigniter.
2. Aplikasi Sistem informasi Penggajian karyawan akan dibangun menggunakan Bahasa pemrograman PHP dan framework Codeigniter

## 4. Model Perancangan

### 4.1. Realisasi Use Case Tahap Perancangan

#### 4.1.1. Use Case Kelola Data Karyawan

##### 4.1.1.1. Identifikasi Elemen WAE – Logical View

*Table 4.1. identifikasi Elemen WAE Usecase Kelola Karyawan*

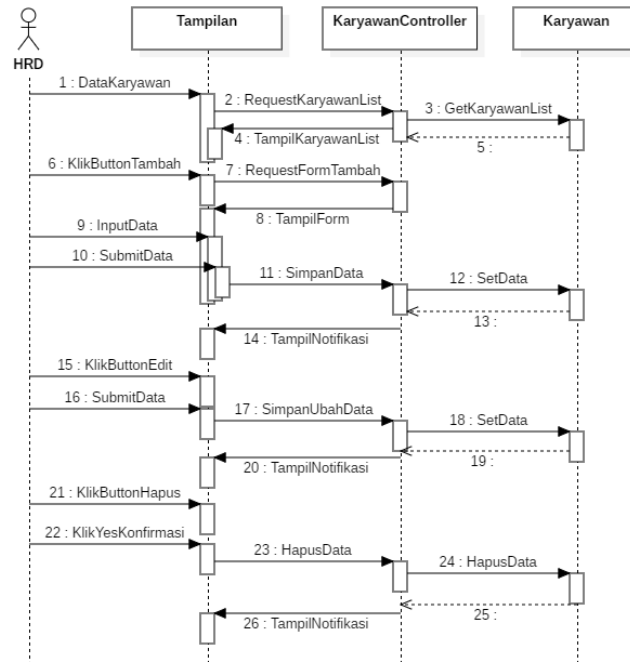
No	Nama Elemen Logik WAE	Stereotype
1	Karyawan Server	Server Page
2	Karyawan Client	Client Page
3	Tampil Karyawan Client	HTML Form
4	Edit Karyawan Client	HTML Form
5	Tambah Karyawan Client	HTML Form

##### 4.1.1.2. Identifikasi Kelas Perancangan

*Table 4.2. identifikasi Kelas Perancangan Karyawan*

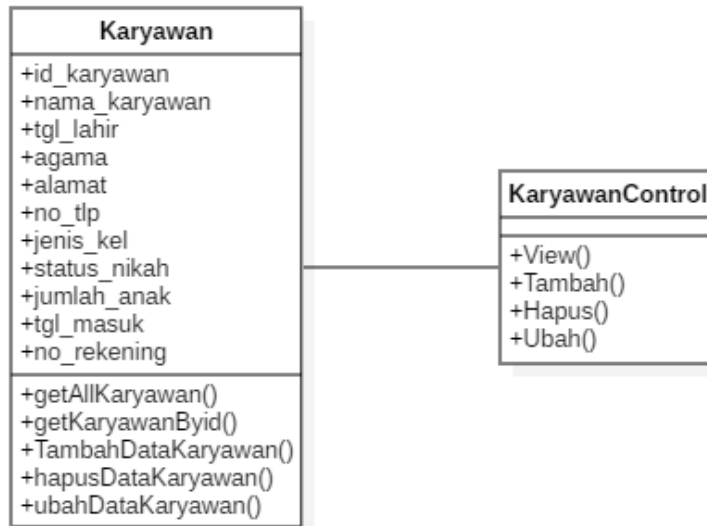
No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	Karyawan Control	Controller Karyawan
2	Karyawan	Entity Karyawan

#### 4.1.1.3. Sequence Diagram



Gambar 4.1. Squence Diagram Kelola Karyawan

#### 4.1.1.4. Diagram Kelas Perancangan



Gambar 4.2. Diagram Kelas Perancangan Karyawan

#### **4.1.2. Use Case Kelola Data Posisi**

##### **4.1.2.1. Identifikasi Elemen WAE – Logical View**

*Tabel 4.3. Identifikasi Elemen WAE Kelola Posisi*

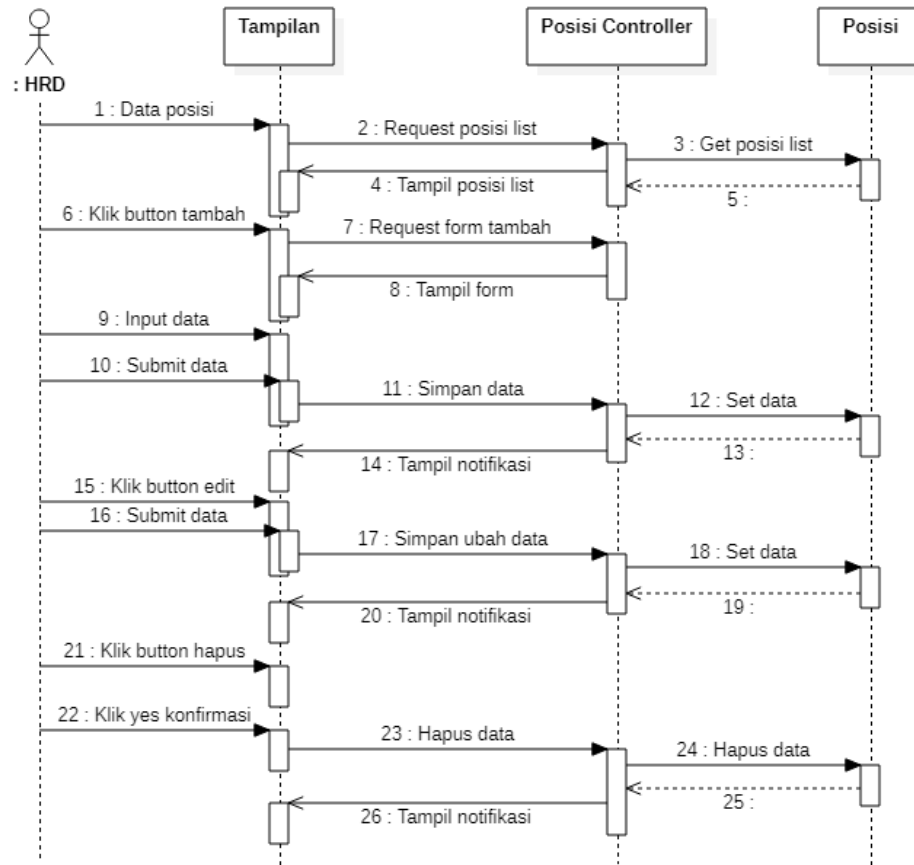
<b>No</b>	<b>Nama Elemen Logik WAE</b>	<b>Stereotype</b>
1	Posisi Server	Server Page
2	Posisi Client	Client Page
3	Tampil Posisi Client	HTML Form
4	Edit Posisi Client	HTML Form
5	Tambah Posisi Client	HTML Form

##### **4.1.2.2. Identifikasi Kelas Perancangan**

*Table 4.4. Identifikasi Kelas Perancangan Posisi*

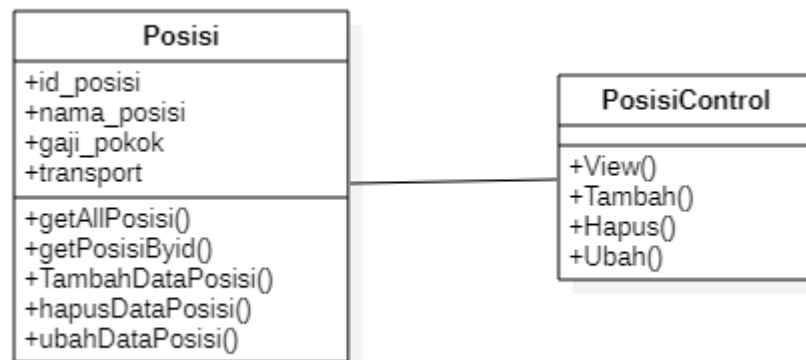
<b>No</b>	<b>Nama Kelas Perancangan</b>	<b>Nama Kelas Analisis Terkait</b>
1	Posisi Control	Controller Posisi
2	Posisi	Entity Posisi

#### 4.1.2.3. Sequence Diagram



Gambar 4.3. Sequence Diagram Kelola Posisi

#### 4.1.2.4. Diagram Kelas Perancangan



Gambar 4.4. Diagram Kelas Perancangan Posisi

#### **4.1.3. Use Case Kelola Data Department**

##### **4.1.3.1. Identifikasi Elemen WAE – Logical View**

*Table 4.5. Identifikasi Elemen WAE Kelola Department*

<b>No</b>	<b>Nama Elemen Logik WAE</b>	<b>Stereotype</b>
1	Department Server	Server Page
2	Department Client	Client Page
3	Tampil Department Client	HTML Form
4	Edit Department Client	HTML Form
5	Tambah Department Client	HTML Form

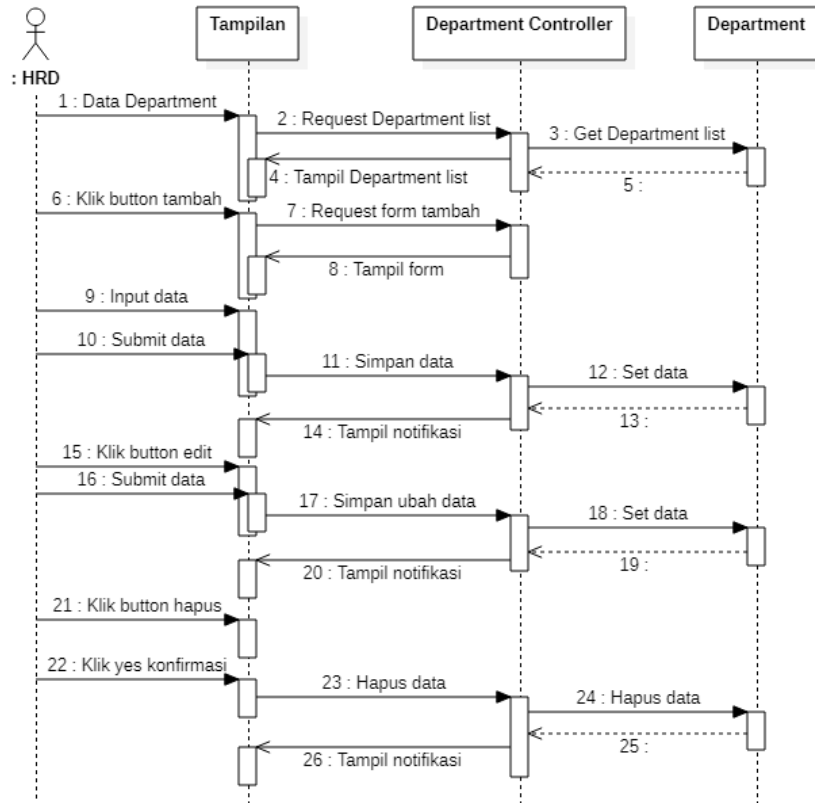
##### **4.1.3.2. Identifikasi Kelas Perancangan**

*Table 4.6. Identifikasi Kelas Perancangan Department*

<b>No</b>	<b>Nama Kelas Perancangan</b>	<b>Nama Kelas Analisis Terkait</b>
1	Department Control	Controller Department
2	Department	Entity Department

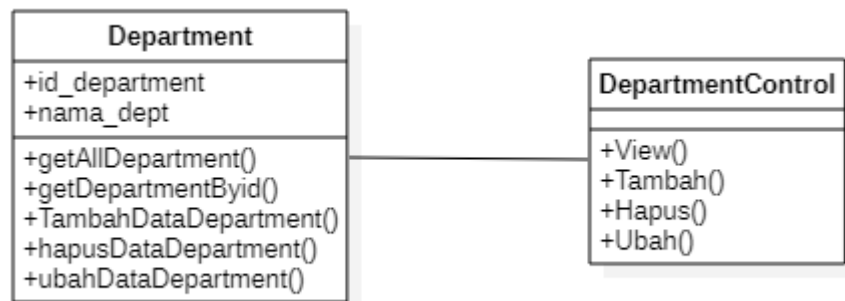
##### **4.1.3.3. Sequence Diagram**





Gambr 4.5. Squence Diagram Kelola Department

#### 4.1.3.4. Diagram Kelas Perancangan



Gambar 4.6. Diagram Kelas Perancangan Department

#### **4.1.4. Use Case Kelola Data cuti**

##### **4.1.4.1. Identifikasi Elemen WAE – Logical View**

*Table 4.7. Identifikasi Elemen WAE Kelola Cuti*

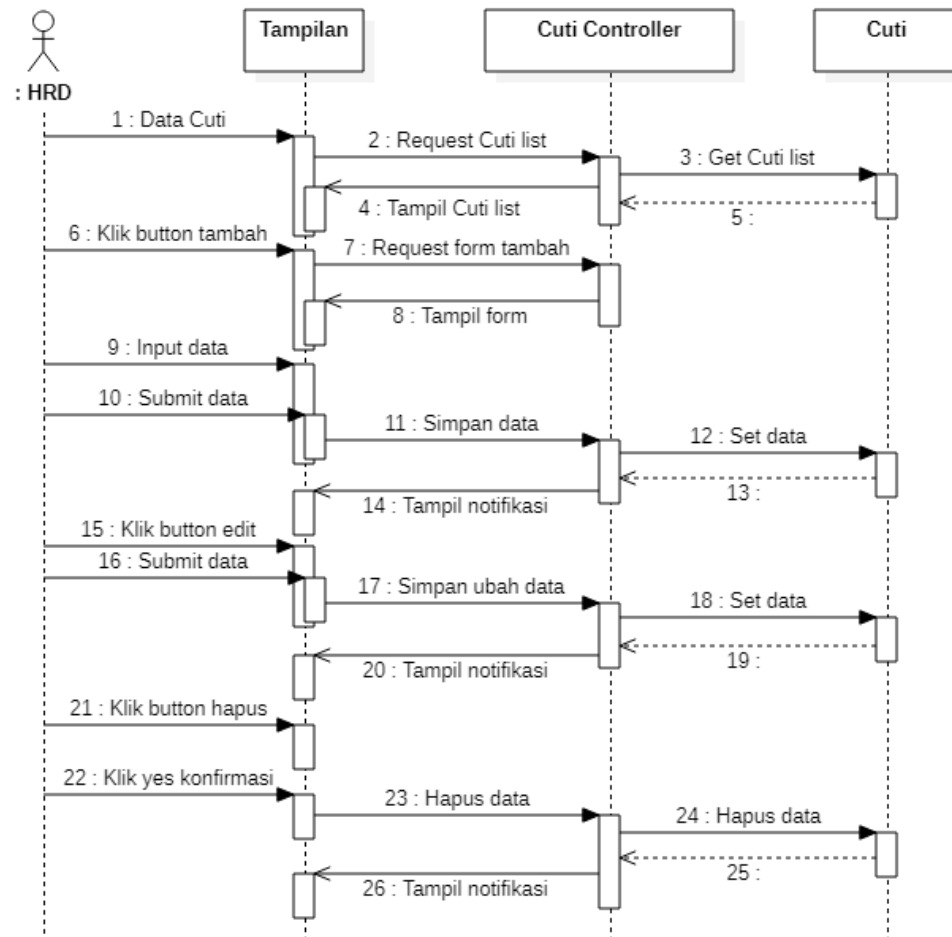
<b>No</b>	<b>Nama Elemen Logik WAE</b>	<b>Stereotype</b>
1	Cuti Server	Server Page
2	Cuti Client	Client Page
3	Tampil Cuti Client	HTML Form
4	Edit Cuti Client	HTML Form
5	Tambah Cuti Client	HTML Form

##### **4.1.4.2. Identifikasi Kelas Perancangan**

*Table 4.8. Identifikasi Kelas Perancangan Cuti*

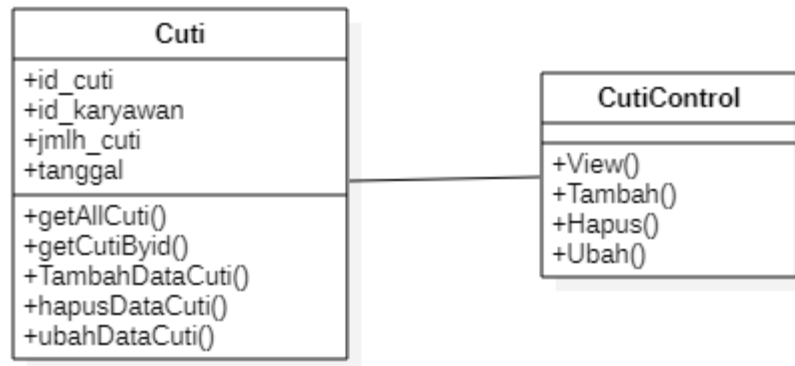
<b>No</b>	<b>Nama Kelas Perancangan</b>	<b>Nama Kelas Analisis Terkait</b>
1	Cuti Control	Controller Cuti
2	Cuti	Entity Cuti

#### 4.1.4.3. Sequence Diagram



Gambar 4.7. Squence Diagram Kelola Cuti

#### 4.1.4.4. Diagram Kelas Perancangan



Gambar 4.8. Diagram Kelas Perancangan Cuti

#### 4.1.5. Use Case Kelola Data Lembur

##### 4.1.5.1. Identifikasi Elemen WAE – Logical View

Table 4.9. Identifikasi Elemen WAE Kelola Lembur

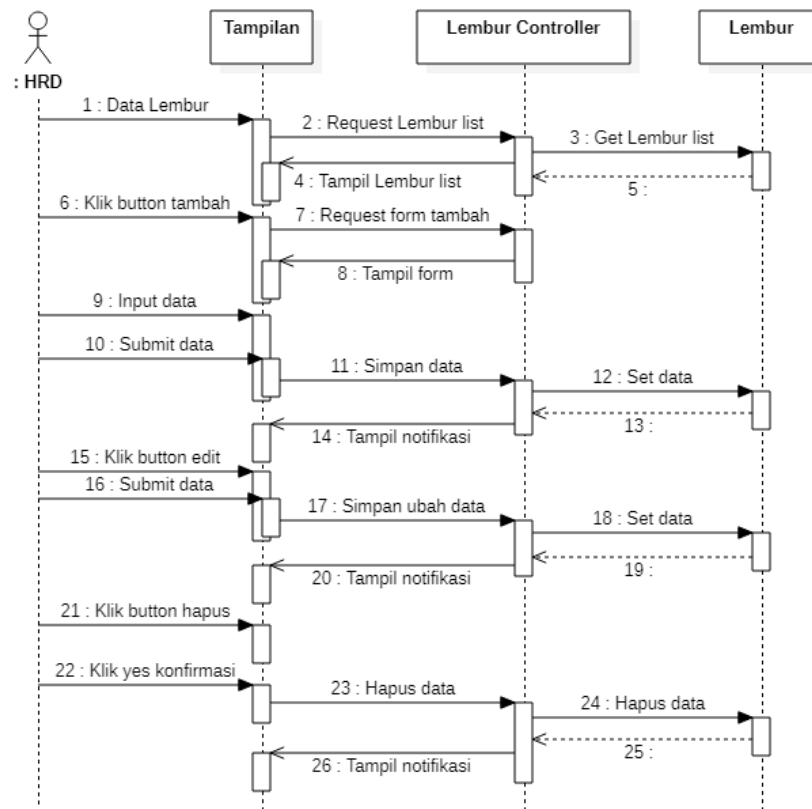
No	Nama Elemen Logik WAE	Stereotype
1	Lembur Server	Server Page
2	Lembur Client	Client Page
3	Tampil Lembur Client	HTML Form
4	Edit Lembur Client	HTML Form
5	Tambah Lembur Client	HTML Form

##### 4.1.5.2. Identifikasi Kelas Perancangan

Table 4.10. Identifikasi Kelas Perancangan Lembur

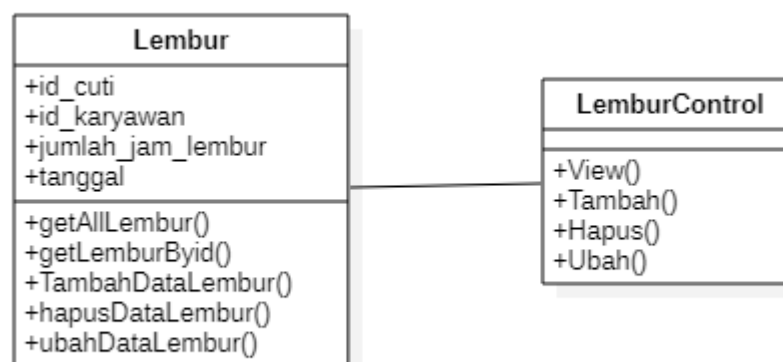
No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	Lembur Control	Controller Lembur
2	Lembur	Entity Lembur

#### 4.1.5.3. Sequence Diagram



Gambar 4.9. Squence Diagram Kelola Lembur

#### 4.1.5.4. Diagram Kelas Perancangan



Gambar 4.10. Diagram Kelas Perancangan Lembur

#### 4.1.6. Use Case Lihat Gaji

##### 4.1.6.1. Identifikasi Elemen WAE – Logical View

Table 4.11. Identifikasi Elemen WAE Lihat Gaji

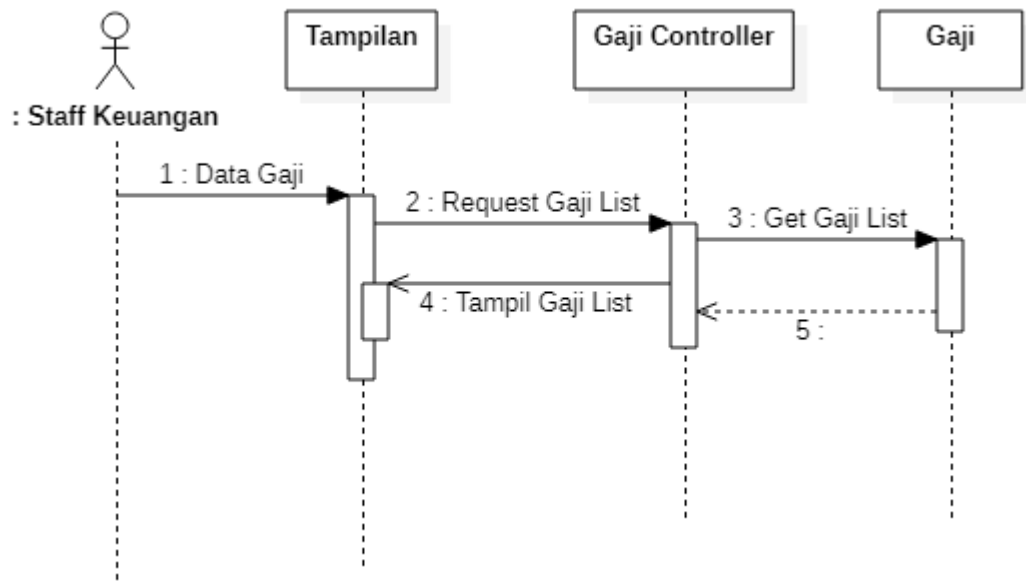
No	Nama Elemen Logik WAE	Stereotype
1	Tampil Gaji Client	Client Page

##### 4.1.6.2. Identifikasi Kelas Perancangan

Table 4.12. Identifikasi Kelas Perancangan Gaji

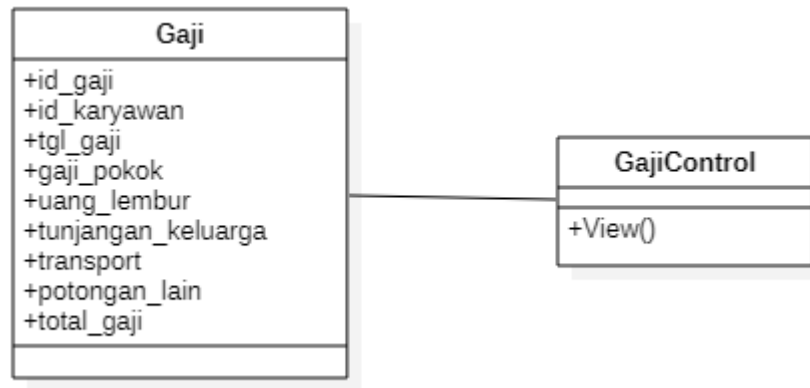
No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	Gaji Control	Controller Gaji
2	Gaji	Entity Gaji

##### 4.1.6.3. Sequence Diagram



Gambar 4.11. Squence Diagram Lihat Gaji

#### 4.1.6.4. Diagram Kelas Perancangan



Gambar 4.12. Diagram Kelas Perancangan Gaji

#### 4.1.7. Use Case Lihat Data Karyawan

##### 4.1.7.1. Identifikasi Elemen WAE – Logical View

Table 4.13. Identifikasi Elemen WAR Lihat Karyawan

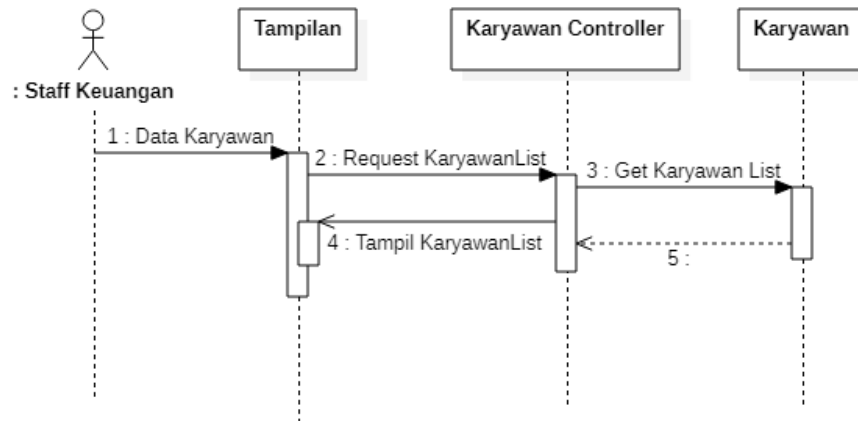
No	Nama Elemen Lojik WAE	Stereotype
1	Tampil Karyawan Client	Client Page

##### 4.1.7.2. Identifikasi Kelas Perancangan

Table 4.14. Identifikasi Kelas Perancangan Lihat Karyawan

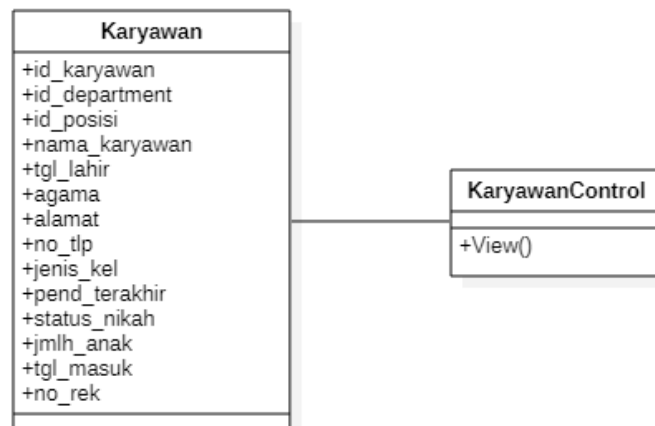
No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	Karyawan Control	Controller Karyawan
2	Karyawan	Entity Karyawan

##### 4.1.7.3. Sequence Diagram



Gambar 4.13. Squence Diagram Lihat Karyawan

#### 4.1.7.4. Diagram Kelas Perancangan



Gambar 4.14. Diagram Kelas Perancangan Lihat Karyawan

#### 4.1.8. Use Case Lihat Data Cuti

##### 4.1.8.1. Identifikasi Elemen WAE – Logical View

Table 4.15. Identifikasi Elemen WAE Lihat Cuti

No	Nama Elemen Lojik WAE	Stereotype
1	Tampil Cuti Client	Client Page

##### 4.1.8.2. Identifikasi Kelas Perancangan

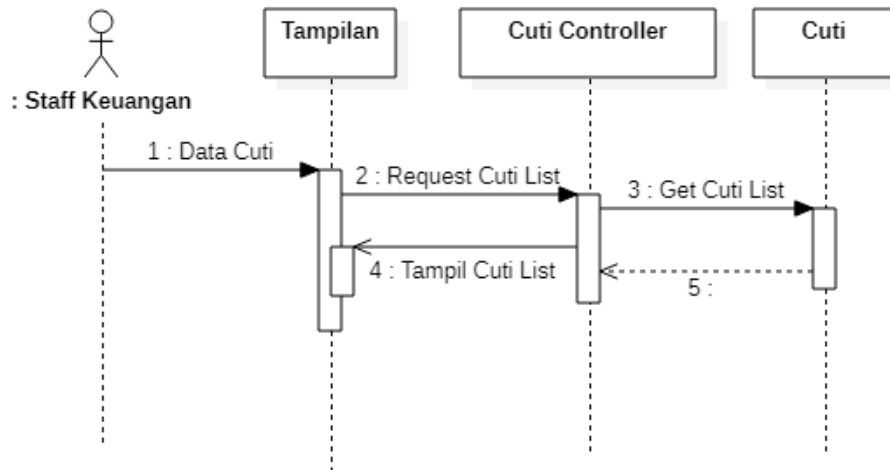
Table 4.16. Identifikasi Kelas Perancangan Lihat Cuti

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	Cuti Control	Controlller Cuti
2	Cuti	Entity Cuti



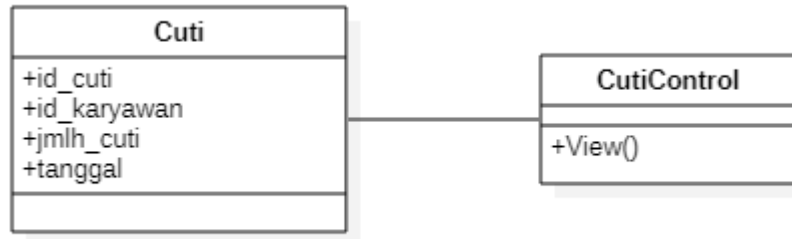
--	--	--

#### 4.1.8.3. Sequence Diagram



Gambar 4.15. Squence Diagram Lihat Cuti

#### 4.1.8.4. Diagram Kelas Perancangan



Gambar 4.16. Diagram Kelas Perancangan Lihat Cuti

#### 4.1.9. Use Case Lihat Data Lembur

##### 4.1.9.1. Identifikasi Elemen WAE – Logical View

Table 4.17. Identifikasi Elemen WAE Lihat Lembur

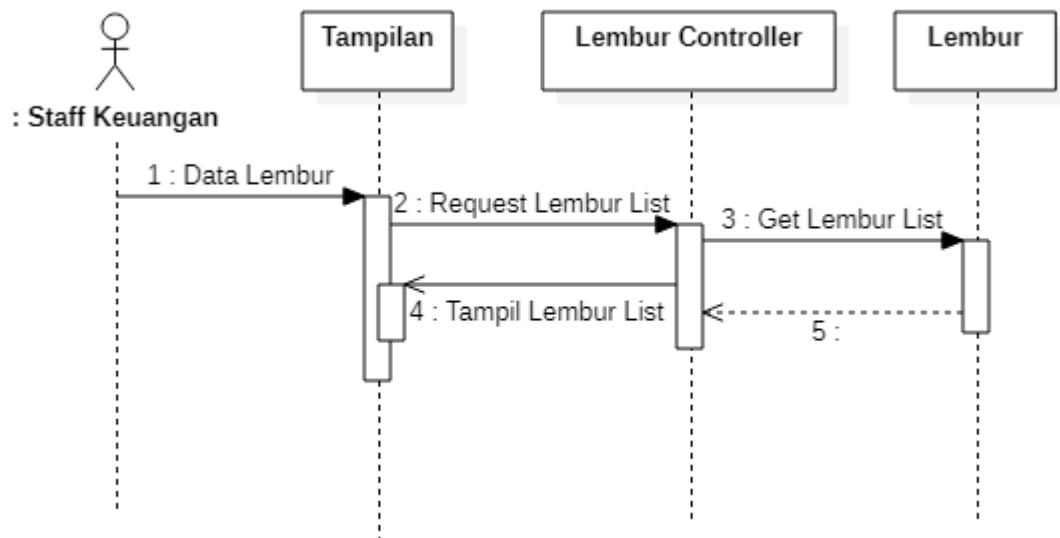
No	Nama Elemen Logik WAE	Stereotype
1	Tampil Lembur Client	Client Page

#### 4.1.9.2. Identifikasi Kelas Perancangan

Table 4.18. Identifikasi Kelas Perancangan Lihat Lembur

No	Nama Kelas Perancangan	Nama Kelas Analisis Terkait
1	Lembur Control	Controller Lembur
2	Lembur	Entity Lembur

#### 4.1.9.3. Sequence Diagram



Gambar 4.17. Sequence Diagram Lihat Lembur

#### 4.1.9.4. Diagram Kelas Perancangan



Gambar 4.18. Diagram Kelas Perancangan Lihat Lembur

#### **4.1.10. Use Case Kelola Data Gaji**

##### **4.1.10.1. Identifikasi Elemen WAE – Logical View**

*Table 4.19. Identifikasi Elemen WAE Kelola Gaji*

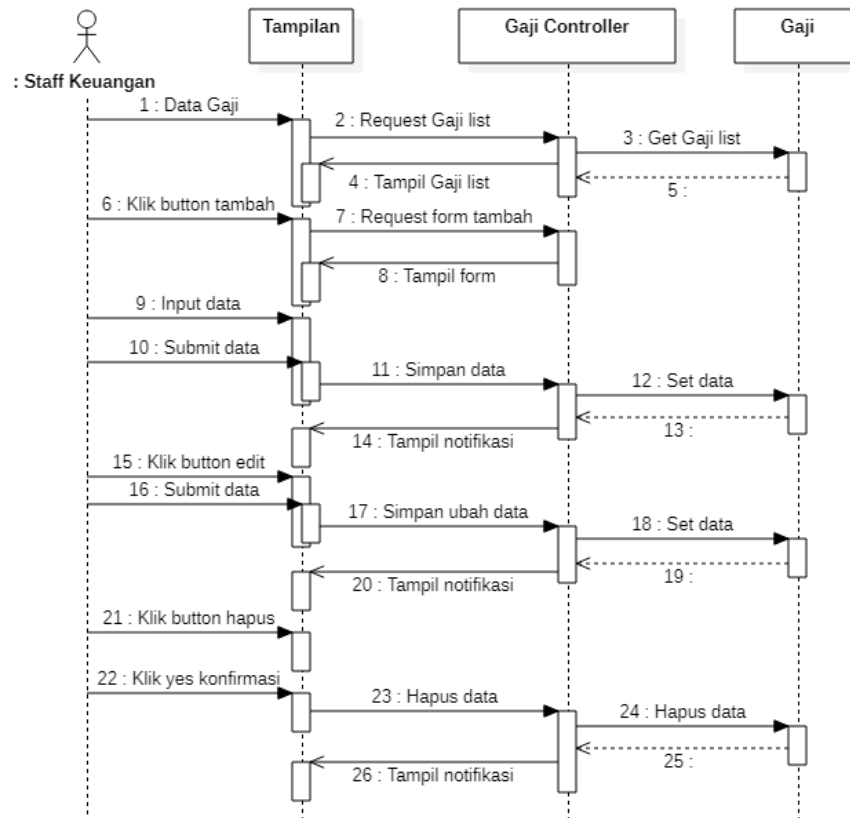
<b>No</b>	<b>Nama Elemen Logik WAE</b>	<b>Stereotype</b>
1	Gaji Server	Server Page
2	Gaji Client	Client Page
3	Tampil Gaji Client	HTML Page
4	Edit Gaji Client	HTML Form
5	Tambah Gaji Client	HTML Form

##### **4.1.10.2. Identifikasi Kelas Perancangan**

*Table 4.20. Identifikasi Kelas Perancangan Gaji*

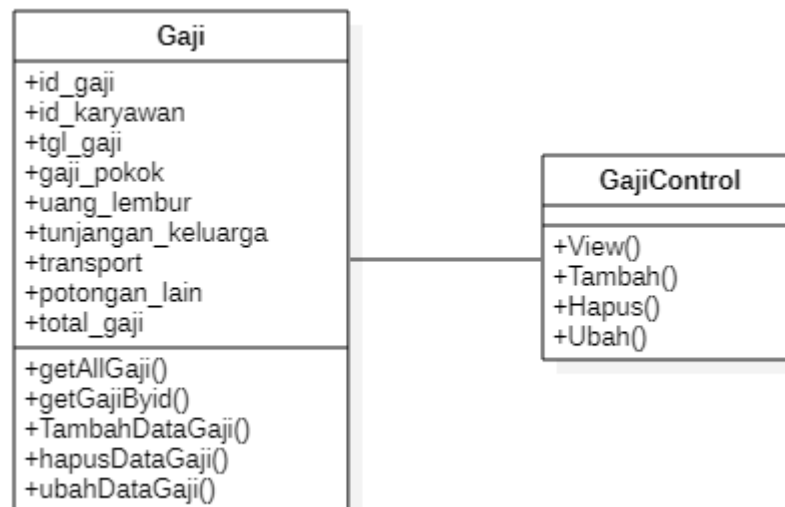
<b>No</b>	<b>Nama Kelas Perancangan</b>	<b>Nama Kelas Analisis Terkait</b>
1	Gaji Control	Controller Gaji
2	Gaji	Entity Gaji

##### **4.1.10.3. Sequence Diagram**



Gambar 4.19. Squence Diagram Kelola Gaji

#### 4.1.10.4. Diagram Kelas Perancangan



Gambar 4.20. Diagram Kelas Perancangan Gaji

## 4.2. Perancangan Detail Elemen Logical View

### 4.2.1. Stereotyped Class Karyawan Server

Nama Kelas : Karyawan Server

Jenis : Server Page

Table 4.21. Stereotyped Class Karyawan Server

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
Index get()	Public	Operasi untuk menampilkan data dari database
Index_post()	Public	
Index_put()	Public	
Index_delete()	Public	
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
-	-	-

### 4.2.2. Stereotyped Class Karyawan Client

Nama Kelas : Karyawan Client

Jenis : Client Page

Table 4.22. Stereotyped Class Karyawan Client

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
Index()	Public	
Tambah()	public	
Ubah()	public	
Hapus()	public	
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>

-	-	-
---	---	---

#### 4.2.3. Stereotyped Class *TampilKaryawanClient, TambahKaryawanClient, EditKaryawanClient*

Nama Kelas :Tampil Karyawan Client, TambahKaryawanClient, EditKaryawanClient  
Jenis : HTML Form

Table 4.23. Stereotyped Class *Tampil, Tambah, Edit Karyawan Client*

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
-	-	-
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
Id_karyawan		<b>Integer</b>
Nama_karyawan		<b>Vachar</b>
Tgl_lahir		<b>Vachar</b>
agama		<b>Vachar</b>
Alamat		<b>Vachar</b>
Jns_kelamin		<b>Vachar</b>
No_tlp		<b>Vachar</b>
Status_nikah		<b>Vachar</b>
Jumlah_anak		<b>Vachar</b>
Tgl_masuk		<b>Vachar</b>
No_rekening		<b>Vachar</b>

#### 4.2.4. Stereotyped Class *Posisi Server*

Nama Kelas : Posisi Server  
Jenis : Server Page

Table 4.24. Stereotyped Class *Posisi Server*

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
Index get()	Public	Operasi untuk menampilkan data dari database
Index_post()	Public	
Index_put()	Public	
Index_delete()	Public	
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>

-	-	-

#### 4.2.5. *Stereotyped Class Posisi Client*

Nama Kelas : Posisi Client

Jenis : Client Page

*Table 4.25. Stereotyped Class Posisi Client*

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
Index()	Public	
Tambah()	public	
Ubah()	public	
Hapus()	public	
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
-	-	-

#### 4.2.6. *Stereotyped Class TampilPosisiClient, TambahPosisiClient, EditPosisiClient.*

Nama Kelas :Tampil Posisi Client, Tambah Posisi Client, Edit Posisi Client

Jenis : HTML Form

*Table 4.26. Stereotyped Class Tampil, Tambah, Edit Posisi Client*

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
-	-	-
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
id_posisi		<b>Integer</b>
nama_posisi		<b>Vachar</b>
gaji_pokok		<b>Vachar</b>
transport		<b>Vachar</b>

### 4.3. Perancangan Detail Kelas

#### 4.3.1. Kelas *KaryawanControl*

Nama Kelas: *Karyawan Control*

Table 4.27. *Kelas KaryawanControl*

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
Index()	Public	Operasi untuk menampilkan list data karyawan
Tambah()	Public	Operasi untuk mengatur proses tambah data
Ubah()	Public	Operasi untuk mengatur proses ubah data
Hapus()	Public	Operasi untuk mengatur proses hapus data
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
-	-	-

#### 4.3.2. Kelas *Karyawan*

Nama Kelas: *Karyawan*

Table 4.28. *Kelas Karyawan*

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
getAllKaryawan()	Public	Operasi yang digunakan untuk menampilkan list data karyawan dari API Server
getAllKaryawanByid()	Public	Operasi yang digunakan untuk menampilkan list data karyawan dari API Server berdasarkan Id
tambahDataKaryawan ()	Public	Operasi yang digunakan untuk mengatur proses tambah data karyawan sesuai



		API Server
ubahDataKaryawan()	public	Operasi yang digunakan untuk mengatur proses ubah data karyawan sesuai API Server
HapusDataKaryawan()	Public	Operasi yang digunakan untuk mengatur proses hapus data karyawan sesuai API Server
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
Id_karyawan		<b>Integer</b>
Nama_karyawan		<b>Vachar</b>
Tgl_lahir		<b>Date</b>
agama		<b>Vachar</b>
Alamat		<b>Vachar</b>
Jns_kelamin		<b>Vachar</b>
No_tlp		<b>Vachar</b>
Status_nikah		<b>Vachar</b>
Jumlah_anak		<b>Vachar</b>
Tgl_masuk		<b>Vachar</b>
No_rekening		<b>Vachar</b>

#### 4.3.3. Kelas PosisiControl

Nama Kelas: Posisi Control

Table 4.29. Kelas PosisiControl

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
Index()	Public	Operasi untuk menampilkan list data posisi
Tambah()	Public	Operasi untuk mengatur proses tambah data
Ubah()	Public	Operasi untuk mengatur proses ubah data
Hapus()	Public	Operasi untuk mengatur proses hapus data

<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
-	-	-

#### 4.3.4. Kelas Posisi

Nama Kelas: Posisi

*Table 4.30. Kelas Posisi*

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
getAllPosisi()	Public	Operasi yang digunakan untuk menampilkan list data Posisi dari API Server
getAllPosisiById()	Public	Operasi yang digunakan untuk menampilkan list data Posisi dari API Server berdasarkan Id
tambahDataPosisi()	Public	Operasi yang digunakan untuk mengatur proses tambah data Posisi sesuai API Server
ubahDataPosisi()	public	Operasi yang digunakan untuk mengatur proses ubah data Posisi sesuai API Server
HapusDataPosisi ()	Public	Operasi yang digunakan untuk mengatur proses hapus data Posisi sesuai API Server
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
id_posisi		<b>Integer</b>
nama_posisi		<b>Vachar</b>
gaji_pokok		<b>Vachar</b>
transport		<b>Vachar</b>

#### 4.3.5. *Kelas DepartmentControl*

Nama Kelas: Department Control

*Table 4.31. Kelas DepartmentControl*

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
Index()	Public	Operasi untuk menampilkan list data Department
Tambah()	Public	Operasi untuk mengatur proses tambah data
Ubah()	Public	Operasi untuk mengatur proses ubah data
Hapus()	Public	Operasi untuk mengatur proses hapus data
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
-	-	-

#### 4.3.6. *KelasDepartment*

Nama Kelas: Department

*Table 4.32. Kelas Department*

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
getAllDepartment()	Public	Operasi yang digunakan untuk menampilkan list data Department dari API Server
getAllDepartmentById()	Public	Operasi yang digunakan untuk menampilkan list data Department dari API Server berdasarkan Id
tambahDataDepartment()	Public	Operasi yang digunakan untuk mengatur proses

		tambah data Department sesuai API Server
ubahDataDepartment()	public	Operasi yang digunakan untuk mengatur proses ubah data Department sesuai API Server
HapusDataDepartment()	Public	Operasi yang digunakan untuk mengatur proses hapus data Department sesuai API Server
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
id_posisi		<b>Integer</b>
nama_dept		<b>Vachar</b>

#### 4.3.7. Kelas GajiControl

Nama Kelas: Gaji Control

*Table 4.33. Kelas GajiControl*

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
Index()	Public	Operasi untuk menampilkan list data Gaji
Tambah()	Public	Operasi untuk mengatur proses tambah data
Ubah()	Public	Operasi untuk mengatur proses ubah data
Hapus()	Public	Operasi untuk mengatur proses hapus data
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
-	-	-

#### 4.3.8. *KelasGaji*

Nama Kelas: Gaji

Table 4.34. Kelas Gaji

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
getAllGaji()	Public	Operasi yang digunakan untuk menampilkan list data Gaji dari API Server
getAllGajiByid()	Public	Operasi yang digunakan untuk menampilkan list data Gaji dari API Server berdasarkan Id
tambahDataGaji()	Public	Operasi yang digunakan untuk mengatur proses tambah data Gaji sesuai API Server
ubahDataGaji()	public	Operasi yang digunakan untuk mengatur proses ubah data Gaji sesuai API Server
HapusDataGaji()	Public	Operasi yang digunakan untuk mengatur proses hapus data Gaji sesuai API Server
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
id_gaji		<b>Integer</b>
id_karyawan		<b>Integer</b>
tgl_gaji		<b>Date</b>
gaji_pokok		<b>Vachar</b>
uang_lembur		<b>Vachar</b>
tunjangan_keluarga		<b>Vachar</b>
transport		<b>Vachar</b>
potongan_lain		<b>Vachar</b>
total_gaji		<b>Vachar</b>

#### 4.3.9. *KelasCutiControl*

Nama Kelas: Cuti Control

Table 4.35. Kelas CutiControl

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
Index()	Public	Operasi untuk menampilkan list data Cuti
Tambah()	Public	Operasi untuk mengatur proses tambah data
Ubah()	Public	Operasi untuk mengatur proses ubah data
Hapus()	Public	Operasi untuk mengatur proses hapus data
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
-	-	-

#### 4.3.10. KelasCuti

Nama Kelas: Cuti

Table 4.36. Kelas Cuti

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
getAllCuti()	Public	Operasi yang digunakan untuk menampilkan list data Cuti dari API Server
getAllCutiByid()	Public	Operasi yang digunakan untuk menampilkan list data Cuti dari API Server berdasarkan Id
tambahDataCuti()	Public	Operasi yang digunakan untuk mengatur proses tambah data Cuti sesuai API Server
ubahDataCuti()	public	Operasi yang digunakan untuk mengatur proses ubah

		data Cuti sesuai API Server
HapusDataCuti()	Public	Operasi yang digunakan untuk mengatur proses hapus data Cuti sesuai API Server
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
id_cuti		<b>Integer</b>
id_karyawan		<b>Integer</b>
Jmlh_cuti		<b>Vachar</b>
tanggal		<b>Date</b>

#### 4.3.11. Kelas LemburControl

Nama Kelas: Lembur Control

Table 4.36. Kelas LemburControl

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
Index()	Public	Operasi untuk menampilkan list data Lembur
Tambah()	Public	Operasi untuk mengatur proses tambah data
Ubah()	Public	Operasi untuk mengatur proses ubah data
Hapus()	Public	Operasi untuk mengatur proses hapus data
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
-	-	-

#### 4.3.12. KelasLembur

Nama Kelas: Lembur

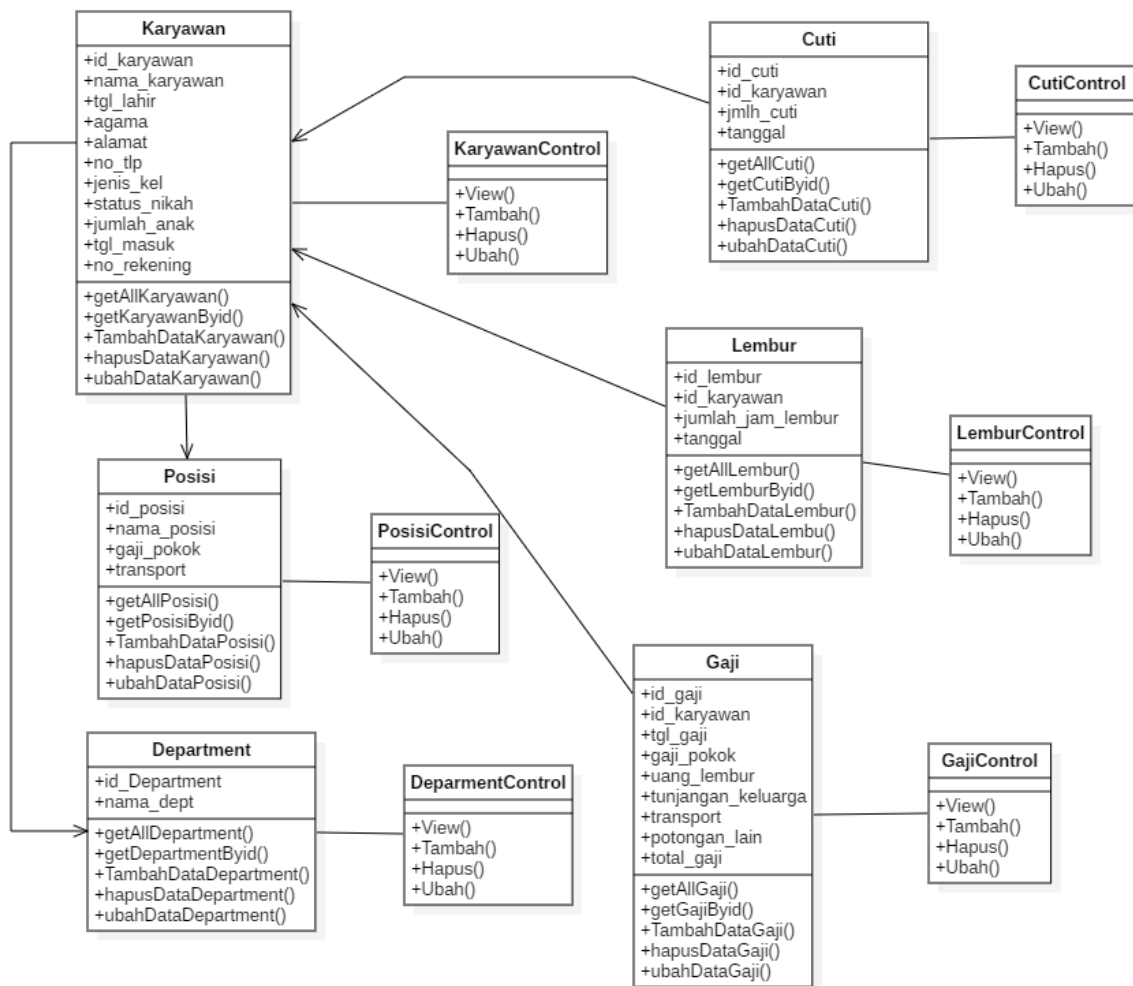
Table 4.37. Kelas Lembur

<b>Nama Operasi</b>	<b>Visibility ( Private, Public)</b>	<b>Keterangan</b>
---------------------	--	-------------------

getAllLembur()	Public	Operasi yang digunakan untuk menampilkan list data Lembur dari API Server
getAllLemburById()	Public	Operasi yang digunakan untuk menampilkan list data Lembur dari API Server berdasarkan Id
tambahDataLembur()	Public	Operasi yang digunakan untuk mengatur proses tambah data Lembur sesuai API Server
ubahDataLembur()	public	Operasi yang digunakan untuk mengatur proses ubah data Lembur sesuai API Server
HapusDataLembur()	Public	Operasi yang digunakan untuk mengatur proses hapus data Lembur sesuai API Server
<b>Nama Atribut</b>	<b>Visibility ( Private, Public)</b>	<b>Tipe</b>
id_lembur		<b>Integer</b>
id_karyawan		<b>Integer</b>
jmlh_jam_lembur		<b>Vachar</b>
tanggal		<b>Date</b>

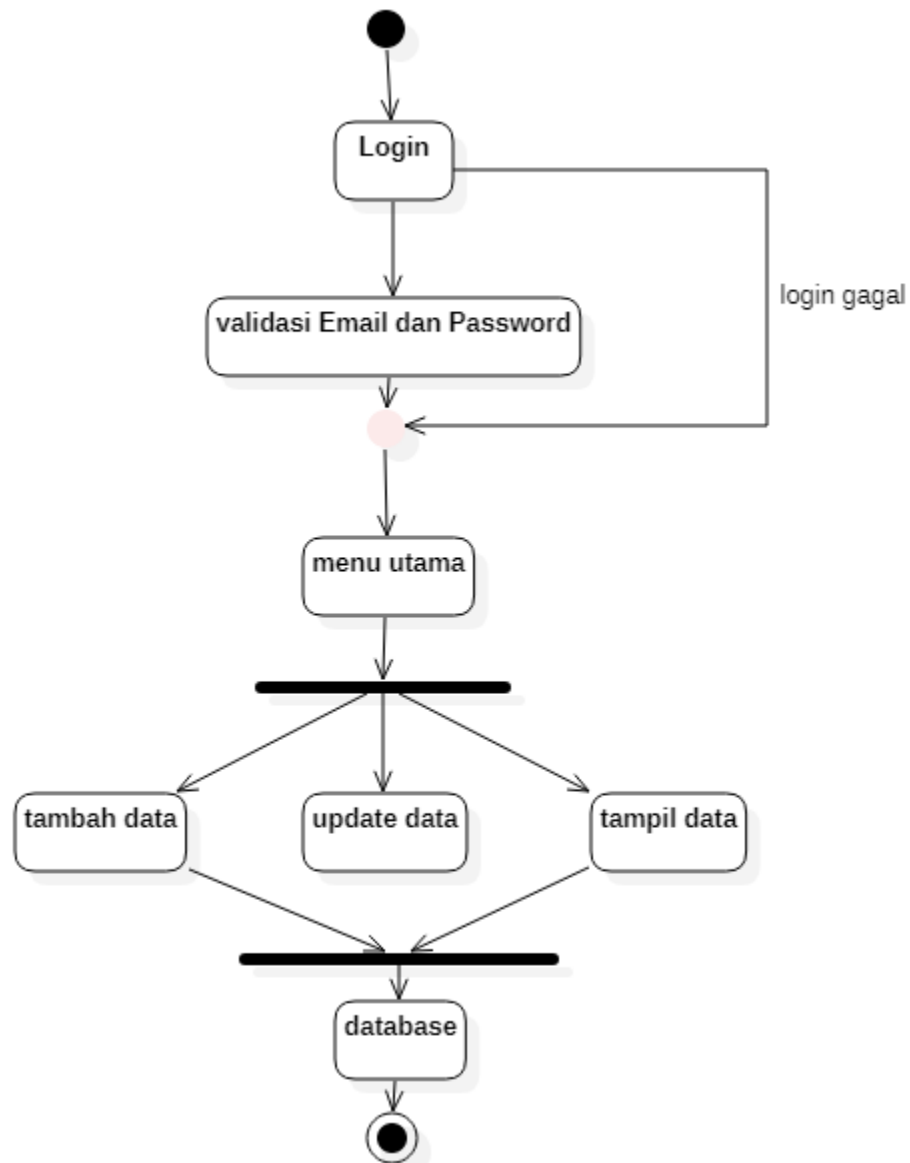
#### 4.4. Diagram Keseluruhan





Gambar 4.21. Diagram Kelas Keseluruhan Perancangan

#### 4.5. Diagram Statechart



Gambar 4.22. Diagram Statechart

## 4.6. Perancangan Antar Muka

### 4.6.1. Login

Antar muka :

The image shows a login form with a title 'Login' at the top. Below the title are two input fields: 'Email' and 'Password'. At the bottom of the form is a dark grey button labeled 'Sign in'.

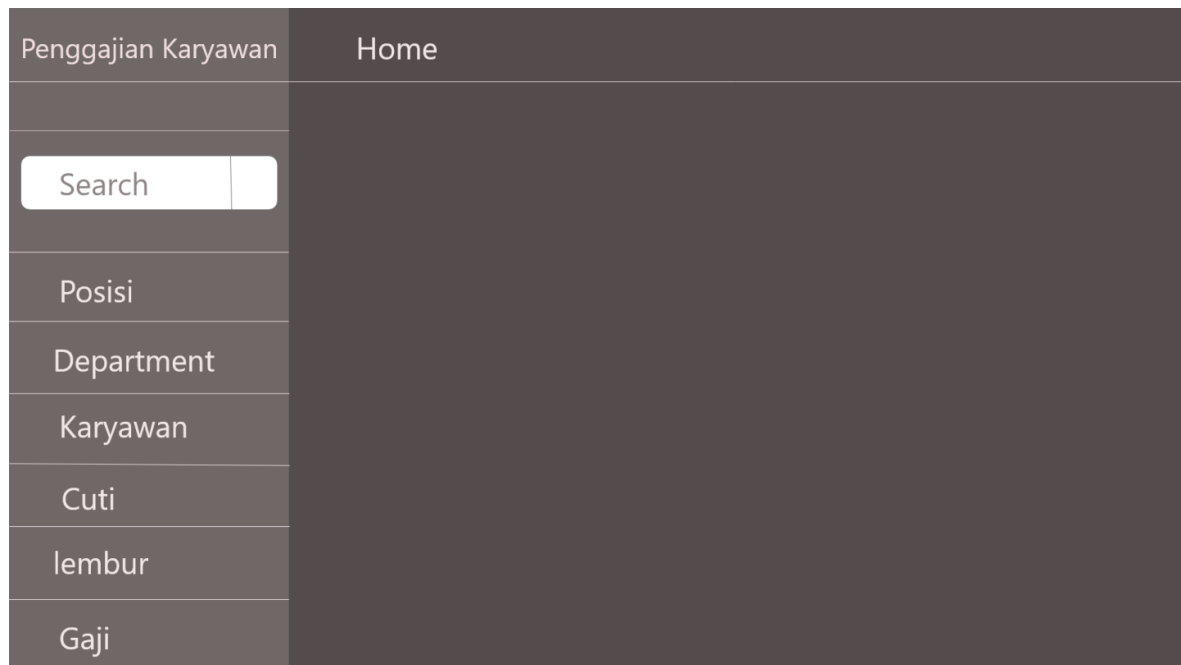
Gambar 4.23. Antar Muka Login

Table 4.38. AntarMuka Login

<i><b>Id_Object</b></i>	<i><b>Jenis</b></i>	<i><b>Nama</b></i>	<i><b>Keterangan</b></i>
		Form Login	Pada saat masuk ke web aplikasi maka akan tampil form login dan user akan diminta untuk memasukkan Email dan password agar system dapat berpindah ke layar selanjutnya dengan menekan button sign in.
Button1	Button	Sign in	Jika di klik akan mengaktifkan proses home
RTF1	RTF Box		Isi text yang disimpan pada file

#### 4.6.2. Home

Antarmuka :



Gambar 4.24. AntarMuka Home

Table 4.39. AntarMuka Home

<i><b>Id_Object</b></i>	<i><b>Jenis</b></i>	<i><b>Nama</b></i>	<i><b>Keterangan</b></i>
		Home	Setelah login maka system akan menampilkan home
Button1	Button	Search	Jika di klik akan mengaktifkan proses mencari data yang diinginkan oleh user
Button2	Button	Posisi	Jika di klik akan mengaktifkan proses menambah data posisi
Button3	Button	Department	Jika di klik akan mengaktifkan proses menambah data departement
Button4	Button	Karyawan	Jika di klik akan mengaktifkan proses menambah data karyawan
Button5	Button	Cuti	Jika di klik akan

			mengaktifkan proses menambah data cuti
Button6	Button	Lembur	Jika di klik akan mengaktifkan proses menambah data lembur
Button7	Button	Gaji	Jika di klik akan mengaktifkan proses menambah data gaji

#### 4.6.3. Form Tambah Data

Antarmuka :

Gambar 4.25. Antarmuka Form Tambah Data

Table 4.40. Antarmuka Form Tambah Data

<i><b>Id_Object</b></i>	<i><b>Jenis</b></i>	<i><b>Nama</b></i>	<i><b>Keterangan</b></i>
		Form tambah data pada posisi	System akan menampilkan form agar user dapat menambah data.
Button1	Button	irim	Jika di klik akan mengaktifkan proses form tampil data
RTF1	RTF Box		Isi text yang disimpan pada file

#### 4.6.4. Form Update Data

Antarmuka :

Gambar 4.26. AntarMuka Form Update Data

Table 4.41. AntarMuka Form Update Data

<i><b>Id_Object</b></i>	<i><b>Jenis</b></i>	<i><b>Nama</b></i>	<i><b>Keterangan</b></i>
		Form update data pada posisi	System akan menampilkan form agar user dapat mengupdate data yang sebelumnya telah disimpan.
Button1	Button	update	Jika di klik akan mengaktifkan proses form tampil data
RTF1	RTF Box		Isi text yang disimpan pada file

#### 4.6.5. Halaman Tampil Data

Antarmuka :

Penggajian Karyawan	Home				
	Tambah Data				
Search					
	ID	Nama Posisi	Gapok	Transport	Aksi
Posisi	123	HRD	Rp.3.000.000	Rp.600.000	Hapus Ubah Detail
Department					
Karyawan					
Cuti					
lembur					
Gaji					

Gambar 4.27. Antarmuka Tampil Data

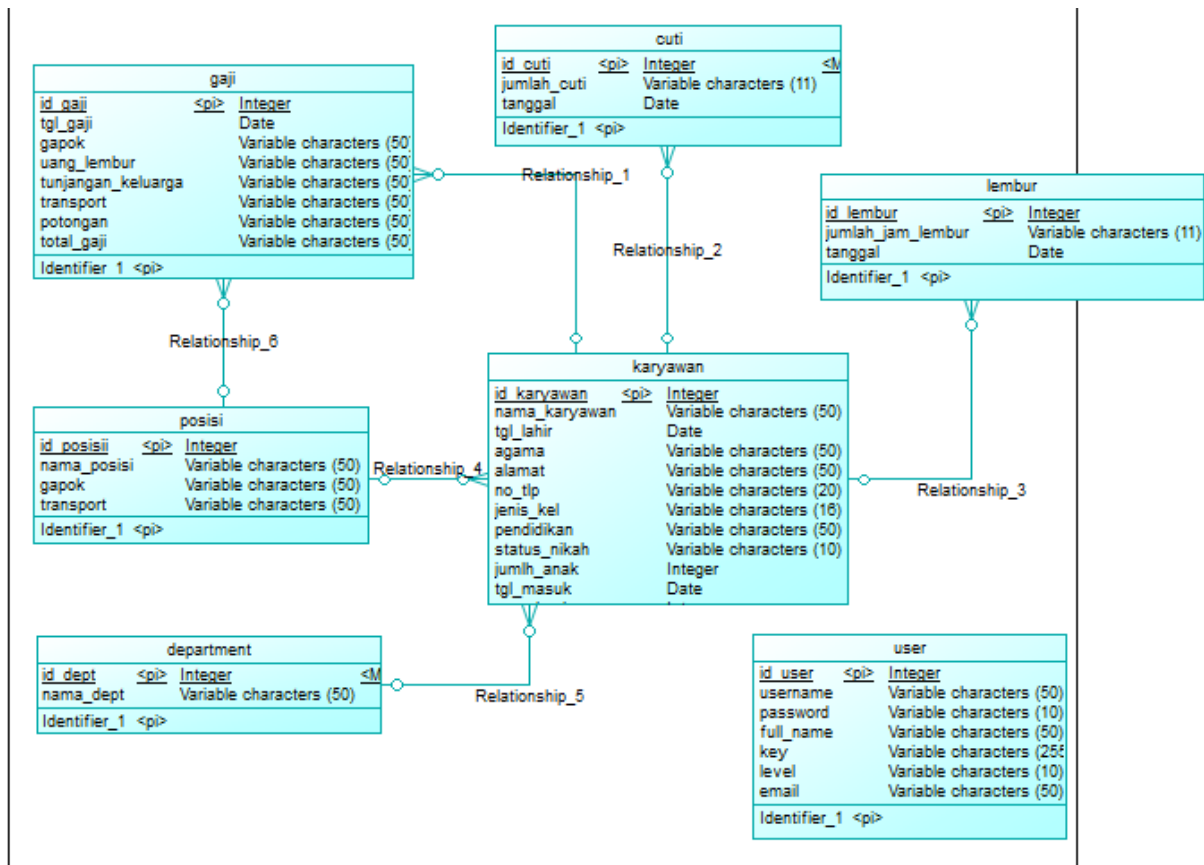
Table 4.42. Antarmuka Tampil Data

<i><b>Id_Object</b></i>	<i><b>Jenis</b></i>	<i><b>Nama</b></i>	<i><b>Keterangan</b></i>
		Form tampil data pada posisi	System akan menampilkan data yang sebelumnya telah disimpan.
Button1	Button	Tambah data	Jika di klik akan mengaktifkan proses form tambah data
RTF1	RTF Box		Isi text yang disimpan pada file

#### 4.7. Perancangan Representasi presistensi Kelas

Rancangan skema basisdata yang akan digunakan pada pembuatan system informasi penggajian ini adalah sebagai berikut :

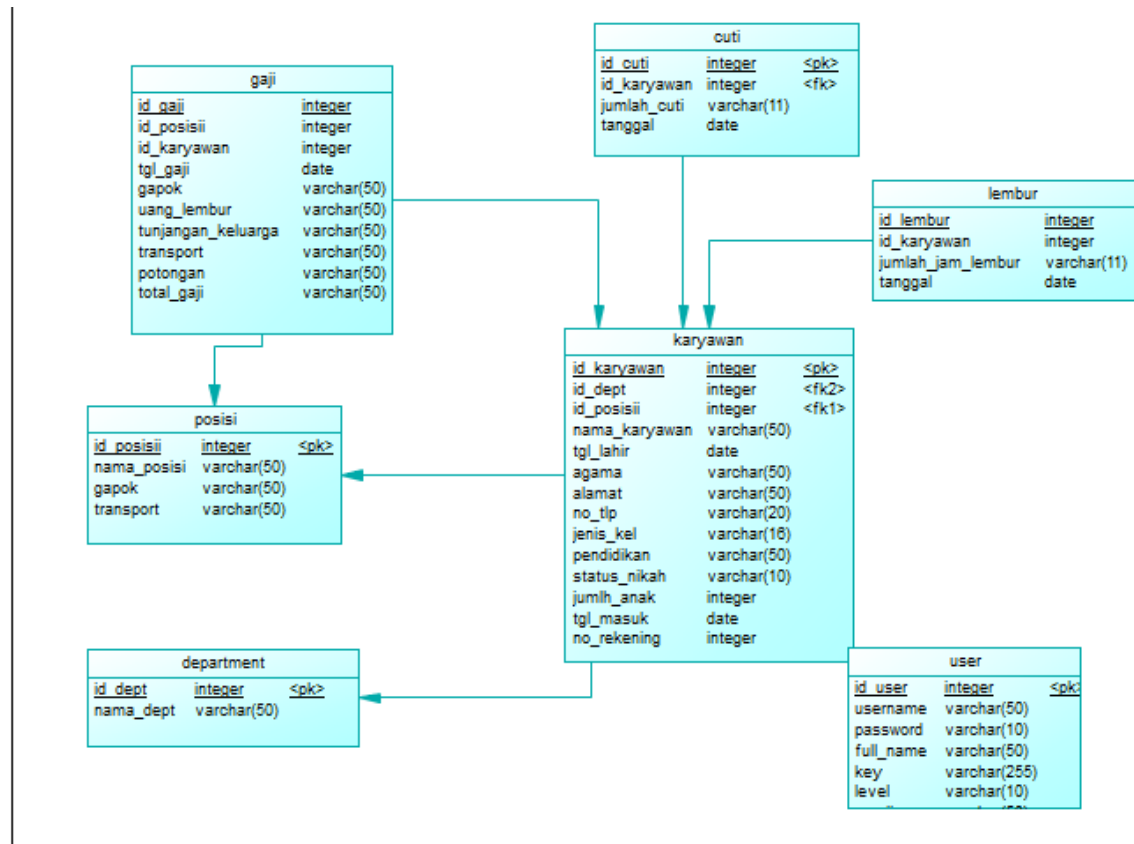
##### 4.7.1. Conceptual Data Model (CDM)



Gambar 4.28. Conceptual Data Model



#### 4.7.2. Physical Data Model (PDM)



Gambar 4.29. Physical Data Model

## 4.8. Coding standard dan Naming Convention

### 4.8.1. Coding Standard

Berikut adalah php coding standard sebagai berikut :

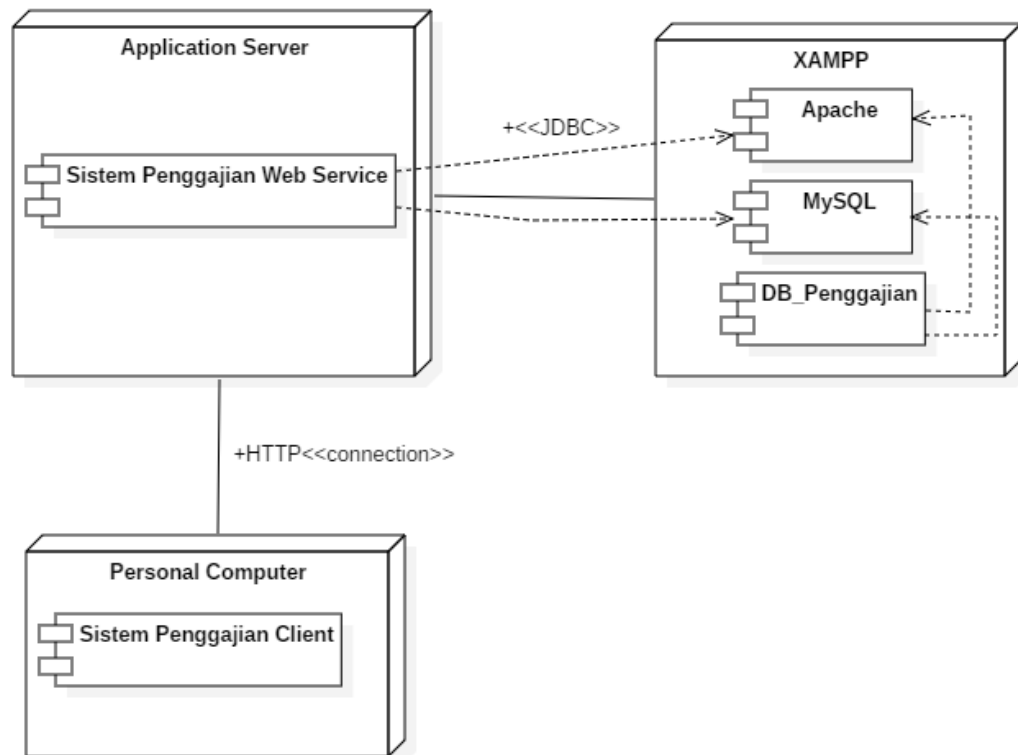
1. File menggunakan Tags PHP `<?PHP?>` atau `<? = ?>`
2. File harus menggunakan UTF-8
3. Penamaan Method menggunakan Camelcase
4. Pembuka brance (kurung kurawal) untuk class harus pada baris berikutnya dan penutup brance harus ada pada baris berikutnya setelah body class.
5. Membuka brance method harus pada baris berikutnya dan menutup brance harus pada baris berikutnya setelah body method.
6. Php constants harus lower case.

### 4.8.2. Naming Convention

*Table 4.43. Naming Convention*

Name	Convention
Class name	Lowercase
Method name	Camelcase
Variable name	Lowercase
Function name	Lowercase
Constans	Lowercase

#### 4.10. Deployment diagram



Gambar 4.30. Deployment Diagram

## 5. Implementasi

### 5.1.Lingkungan Implementasi

Aplikasi ini akan berjalan dengan baik dan sesuai tujuan karena didukung oleh perangkat lunak dan perangkat keras yang sesuai.

#### 5.1.1. Kebutuhan Perangkat Keras

Table 5.1. Kebutuhan Perangkat Keras

Perangkat Keras	Spesifikasi	Keterangan
Hardisk	320 GB	Media untuk menyimpan data aplikasi yang dibuat
Memory	3 GB	Memory System yang digunakan
Processor	Intel® core™ 2 duo CPU T6570 @2,10 Ghz	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer
Layar	LCD 14,1 Inchi	Media untuk menampilkan
Jaringan	SM, WCDMA, FDD-LTE	Untuk koneksi ke internet

#### 5.1.2. Kebutuhan Perangkat Lunak

Table 5.2. Kebutuhan Perangkat Lunak

Tools / Software	Fungsi	Keterangan
Windows 8 dan 10	System Operasi	-
Xampp	Server Basis Data	-
Web Browser	Browser	-
Web Api Penggajian	Aplikasi Server	-

### 5.2.Implementasi Kelas

Table 5.3. Implementasi Kelas

No	Nama Kelas	Nama File Fisik	Nama File Executable
1	Karyawan Control	Karyawan.php	-
2	Posisi Control	Posisi.php	-
3	Department Control	Department.php	-

4	Cuti Control	Cuti.php	-
5	Lembur Control	Lembur.php	-
6	Gaji Control	Gaji.php	-
7	Karyawan	Karyawan_m.php	-
8	Department	Department_m.php	-
9	Posisi	Posisi_m.php	-
10	Cuti	Cuti_m.php	-
11	Lembur	Lembur_m.php	-
12	Gaji	Gaji_m.php	-

### 5.3. Implementasi Elemen WAE ( Component View)

*Table 5.4. Implementasi Elemen WAE*

No	Nama File Fisik	Element Component View	UseCase	Elemen Logical View
1	Karyawan a. List b. Tambah c. Ubah	Dynamic page	Kelola data karyawan	a. Tampil karyawan client b. Tambah karyawan client c. Ubah karyawan client
2	Posisi a. List b. Tambah c. Ubah	Dynamic Page	Kelola data posisi	a. Tampil posisi client b. Tambah posisi client c. Ubah posisi client
3	Department a. List b. Tambah c. Ubah	Dynamic page	Kelola data department	a. Tampil department client b. Tambah

				department client c. Ubah department client
4	Cuti a. List b. Tambah c. Ubah	Dynamic page	Kelola data cuti	a. Tampil cuti client b. Tambah cuti client c. Ubah cuti client
5	Lembur a. List b. Tambah c. Ubah	Dynamic page	Kelola data lembur	a. Tampil lembur client b. Tambah lembur client c. Ubah lembur client
6	Gaji a. List b. Tambah c. Ubah	Dynamic page	Kelola data gaji	a. Tampil l gaji client b. Tambah gaji client c. Ubah gaji client

#### 5.4.Impelementasi Antarmuka

*Table 5.5. Implementasu Antarmuka*

No	Antarmuka	Nama File Fisik	Nama File Executable
1	Login	Login.php	-
2	Home	Menu.php	-
3	Form Tambah Data	Tambah.php	-
6	Form Update Data	Ubah.php	-
5	Halaman Tampil Data	List.php	-

#### 5.5.File Lain

## 6. Pengujian

### 6.1. Rencana dan Prosedur Pengujian

#### 6.1.1. Rencana Pengujian

Table 6.1. Rencana Pengujian

No	Use Case	Pengujian	Jenis Pengujian	Identifikasi
1	Login	1. Memasukkan username dan password	Black Box	
2	Kelola Data Karyawan	1. Menambahkan data karyawan 2. Mengedit data karyawan 3. Menghapus data karyawan	Black Box	
3	Kelola Data Posisi	1. Menambahkan data posisi 2. Mengedit data posisi 3. Menghapus data posisi	Black Box	
4	Kelola Data Department	1. Menambahkan data department 2. Mengedit data department 3. Menghapus data department	Black Box	
5	Kelola Data Cuti	1. Menambahkan data cuti 2. Mengedit data cuti 3. Menghapus data cuti	Black Box	
6	Kelola Data Lembur	1. Menambahkan	Black Box	

		data lembur 2. Mengedit data lembur 3. Menghapus data lembur		
7	Kelola Data Gaji	1. Menambahkan data gaji 2. Mengedit data gaji 3. Menghapus data gaji	Black Box	

## 6.2. Kasus Uji

### 6.2.1. Pengujian Use Case Login

*Table 6.2. Pengujian Use Case Login*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
Login	Memasukkan username dan password kemudian klik tombol submit	jika data cocok akan masuk ke dashboar sesuai dengan hak akses	<b>Berhasil</b>

### 6.2.2. Pengujian Use Case Kelola Data Karyawan

*Table 6.3. Pengujian Use Case Kelola Karyawan*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
Kelola data karyawan	1. Klik tombol tambah, kemudian isikan form dengan data lalu klik button kirim 2. Klik tombol edit, mengubah data lalu klik tombol update 3. Klik tombol hapus,	1. Menampilkan data yang sudah ditambahkan. 2. Menampilkan data yang sudah di edit.	<b>Berhasil</b>



	kemudian klik yes untuk konfirmasi.		
--	---	--	--

### 6.2.3. Pengujian Use Case Kelola Data Posisi

*Table 6.4. Pengujian Use Case Kelola Posisi*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
Kelola data Posisi	<ol style="list-style-type: none"> <li>Klik tombol tambah, kemudian isikan form dengan data lalu klik button kirim</li> <li>Klik tombol edit, mengubah data lalu klik tombol update</li> <li>Klik tombol hapus, kemudian klik yes untuk konfirmasi.</li> </ol>	<ol style="list-style-type: none"> <li>Menampilkan data yang sudah ditambahkan.</li> <li>Menampilkan data yang sudah di edit.</li> </ol>	<b>Berhasil</b>

### 6.2.4. Pengujian Use Case Kelola Data Department

*Table 6.5. Pengujian Use Case Kelola Department*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
Kelola data department	<ol style="list-style-type: none"> <li>Klik tombol tambah, kemudian isikan form dengan data lalu klik button kirim</li> <li>Klik tombol edit, mengubah data lalu klik</li> </ol>	<ol style="list-style-type: none"> <li>Menampilkan data yang sudah ditambahkan.</li> <li>Menampilkan data yang sudah di edit.</li> </ol>	<b>Berhasil</b>

	tombol update 3. Klik tombol hapus, kemudian klik yes untuk konfirmasi.		
--	--	--	--

#### 6.2.5. Pengujian Use Case Kelola Data Cuti

*Table 6.6. Pengujian Use Case Kelola Cuti*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
Kelola data cuti	1. Klik tombol tambah, kemudian isikan form dengan data lalu klik button kirim 2. Klik tombol edit, mengubah data lalu klik tombol update 3. Klik tombol hapus, kemudian klik yes untuk konfirmasi.	1. Menampilkan data yang sudah ditambahkan. 2. Menampilkan data yang sudah di edit.	<b>Berhasil</b>

#### 6.2.6. Pengujian Use Case Kelola Data Lembur

*Table 6.7. Pengujian Use Case Kelola Lembur*

Kelas Uji	Skenario Uji	Hasil yang diharapkan	Kesimpulan
Kelola data lembur	1. Klik tombol tambah, kemudian isikan form dengan data lalu klik button kirim	1. Menampilkan data yang sudah ditambahkan. 2. Menampilkan data yang sudah di edit.	<b>Berhasil</b>

	<ol style="list-style-type: none"> <li>2. Klik tombol edit, mengubah data lalu klik tombol update</li> <li>3. Klik tombol hapus, kemudian klik yes untuk konfirmasi.</li> </ol>		
--	---	--	--

### 6.2.7. Pengujian Use Case Kelola Data Gaji

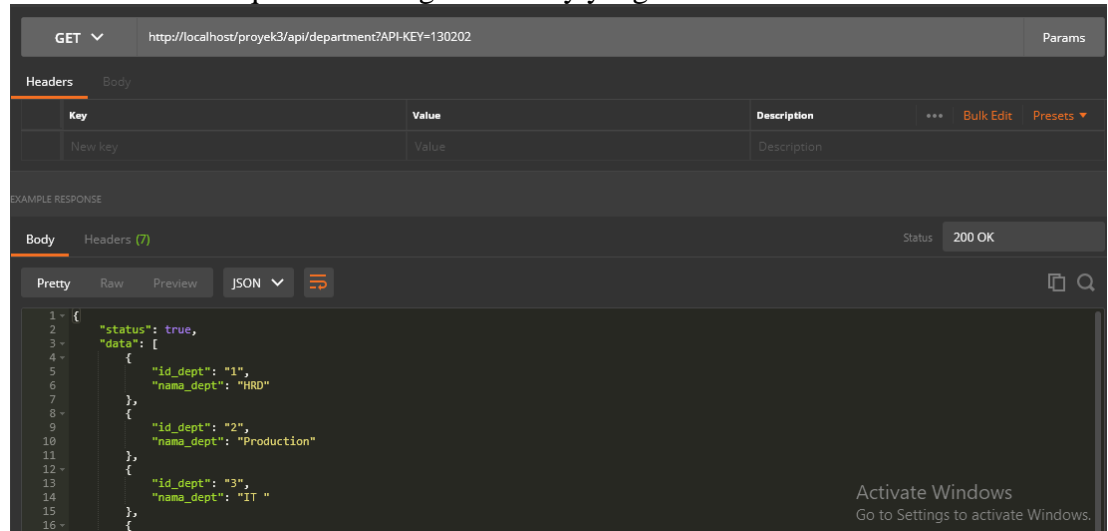
*Table 6.8. Pengujian Use Case Kelola Gaji*

<b>Kelas Uji</b>	<b>Skenario Uji</b>	<b>Hasil yang diharapkan</b>	<b>Kesimpulan</b>
Kelola data gaji	<ol style="list-style-type: none"> <li>1. Klik tombol tambah, kemudian isikan form dengan data lalu klik button kirim</li> <li>2. Klik tombol edit, mengubah data lalu klik tombol update</li> <li>3. Klik tombol hapus, kemudian klik yes untuk konfirmasi.</li> </ol>	<ol style="list-style-type: none"> <li>1. Menampilkan data yang sudah ditambahkan.</li> <li>2. Menampilkan data yang sudah di edit.</li> </ol>	<b>Berhasil</b>

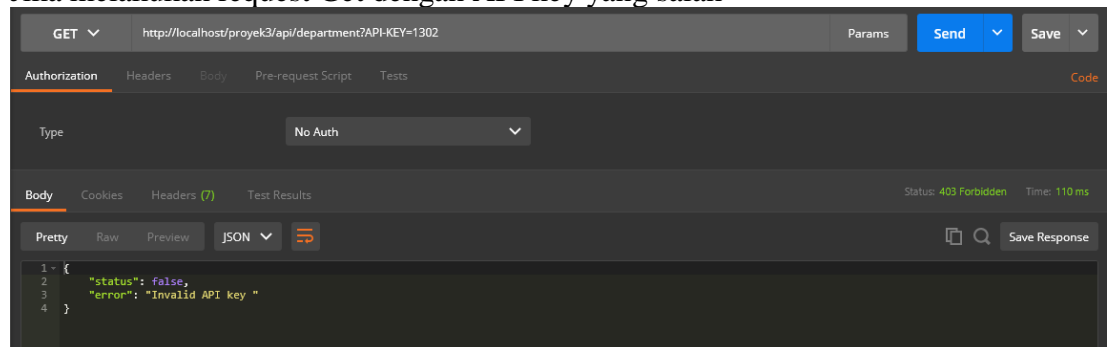
## 6.2.8. Pengujian Fungsi Web Service

### 6.2.8.1. Pengujian Fungsi GET

1. Jika melakukan request Get dengan API key yang benar

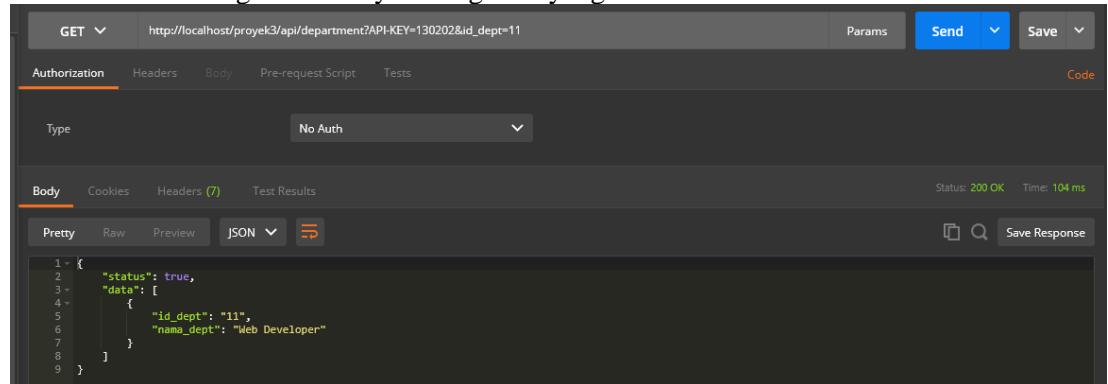


2. Jika melakukan request Get dengan API key yang salah

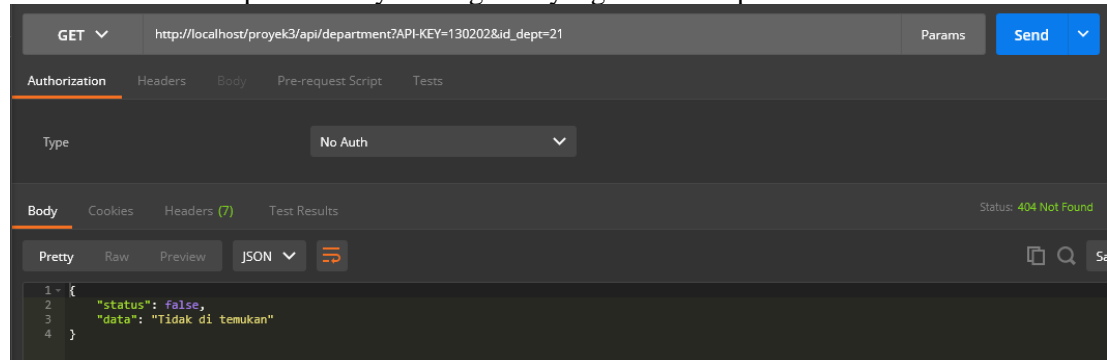


### 6.2.8.2. Pengujian Fungsi Get By id

1. Jika Melakukan Reguest Get By id dengan id yang ada di database

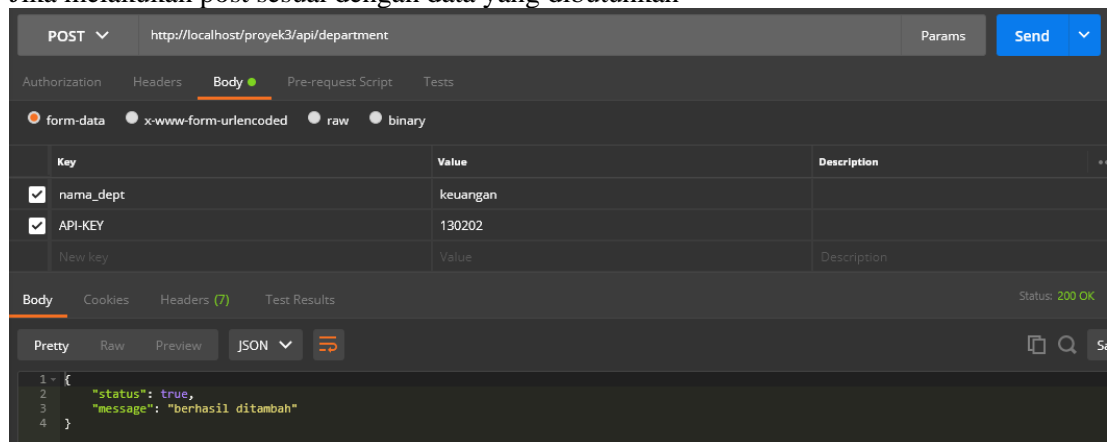


2. Jika Melakukan Request Get By id dengan id yang tidak terdapat di database

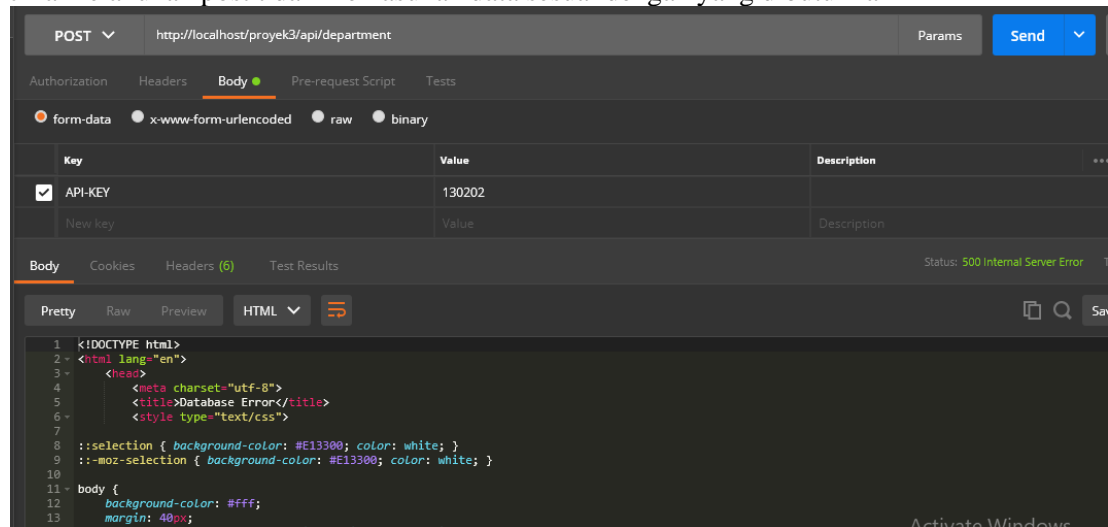


### 6.2.8.3. Pengujian Fungsi Post

1. Jika melakukan post sesuai dengan data yang dibutuhkan

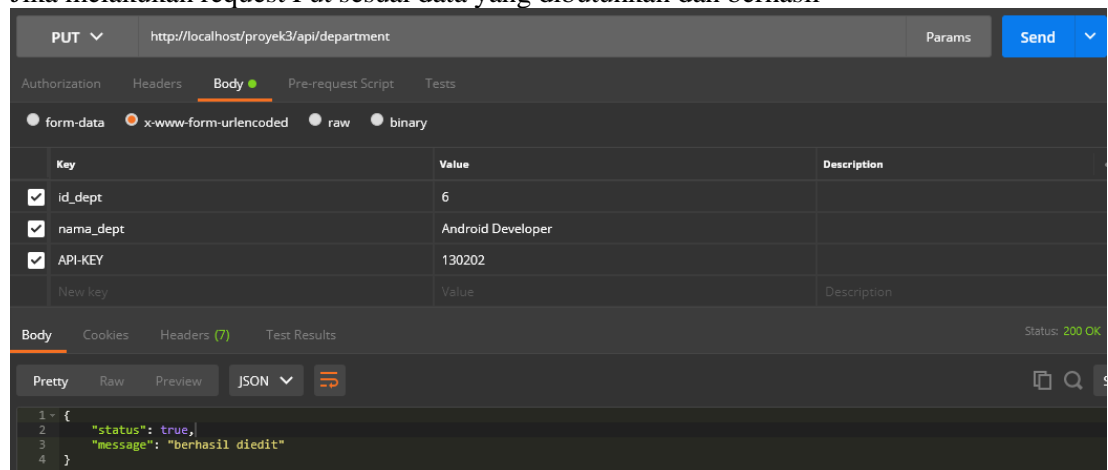


2. Jika melakukan post tidak memasukan data sesuai dengan yang dibutuhkan

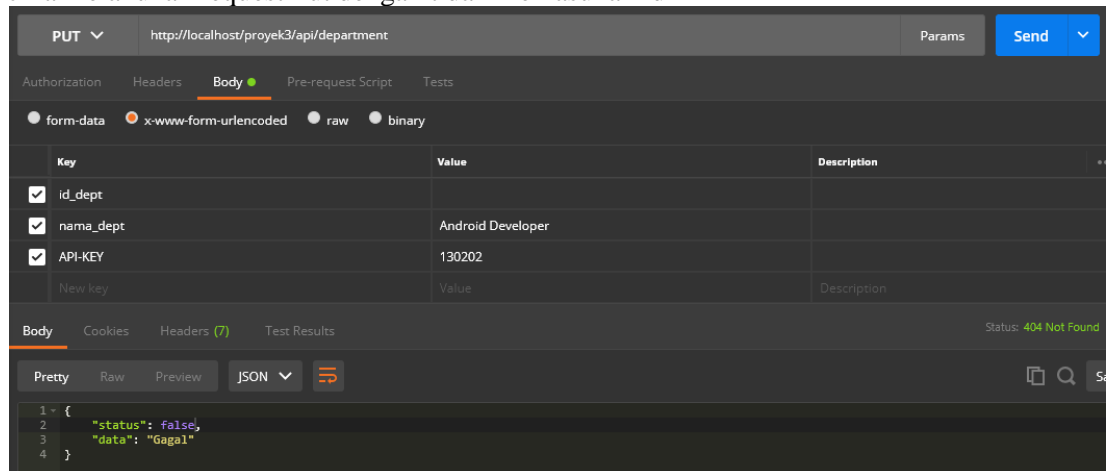


#### 6.2.8.4. Pengujian Fungsi Put

1. Jika melakukan request Put sesuai data yang dibutuhkan dan berhasil

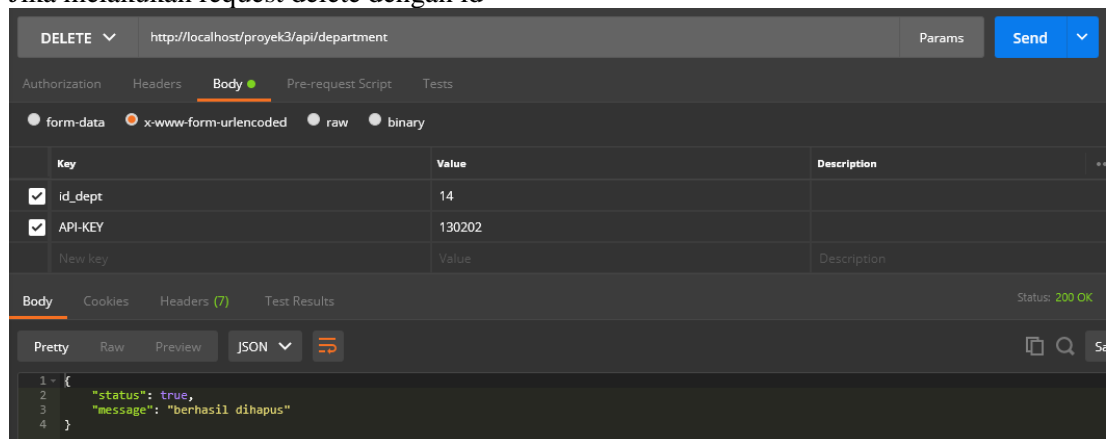


2. Jika melakukan request Put dengan tidak memasukan id

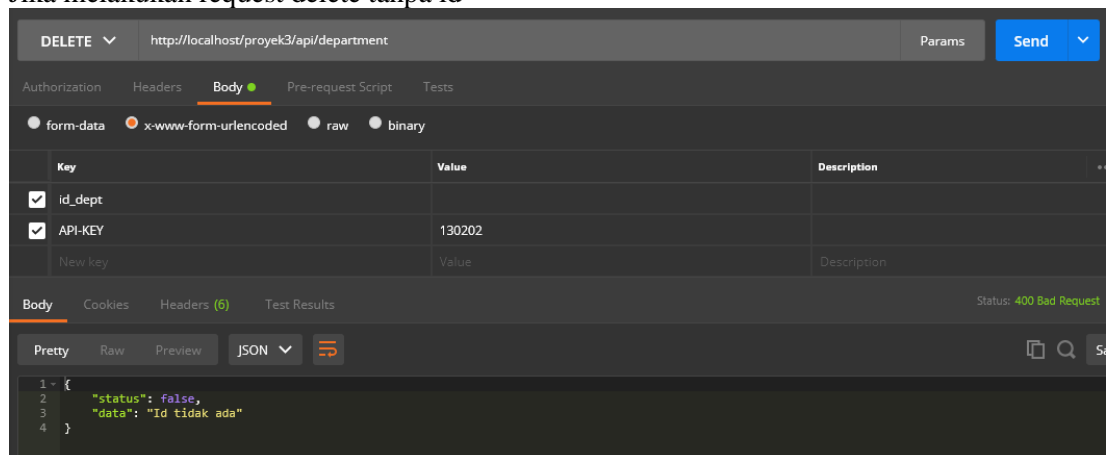


### 6.2.8.5. Pengujian Fungsi Delete

1. Jika melakukan request delete dengan id



2. Jika melakukan request delete tanpa id



## 7. Lampiran

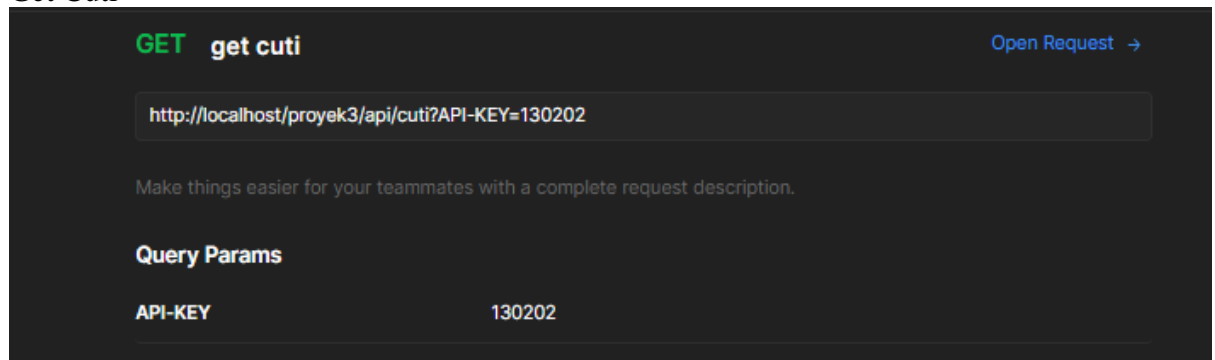
### 7.1. Pembagian Tugas

Pembagian tugas pada pembuatan system informasi penggajian karyawan dapat dilihat dibawah ini :

1. API dibangun menggunakan Bahasa pemrograman PHP dan framework codeigniter yang dibangun oleh Fahriza Rizky Amalia
2. Sistem informasi penggajian karyawan yang dibangun menggunakan Bahasa pemrograman PHP dan framework Codeigniter juga menggunakan library Guzzle dibangun oleh Salsabila Vebi Natasya

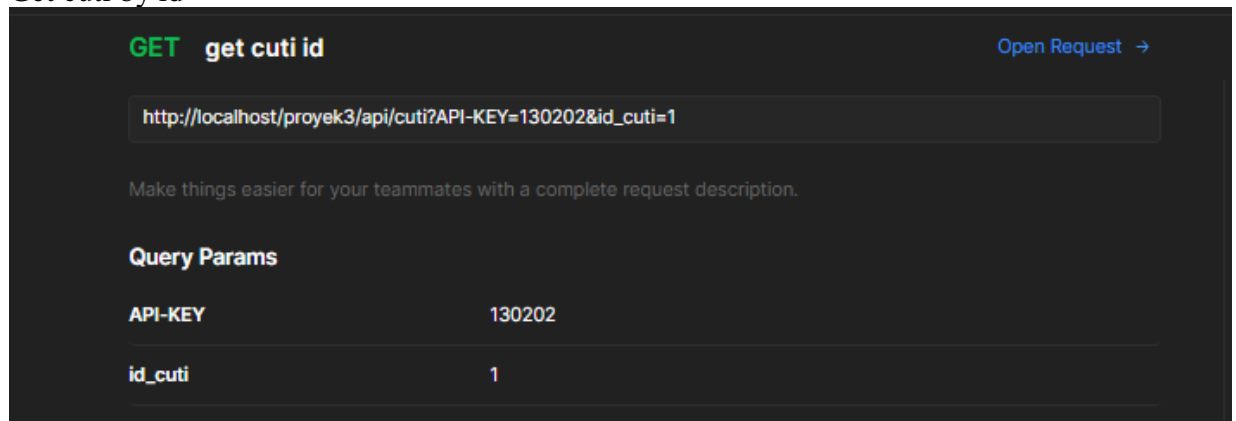
### 7.2. Dokumentasi Web Service

1. Table Cuti
  - a. Get Cuti



*Gambar 7.1. Get Cuti*

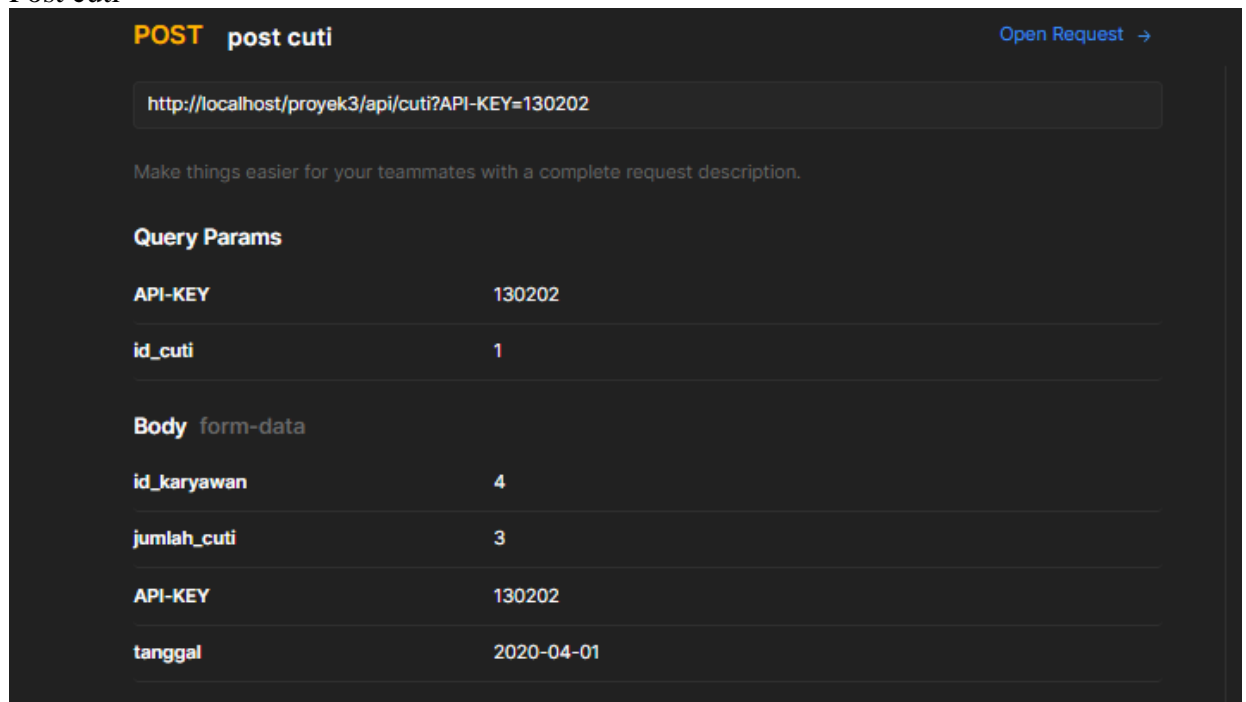
- b. Get cuti by id



*Gambar 7.2. Get cuti by id*



c. Post cuti



The screenshot displays a REST client interface for a POST request. The URL bar shows `http://localhost/proyek3/api/cuti?API-KEY=130202`. Below the URL, there is a section for 'Query Params' with two entries: `API-KEY` with value `130202` and `id_cuti` with value `1`. The 'Body' section is set to 'form-data' and contains three entries: `id_karyawan` with value `4`, `jumlah_cuti` with value `3`, and `tanggal` with value `2020-04-01`. An 'API-KEY' entry with value `130202` is also present in the body section. A link 'Open Request →' is visible in the top right corner.

Query Params	
API-KEY	130202
id_cuti	1

Body form-data	
id_karyawan	4
jumlah_cuti	3
API-KEY	130202
tanggal	2020-04-01

*Gambar 7.3. Post Cuti*

d. Put cuti

**PUT** put Open Request →

`http://localhost/proyek3/api/cuti`

Make things easier for your teammates with a complete request description.

**Query Params**

API-KEY	130202
id_cuti	1

**Body** urlencoded

id_cuti	1
id_karyawan	3
jumlah_cuti	4
tanggal	2020-01-01
API-KEY	130202

*Gambar 7.4. Put Cuti*

e. Delete cuti

**DEL** Delete Cuti Open Request →

`http://localhost/proyek3/api/cuti`

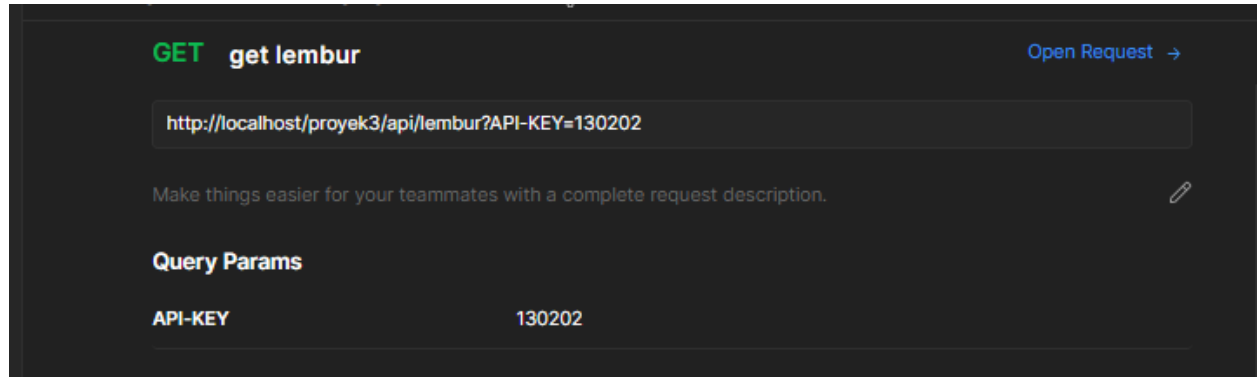
Make things easier for your teammates with a complete request description.

**Body** urlencoded

id_cuti	1
API-KEY	130202

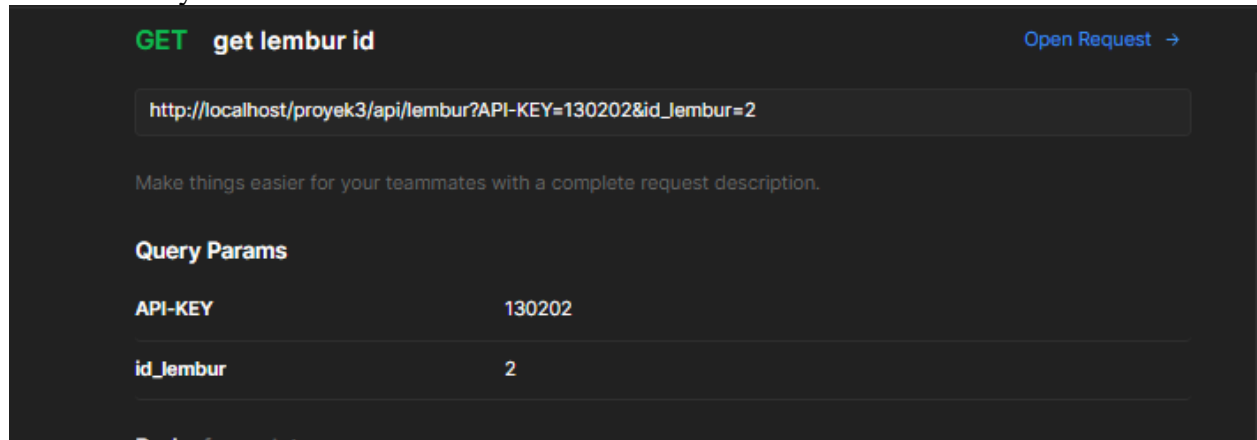
*Gambar 7.5. Delete Cuti*

2. Table Lembur
  - a. Get Lembur



*Gambar 7.6. Get lembur*

- b. Get lembur by id



*Gambar 7.7. Get Lembur By id*

- c. Post lembur

**POST** Post Lembur Open Request →

http://localhost/proyek3/api/lembur

Make things easier for your teammates with a complete request description.

**Body** urlencoded

id_karyawan	18
jumlah_jam_lembur	6
tanggal	2019-01-20
API-KEY	130202

Gambar 7.8. Post Lembur

d. Put lembur

**PUT** put Open Request →

http://localhost/proyek3/api/lembur

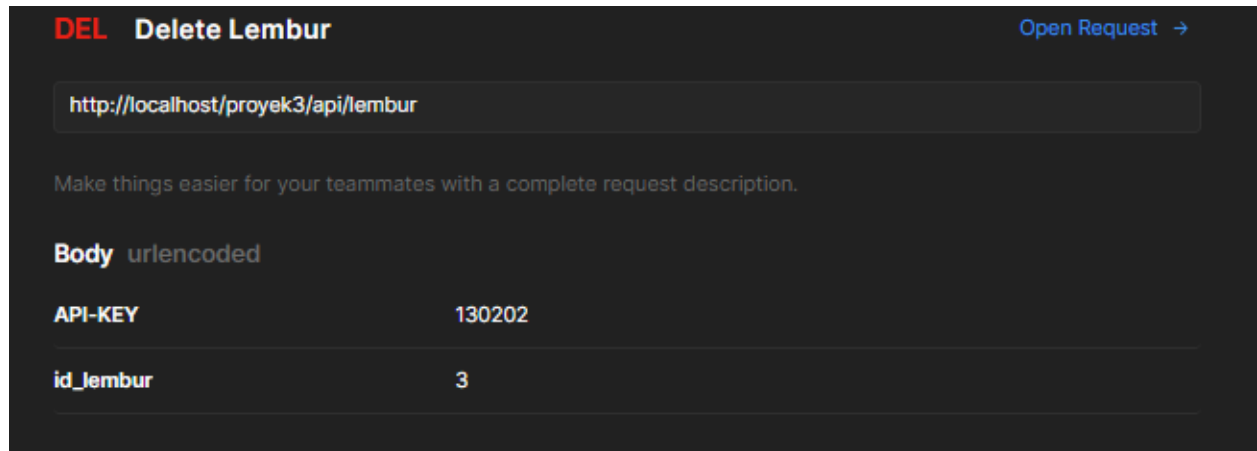
Make things easier for your teammates with a complete request description.

**Body** urlencoded

id_lembur	2
id_karyawan	18
jumlah_jam_lembur	4
tanggal	2019-10-01
API-KEY	130202

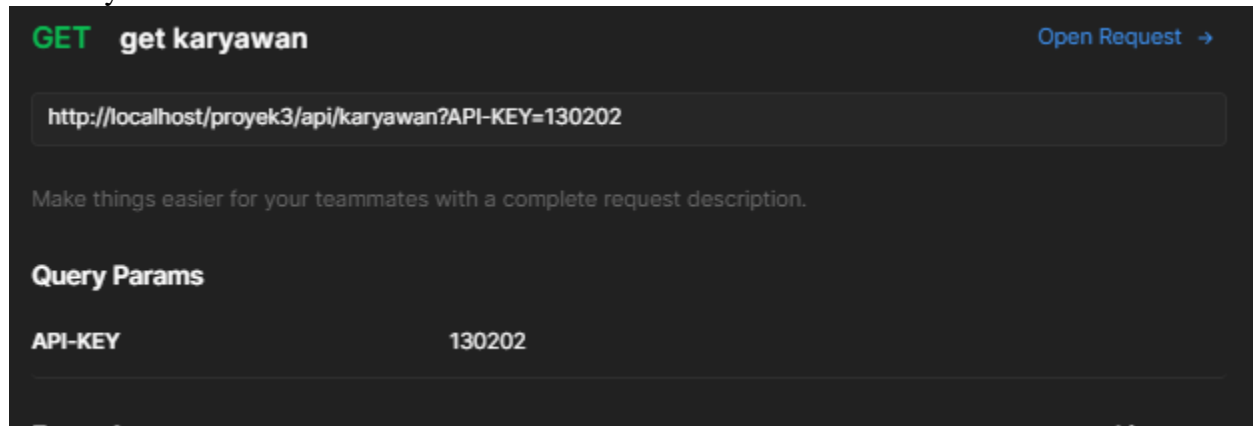
Gambar 7.9. Put Lembur

e. Delete lembur



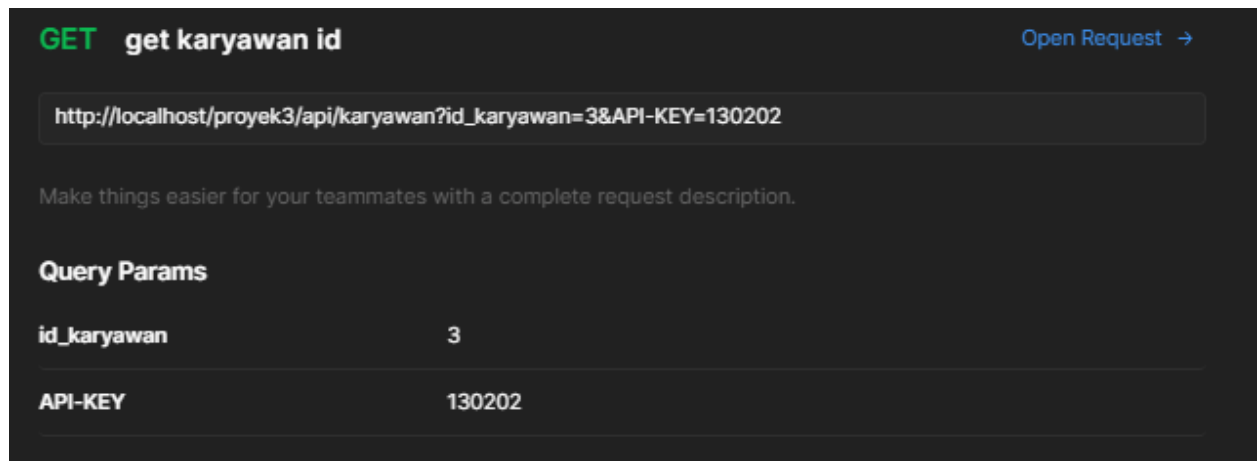
*Gambar 7.10. Delete Lembur*

3. Table karyawan
  - a. Get karyawan



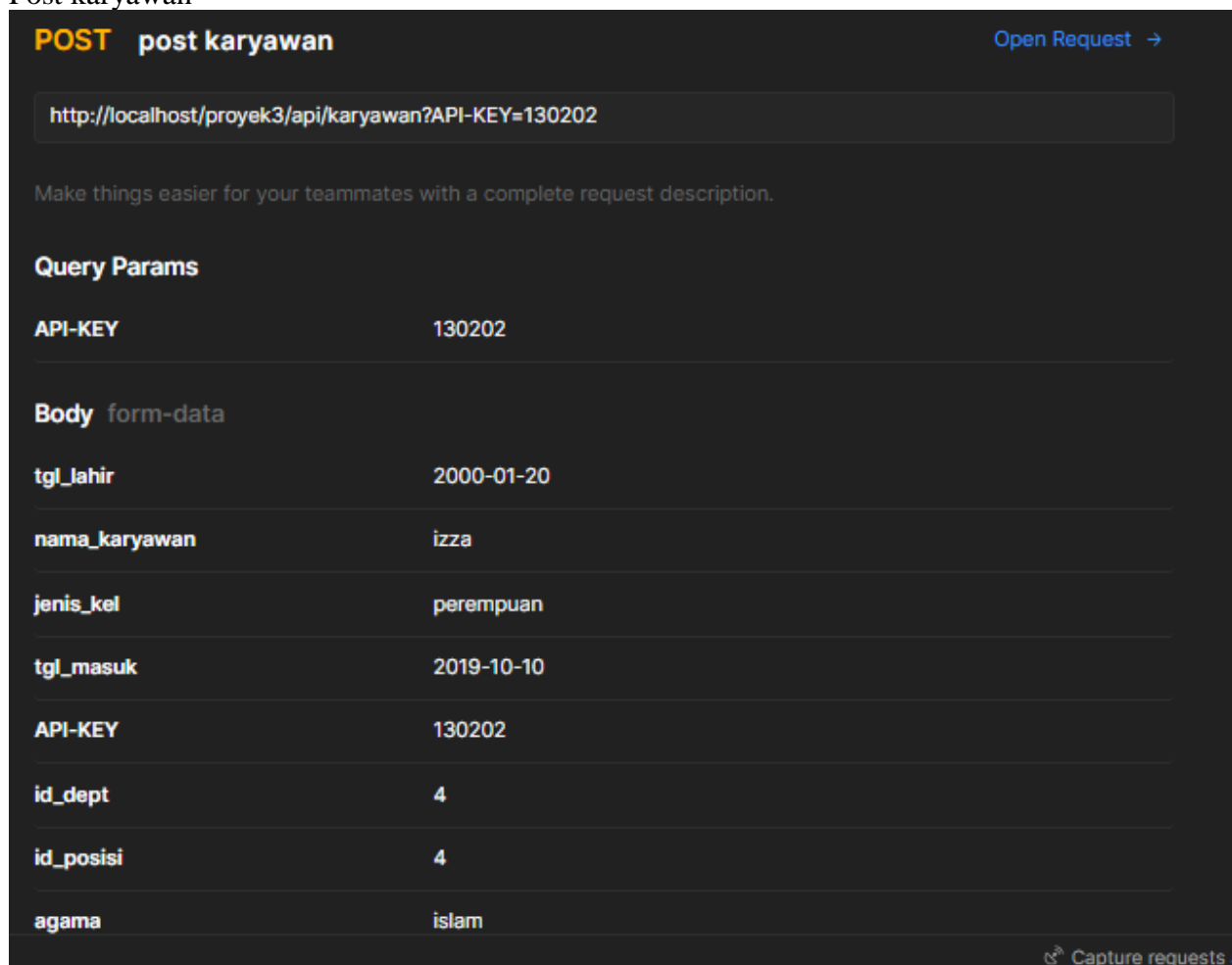
*Gambar 7.11. Get Karyawan*

- b. Get karyawan by id



Gambar 7.12. Get Karyawan By id

c. Post karyawan



Gambar 7.13. Post Karyawan

d. Put karyawan

**PUT** put karyawan Open Request →

http://localhost/proyek3/api/karyawan

Make things easier for your teammates with a complete request description.

**Query Params**

API-KEY	130202
---------	--------

**Body** urlencoded

id_karyawan	18
nama_karyawan	khansa
tgl_lahir	2001-01-02
jenis_kel	perempuan
tgl_masuk	2019-04-10
API-KEY	130202
id_dept	3
id_posisi	4

Capture requests and

Gambar 7.14. Put Karyawan

e. Delete karyawan

**DEL** Delete Karyawan Open Request →

http://localhost/proyek3/api/karyawan

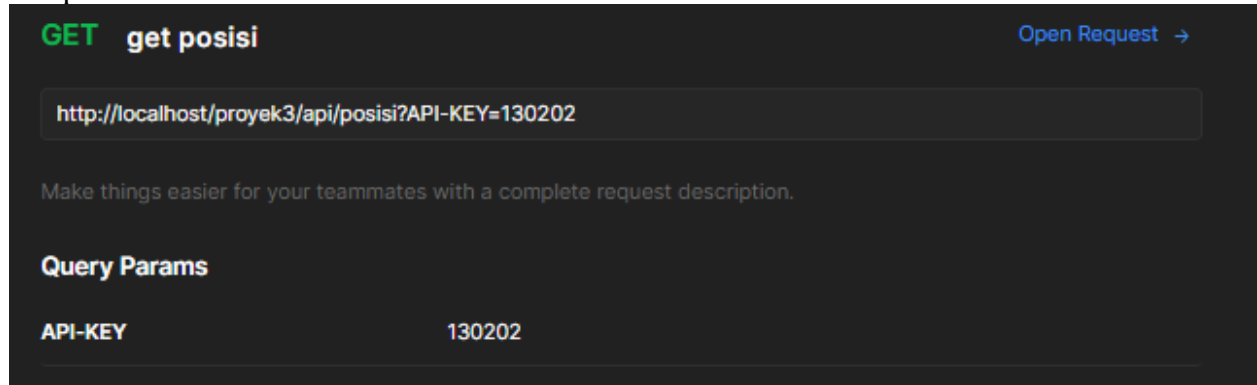
Make things easier for your teammates with a complete request description.

**Body** urlencoded

id_karyawan	25
API-KEY	130202

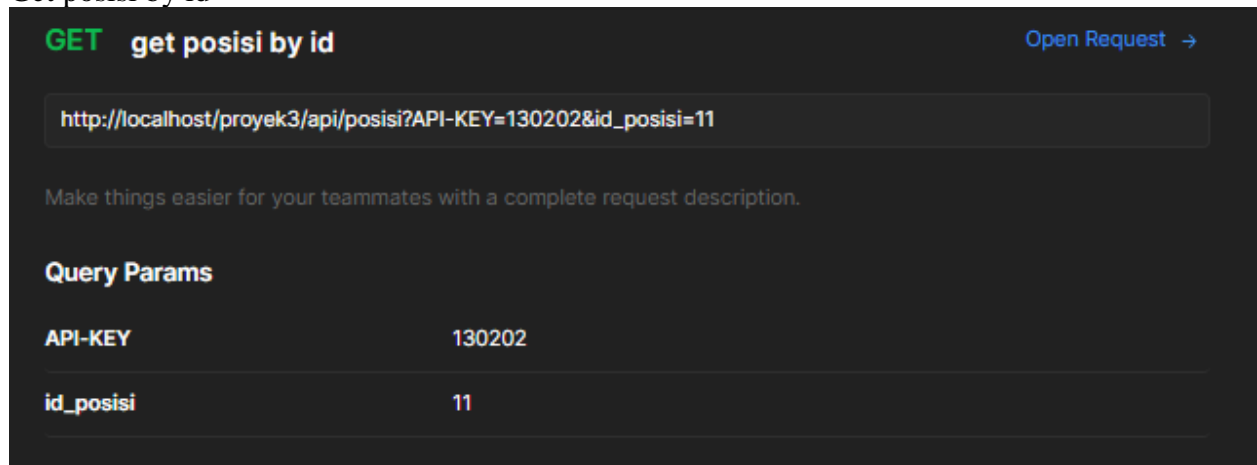
Gambar 7.15. Delete Karyawan

4. Table Posisi
  - a. Get posisi



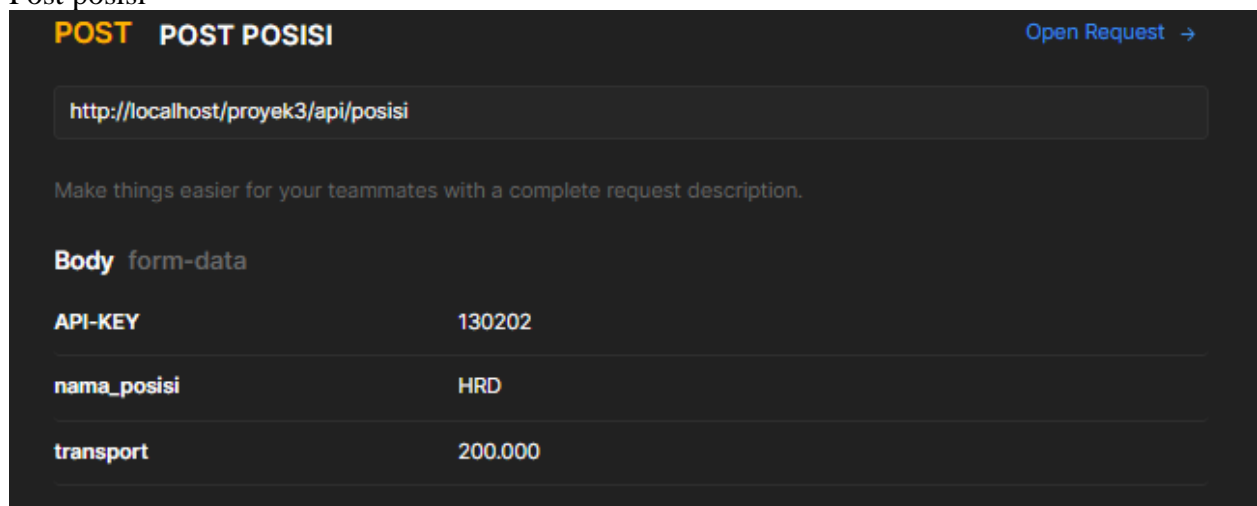
*Gambar 7.16. Get Posisi*

- b. Get posisi by id



*Gambar 7.17. Get Posisi By id*

- c. Post posisi



*Gambar 7.18. Post Posisi*



d. Put posisi

**PUT** put posisi [Open Request →](#)

`http://localhost/proyek3/api/posisi`

Make things easier for your teammates with a complete request description.

**Body** urlencoded

API-KEY	130202
nama_posisi	cleaning service
id_posisi	7
transport	Rp 400.000

*Gambar 7.19 Put Posisi*

e. Delete posisi

**DEL** posisi Delete [Open Request →](#)

`http://localhost/proyek3/api/posisi`

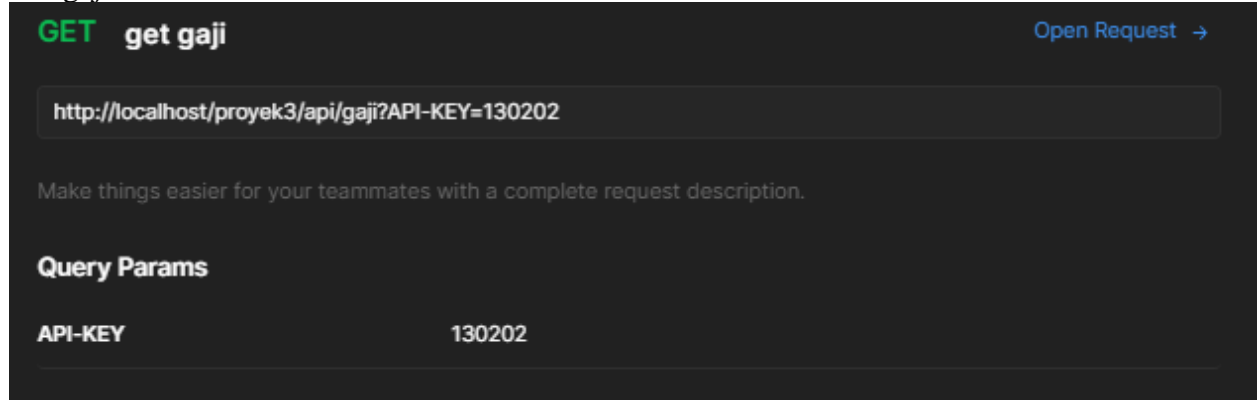
Make things easier for your teammates with a complete request description.

**Body** urlencoded

API-KEY	130202
id_posisi	12

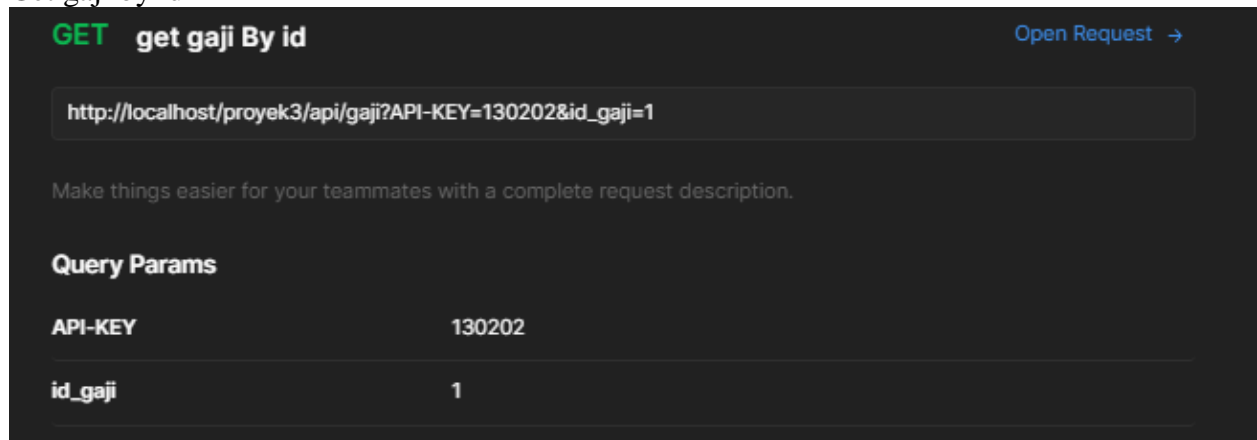
*Gambar 7.20. Delete Posisi*

5. Table Gaji
- a. Get gaji



*Gambar 7.21. Get Gaji*

- b. Get gaji by id



*Gambar 7.22. Get Gaji By id*

- c. Post gaji

**POST**
**Post Gaji**
[Open Request →](#)

<http://localhost/proyek3/api/gaji>

Make things easier for your teammates with a complete request description.

**Body** urlencoded

id_karyawan	18
GAPOK	Rp5.000.000
uang lembur	150.000
API-KEY	130202
tunjangan_keluarga	200.000
transport	100.000
potongan	0
total_gaji	Rp5.450.000
tgl_gaji	2022-01-10

Gambar 7.23. Post Gaji

d. Put gaji

**PUT** put Gaji Open Request →

http://localhost/proyek3/api/gaji

Make things easier for your teammates with a complete request description.

**Body** urlencoded

id_gaji	1
id_karyawan	3
GAPOK	Rp5.000.000
uang lembur	150.000
API-KEY	130202
tunjangan_keluarga	250.000
transport	100.000
potongan	0
total_gaji	Rp5.500.000
tgl_gaji	2022-02-09

Gambar 7.24. Put Gaji

e. Delete gaji

**DEL** Delete Gaji Open Request →

http://localhost/proyek3/api/gaji

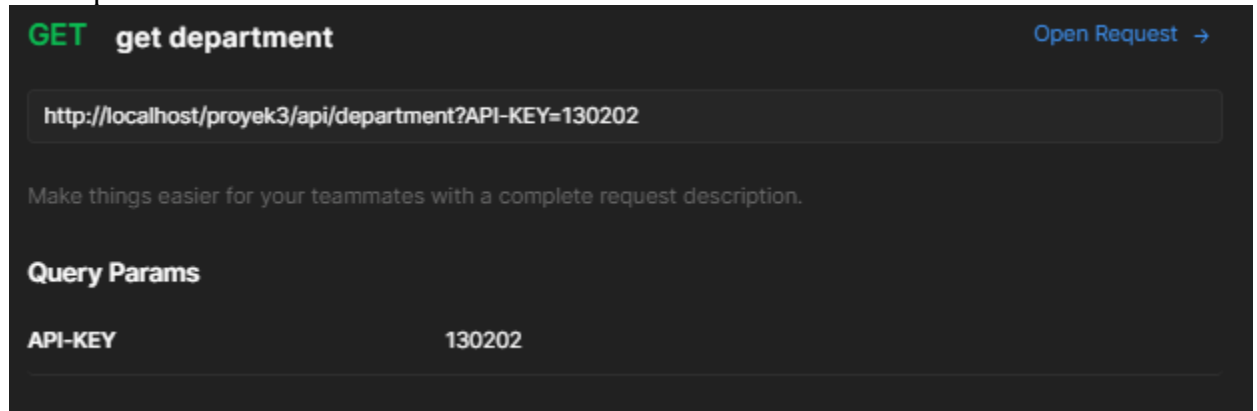
Make things easier for your teammates with a complete request description.

**Body** urlencoded

id_gaji	1
API-KEY	130202

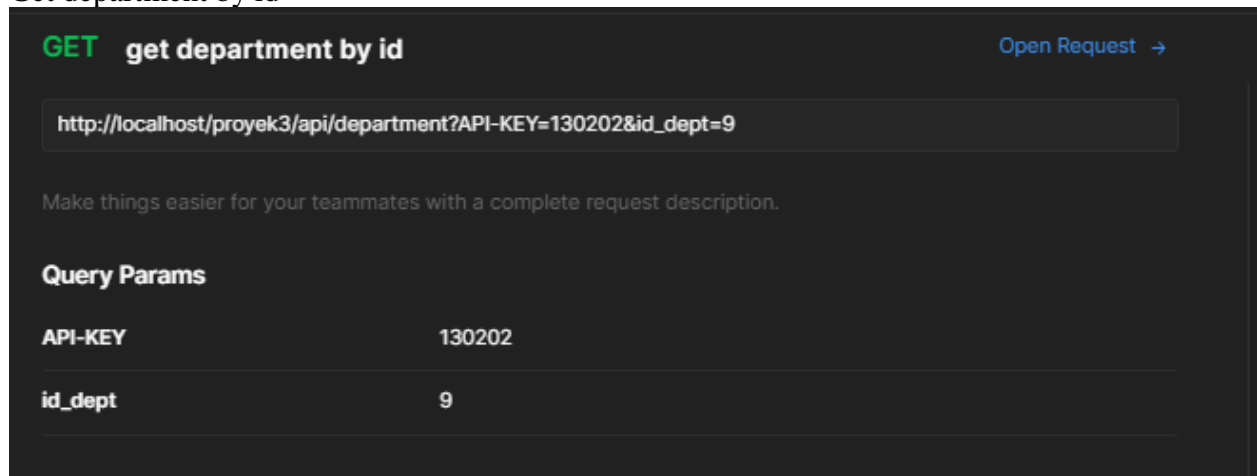
Gambar 7.25. Delete Gaji

6. Table Department
  - a. Get department



*Gambar 7.26. Get Department*

- b. Get department by id



*Gambar 7.27. Get Departmetn By id*

- c. Post department

**POST** post department Open Request →

`http://localhost/proyek3/api/department`

Make things easier for your teammates with a complete request description.

**Body** form-data

nama_dept	keuangan
API-KEY	130202

Gambar 7.28. Post Department

d. Put department

**PUT** put Department Open Request →

`http://localhost/proyek3/api/department`

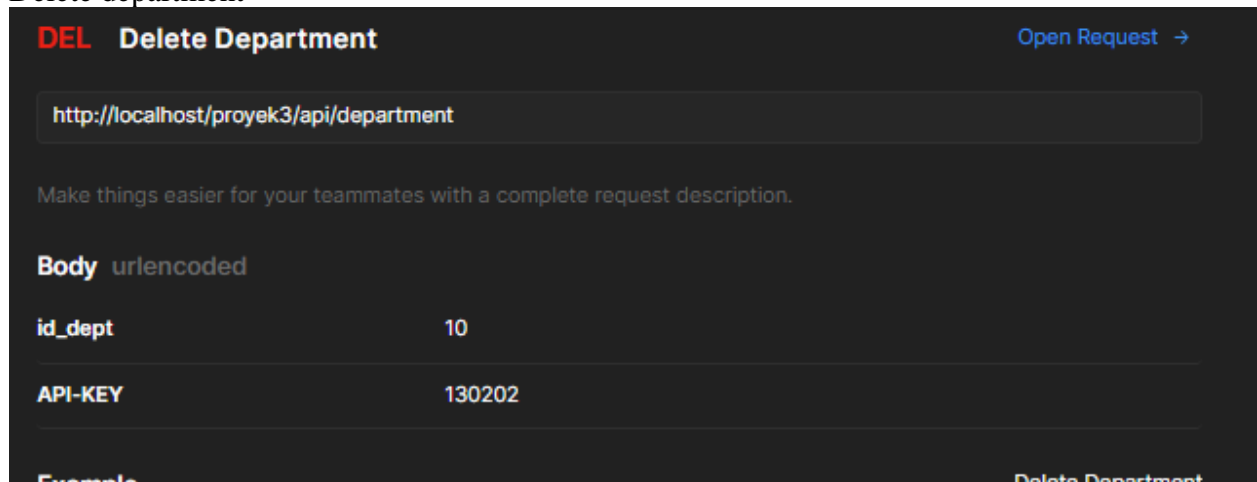
Make things easier for your teammates with a complete request description.

**Body** urlencoded

id_dept	9
nama_dept	Android Developer
API-KEY	130202

Gambar 7.29. Put Department

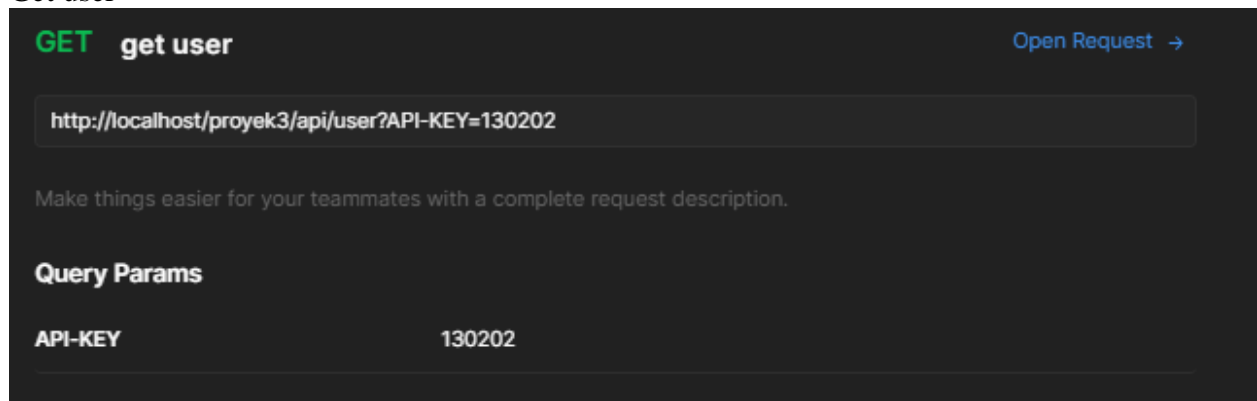
e. Delete department



Gambar 7.30. Delete Department

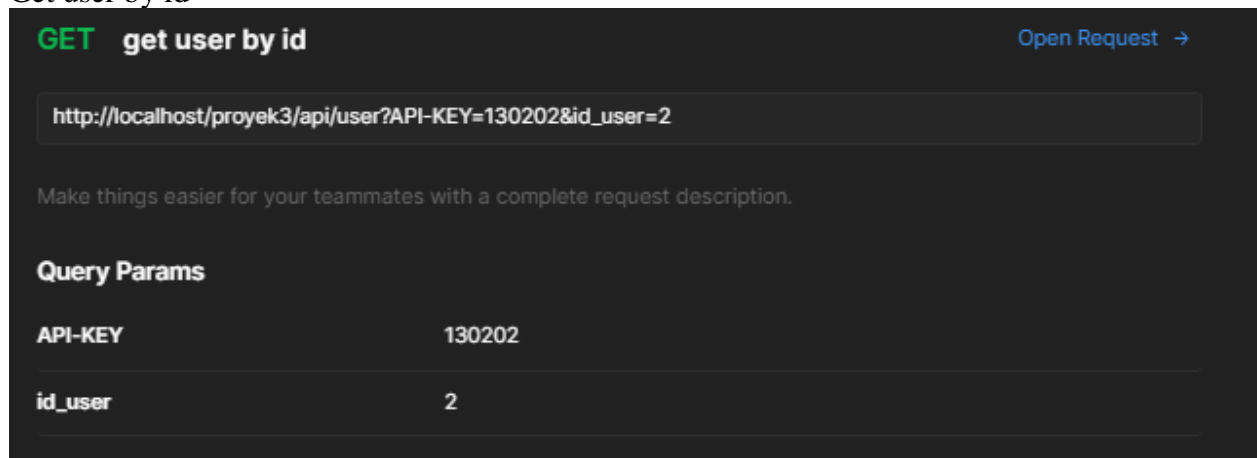
7. Table User

a. Get user



Gambar 7.36. Get User

b. Get user by id



Gambar 7.37. Get User By id

c. Post user

**POST** post user [Open Request →](#)

http://localhost/proyek3/api/user

Make things easier for your teammates with a complete request description.

**Body** form-data

username	izza
API-KEY	130202
password	izza
full_name	izza
email	izza@gmail.com
key	121244
level	HRD

*Gambar 7.38. Post User*

d. Put user



**PUT** put user Open Request →

`http://localhost/proyek3/api/user`

Make things easier for your teammates with a complete request description.

**Body** urlencoded

API-KEY	130202
id_user	6
username	admin3
password	admin3
full_name	izza
email	izza
key	123435
level	HRD

Gambar 7.39. Put User

e. Delete user

**DEL** Delete User Open Request →

`http://localhost/proyek3/api/user`

Make things easier for your teammates with a complete request description.

**Body** urlencoded

API-KEY	130202
id_user	6

Gambar 7.40. Delete User