

**BUKU TUTORIAL PENGGUNAAN DAN PENJELASAN APLIKASI
PENDATAAN GAJI DAN PEMBERIAN PINJAMAN DENGAN METODE
SIMPLE ADDITIVE WEIGHTING**

**“APLIKASI PEMBERITAHUAN PENGGAJIAN DAN SISTEM
PEMBERIAN REKOMENDASI PINJAMAN KREDIT(BANK BJB)
DENGAN METODE SIMPLE ADDITIVE WEIGHTING
(STUDI KASUS: RUTAN KLAS 1 BANDUNG)”**

Buku ini dibuat untuk memenuhi persyaratan kelulusan
matakuliah Program Internship I



**Dibuat Oleh,
1.16.4.012 Farhan Maulana**

**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA
POLITEKNIK POS INDONESIA
BANDUNG
2020**

Dengan menyebut nama Allah SWT Yang Maha Pengasih lagi Maha Penayang, penulis panjatkan puji serta syukur atas kehadiran Allah SWT. yang telah memberikan rahmat serta hidayahNya, sehingga penulis dapat menyelesaikan buku ini yang berjudul “Buku Tutorial Penggunaan Dan Penjelasan Aplikasi Pendataan Gaji Dan Pemberian Pinjaman Dengan Metode Simple Additive Weighting” dengan tepat waktu.

Dalam penulisan buku ini terdapat banyak kendala, terutama disebabkan karena keterbatasan ilmu pengetahuan, pengalaman serta kemampuan yang terbatas. Penulis mengharapkan adanya kritik dan saran dari semua pihak demi penyempurnaan pembuatan buku ini. Dalam buku ini tidak lepas dari bantuan, bimbingan dari semua pihak. Untuk itu penulis mengucapkan terima kasih kepada:

1. Allah SWT, karena dengan Rahmat dan Ridho-Nya kami dapat menyelesaikan program internship I ini.
2. Kedua orang tua dan keluarga kami yang telah mendukung dan memberi semangat kepada kami.
3. Bapak M. Yusril Helmi Setyawan, S.Kom., M.Kom selaku Ketua Program Studi DIV Teknik Informatika.
4. Ibu Nisa Hanum Harani, S.Kom, M.T. selaku Koordinator Internship I Tahun Akademik 2019/2020 sekaligus Pembimbing Internal dan Penguji Utama dalam penyusunan laporan Internship I ini;
5. Bapak Syaeful Bachri, A.md. selaku Pembimbing Eksternal dalam penyusunan laporan Internship I ini;

Harapan penulis semoga apa yang telah ditulis dapat dikembangkan lagi dan bermanfaat bagi pihak-pihak yang menggunakan.

Bandung, 20 Desember 2019

Penulis

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI	3
DAFTAR GAMBAR	4
DAFTAR TABEL.....	5
BAB I	
1.1 Pengantar Pemrograman Web	
1.1.1 Pengertian dan Penjelasan	
1.1.2 Pemrograman Web dan Jenisnya.....	
BAB II	
2.1 Pengenalan Metode Simple Additive Weighting	
2.1.1 Sistem Pendukung Keputusan	
2.1.1.1 Pengertian.....	
2.1.1.2 Fungsi	
2.1.1.3 Karakteristik	
2.1.2 Simple Additive Weighting	
2.1.2.1 Pengertian Simple Additive Weighting	
2.1.2.2 Rumus Perhitungan Metode Simple Additive Weighting	
2.1.2.3 Langkah Penyelesaian Metode Simple Additive Weighting	
BAB III.....	
3.1 Kebutuhan Pembuatan Program	
3.1.1 Database	
3.1.1.1 Pengertian.....	
3.1.1.2 Database Yang Digunakan	
3.1.1.3 Perancangan Database Program.....	
3.1.2 Framework.....	
3.1.2.1 Pengertian.....	

3.1.2.2 Daftar MVC Program	
3.1.3 Bahasa Pemrograman.....	
3.1.3.1 Pengertian.....	
3.1.3.2 PHP	
3.1.3.2 JavaScript	
BAB IV.....	
4.1 Penjelasan Program.....	
4.1.1 Penjelasan Setiap Halaman Pada Aplikasi	
4.1.1.1 Halaman Login	
4.1.1.2 Halaman Tabel Pegawai	
4.1.1.3 Halaman Tabel Potongan	
4.1.1.4 Halaman Tabel Pinjaman	
4.1.1.5 Halaman Detail Potongan Gaji Berdasarkan Bulan	
4.1.1.6 Halaman Detail Potongan Gaji Pegawai.....	
4.1.1.7 Halaman Pemberian Rekomendasi Pinjaman	
4.1.1.8 Halaman Pengambilan Data Pegawai (1).....	
4.1.1.9 Halaman Pengambilan Data Pegawai (2).....	
4.1.1.10 Halaman Pengambilan Data Potongan (1)	
4.1.1.11 Halaman Pengambilan Data Potongan (2)	
4.1.2 Penjelasan Fungsi Element HTML Disetiap Halaman	
4.1.2.1 Halaman Login	
4.1.2.2 Halaman Tabel Pegawai	
4.1.2.3 Halaman Tabel Potongan	
4.1.2.4 Halaman Tabel Pinjaman	
4.1.2.5 Halaman Detail Potongan Gaji Berdasarkan Bulan	
4.1.2.6 Halaman Detail Potongan Gaji Pegawai.....	
4.1.2.7 Halaman Pemberian Rekomendasi Pinjaman	
4.1.2.8 Halaman Pengambilan Data Pegawai (1).....	

4.1.2.9 Halaman Pengambilan Data Pegawai (2).....	
4.1.2.10 Halaman Pengambilan Data Potongan (1)	
4.1.2.11 Halaman Pengambilan Data Potongan (2)	

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

1.1 PENGANTAR PEMROGRAMAN WEB

1.1.1 Pengertian dan Penjelasan

Web Programming atau dalam bahasa Indonesia biasa disebut Pemrograman Web, Web programming ini terdiri dari 2 kata yang pertama yaitu yaitu web dan yang kedua adalah programming, programming ini bisa di artikan sebagai suatu proses pembuatan program atau aplikasi. Sedangkan untuk web dapat di artikan sebagai jaringan computer atau disebut juga dengan website yang didalamnya dapat berisi situs-situs jaringan internet yang menawarkan beberapa fitur dengan jenis yang beragam mulai dari text, grafik, suara, dan juga memelihara kode untuk membuat suatu pemrograman computer, kode ini dapat ditulis dalam berbagai macam bahasa pemrograman berbeda namun memiliki tujuan yang sama yaitu untuk membuat suatu program yang dapat melakukan perhitungan tertentu atau proses sesuai dengan keinginan programmer yang di akses melalui protocol HTTP.



Gambar 1.2 HTTP

HTTP Merupakan protocol lapisan aplikasi (application layer) yang didevelop atau biasa disebut dikembangkan untuk dapat membantu suatu proses transfer atau pengiriman dari satu computer ke computer lain. Protokol ini memiliki fungsi untuk melakukan transfer atau pengiriman berupa dokumen, file, gambar, dan video dari satu computer ke computer lain. Protokol HTTP ini menyediakan kumpulan-kumpulan perintah didalam suatu komunikasi antar jaringan.

Fungsi dari HTTP ini adalah mengatur format dan bagaimana suatu data di transmisikan atau biasa disebut dikirimkan dan diteruskan. HTTP juga memiliki fungsi untuk mengatur bagaimana web server dan web browser saling terkoneksi atau bisa disebut terhubung dan memproses berbagai perintah yang masuk. Fungsi lain dari HTTP ini bisa juga untuk mengamankan data dari perbuatan buruk berupa pencurian dan pembobolan. Hal ini bisa dilihat dengan hadirnya Hypertext transfer Protocol Secure atau biasa disebut HTTPS.



Gambar 1.3 *World Wide Web (WWW)*

WWW dapat diartikan sebagai beberapa halaman website yang dapat saling terhubung atau biasa disebut terkoneksi antara satu dengan lainnya (hyperlink) yang akan membentuk samudra informasi atau dengan kata lain berarti informasi yang sangat banyak. WWW berjalan dengan protokol HyperText Transfer Protokol (HTTP).

Halaman dari website sebenarnya merupakan file teks murni (plain text) yang memiliki isi sintaks-sintaks atau kode-kode HTML yang dapat dibuka kemudian dilihat dan terakhir diterjemahkan oleh Internet Browser. Sintaks atau kode HTML dapat memuat beberapa konten diantaranya text, audio, video, animasi, dan gambar. Sekarang internet lebih identik sebenarnya dengan web, karena dengan kepopulerannya, web bisa disebut sebagai standar interface atau tatap muka dengan pengakses pada layanan-layanan yang ada di internet, dari awalnya hanya sebagai penyedia informasi, ini digunakan juga untuk komunikasi dari email sampai dengan chatting, sampai dengan melakukan transaksi bisnis (commerce).

1.1.2 Pemrograman Web dan Jenisnya

Web adalah fasilitas dari Hypertext yang memiliki fungsi untuk menampilkan data berupa text, gambar, suara, animasi dan data multimedia dan jika ingin dapat menguasai web maka diperlukan mengenal beberapa bahasa pemrograman web yang populer dan dapat digunakan untuk membangun program berbasis web yang tergolong dalam script server side yaitu:

- PHP
- ASP.NET

- Ruby on Rails
- Pearl
- ASP classic
- Python
- JSP

Selain Script Server Side ada juga Script Client Side, maksudnya adalah sebagai interface pada pengguna aplikasi tersebut dalam menjalankan fungsi-fungsi yang terdapat dalam aplikasi, beberapa diantaranya yaitu:

- HTML
- XHTML
- CSS
- JavaScript
- XML

Web mengalami perkembangan yang sangat pesat mulai dari situs web E-commerce sampai dengan non profit situs. Dan dapat dikategorikan menjadi 2 yaitu web statis dan web dinamis.

1. Web Statis



Gambar 1.4 Web Statis

Website Statis atau dalam bahasa Inggris disebut Static Website adalah sebuah website yang memiliki isi atau biasa disebut konten bersifat statis atau juga dapat diartikan tidak berubah-ubah. Sekali dibuat dan sudah dihostingkan atau juga sudah online di Internet, sebenarnya website tersebut tidak dapat diubah kecuali apabila pengubahannya dilakukan secara manual dengan cara mengubah bahasa pemrograman website tersebut. Maka dari itu, interaksi yang terjadi pun sebenarnya cukup jarang, yang mengakibatkan dapat dikatakan seperti brosur online saja karena informasi yang diberikan juga terbatas.

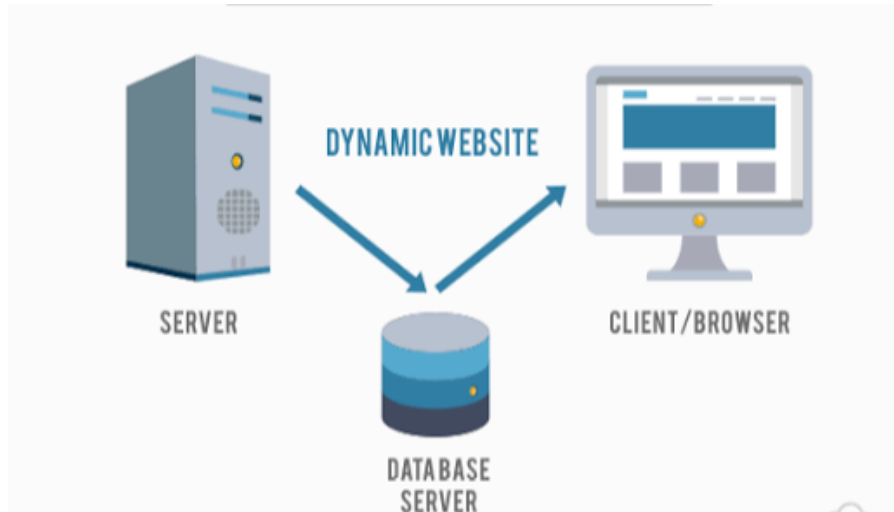
Ada beberapa keuntungan yang diberikan oleh website statis ini, diantaranya:

- Tidak terlalu membutuhkan keahlian pemrograman untuk membuat halaman yang statis
- Dapat dilihat secara langsung oleh web browser tanpa membutuhkan aplikasi server
- Lebih mudah untuk pengembang website mengembangkan websitenya karena hanya sebatas HTML

Selain keuntungan, web statis juga memiliki beberapa kerugian juga, diantaranya:

- Tampilannya kurang menarik
- Kontennya statis, tidak berubah-ubah
- Terbatas dalam interaksi dengan klien
- Tidak menggunakan database
- Tidak menggunakan pemrograman PHP di server

2. Web Dinamis



Gambar 1.5 Web Dinamis

Website Dinamis atau dalam bahasa Inggris disebut Dynamic Website bisa diartikan sebagai jenis halaman web yang tersusun atas konten beserta layout dengan banyak informasi didalamnya. Diberi nama website Dinamis sebenarnya karena konten yang terkandung dalam website tersebut dapat berubah-ubah sesuai dengan fungsi yang diatur. Dengan kata lain, ada program diluar website yang berjalan dan memiliki fungsi mengatur perubahan data yang ditampilkan tersebut didalam website. Halaman web yang dibuat atau dikembangkan menggunakan bahasa server seperti ColdFusion, PHP, ASP, Perl, ASP.NET, JSP, dan bahasa-bahasa yang lainnya. Untuk jenis website Dinamis ini bisa dibilang sangat cocok untuk website dengan tema atau dasar E-Commerce yang sangat membutuhkan pembaharuan terus menerus dari datanya. Selain itu untuk pemeliharaan website dinamis ini pun akan lebih mudah dibandingkan dengan Website Statis karena dapat menggunakan Content Management System atau biasa disebut CMS.

Ada beberapa keuntungan yang diberikan oleh website dinamis ini, diantaranya:

- Tampilannya menarik
- Konten dan layout dapat berubah-ubah
- Menggunakan dynamic html (DHTML)
- Menggunakan pemrograman server untuk mengatur perubahan data
- Dapat menggunakan CMS untuk mengubah konten website
- Isi dari halaman dan Layout dari halaman akan dibuat terpisah, sehingga memiliki keuntungan melakukan load halaman menjadi lebih cepat lebih cepat
- Dapat menggunakan database untuk menyimpan konten

Selain keuntungan, web dinamis juga memiliki beberapa kerugian juga, diantaranya:

- Aplikasi atau program akan lebih kompleks, hal ini membuat semakin sulit dalam melakukan pencarian dengan search engine

BAB II

2.1 PENGENALAN METODE SIMPLE ADDITIVE WEIGHTING

2.1.1 Sistem Pendukung Keputusan

2.1.1.1 Pengertian

Sistem Pendukung Keputusan atau biasa disebut dengan SPK adalah sebuah bagian dari aspek sistem informasi yang memiliki basis komputer termasuk juga system yang memiliki basis pengetahuan atau bisa juga tentang manajemen pengetahuan yang dipakai untuk mendukung keputusan yang diambil dalam suatu organisasi atau juga untuk individu. Selain itu bisa juga dikatakan

sebagai sistem komputer yang dapat melakukan pengolahan data menjadi informasi yang digunakan sebagai pengambilan keputusan dari masalah semi terstruktur yang spesifik.

Sistem Pendukung Keputusan atau biasa disebut dengan SPK ini dapat divisualisasikan sebagai sebuah sistem yang memiliki kemampuan mendukung sebuah analisis adhoc data, pemodelan dari sebuah keputusan, berorientasi pada keputusan yang akan diambil, orientasi dari perencanaan ke masa depan yang akan digunakan pada saat-saat yang bisa dibilang tidak biasa. Sistem Pendukung Keputusan atau biasa disebut SPK ini juga merupakan sebuah hasil gabungan dari beberapa sumber kecerdasan dari individu dengan kemampuan dari komponen untuk melakukan perbaikan kualitas dari sebuah keputusan dan menjadi sistem informasi yang memiliki basis komputer dalam hal manajemen pengambilan keputusan yang menangani beberapa masalah semi struktur.

Dari hasil penjelasan sebelumnya, diambil kesimpulan bahwa Sistem Pendukung Keputusan atau biasa disebut SPK ini sebenarnya bukan alat pengambilan keputusan, tapi SPK ini merupakan sebuah sistem yang dapat memberi bantuan dalam hal mengambil sebuah keputusan untuk melengkapi informasi dari data yang sebelumnya telah diolah dengan relevan dan juga dibutuhkan untuk membuat sebuah keputusan tentang suatu masalah dengan cara yang lebih efektif dan akurat. Sehingga sebenarnya sistem ini dibuat tidak dimaksudkan untuk menggantikan pengambil keputusan dalam proses pembuatan keputusan.

2.1.1.2 Fungsi

Secara global atau bisa disebut secara umum dapat dikatakan bahwa fungsi dari suatu Sistem Pendukung Keputusan atau lebih dikenal dengan SPK adalah untuk meningkatkan kemampuan dari seorang pengambil keputusan dengan memberikan beberapa alternatif keputusan yang bisa disebut lebih banyak atau lebih baik karena berdasar pada beberapa kriteria terkait, hal itu dapat membantu untuk merumuskan masalah dan keadaan yang akan dihadapi. Dengan demikian Sistem Pendukung Keputusan atau biasa disebut SPK akan dapat menghemat dalam hal waktu, tenaga kemudian biaya. Jadi dapat dikatakan Sistem Pendukung Keputusan mempunyai tujuan untuk lebih meningkatkan efektivitas (do the right things) dan juga tentunya efisiensi (do the things right) suatu kelompok atau individu dalam melakukan pengambilan keputusan. Walaupun demikian penekanan dari suatu Sistem Pendukung Keputusan atau biasa disebut SPK ini adalah pada peningkatan dalam hal efektivitas dari pengambilan suatu keputusan dari segi efisiensinya.

2.1.1.3 Karakteristik

Sistem Pendukung Keputusan atau biasa dikenal dengan SPK memiliki 6 karakteristik diantaranya adalah sebagai berikut :

- Mendukung proses dari pengambilan sebuah keputusan dengan berdasar pada manajemen berikut dengan persepsi.
- Adanya antar muka atau biasa disebut interface manusia atau mesin yang dimana manusia tersebut sebagai user tetap yang memegang kemudi proses dari pengambilan keputusan.
- Mendukung pengambilan dari sebuah keputusan untuk membahas perihal permasalahan yang terstruktur, semi terstruktur dan juga tidak terstruktur.

- Memiliki sebuah kapasitas dialog untuk memperoleh sebuah informasi yang sesuai dengan kebutuhan saat itu.
- Memiliki subsistem-subsistem yang saling terhubung sedemikian rupa yang akhirnya dapat berfungsi sebagai kesatuan sistem.
- Membutuhkan sebuah struktur data yang komprehensif dan dapat melayani suatu kebutuhan terhadap informasi dari seluruh tahapan manajemen.

2.1.2 Simple Additive Weighting

2.1.2.1 Pengertian Metode *Simple Additive Weighting* (SAW)

Metode SAW sebenarnya merupakan sebuah metode MADM yang bisa dibilang paling sederhana dan juga paling banyak digunakan karena cukup simpel. Metode ini juga metode yang paling mudah untuk diaplikasikan pada berbagai hal, karena mempunyai algoritma yang bisa dikatakan tidak terlalu rumit.

Metode SAW biasa dikenal juga dengan istilah metode dengan penjumlahan yang terbobot. Konsep dasar dari metode SAW ini sebenarnya mencari penjumlahan yang terbobot dari nilai kinerja pada masing-masing alternatif di semua atribut. Metode ini juga harus ada proses dari normalisasi terlebih dahulu pada matriks keputusan (X) ke suatu skala yang bertujuan untuk pembandingan dengan semua nilai alternatif yang ada.

2.1.2.2 Rumus Perhitungan Metode *Simple Additive Weighting* (SAW)

Formula dibawah ini adalah formula yang digunakan untuk melakukan normalisasi dari perhitungan metode SAW adalah sebagai berikut:

$$R_{ij} \begin{cases} \frac{X_{ij}}{\max X_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min X_{ij}}{X_{ij}} & \text{jika } j \text{ adalah atribut pengeluaran (cost)} \end{cases}$$

Keterangan:

R_{ij} = Nilai kinerja yang sudah dinormalisasi dari alternatif A_i pada atribut

C_j : $i=1,2,\dots,m$ dan $j = 1,2, \dots, n$

$\max X_{ij}$ = Nilai terbesar dari setiap kriteria i

$\min X_{ij}$ = Nilai terkecil dari setiap kriteria i

X_{ij} = Nilai atribut dari setiap kriteria yang ada

Benefit = Jika nilai terbesar adalah terbaik

Cost = Jika nilai terkecil adalah terbaik

Nilai untuk setiap alternatif akan diberikan rumus seperti dibawah ini:

$$V_i = \sum_{j=1}^n W_j r_{ij}$$

Keterangan:

V_i = Ranking untuk setiap alternatif

W_j = Nilai bobot ranking (dari setiap alternatif)

r_{ij} = Nilai rating kinerja ternormalisasi

Nilai dengan perhitungan yang lebih besar disbanding yang lain akan mengindikasikan alternatif tersebut lebih terpilih yaitu alternative.

2.1.2.3 Langkah Penyelesaian Metode *Simple Additive Weighting* (SAW)

- 1) Menentukan beberapa kriteria terpilih sebagai acuan untuk mengambil keputusan, yaitu C_i .

- 2) Menentukan rating kecocokan dari setiap alternatif pada setiap atributnya.
- 3) Menyiapkan matriks keputusan yang didasari oleh kriteria (C_i), kemudian dilakukan perhitungan normalisasi matriks tersebut dari persamaan yang sebelumnya telah disesuaikan dengan atribut (atribut benefit ataupun atribut cost) maka akan didapat matriks ternormalisasi.
- 4) Hasil akhir dari semua perhitungan diperoleh melalui proses perangkingan dengan cara menjumlahkan dari hasil perkalian matriks ternormalisasi R dengan hasil vektor bobot yang akan memperoleh nilai terbesar kemudian dinyatakan sebagai alternative terbaik untuk dipilih sebagai solusi.

BAB III

3.1 KEBUTUHAN PEMBUATAN PROGRAM

3.1.1 DATABASE



Gambar 3.1 Database

3.1.1.1 Pengertian

Database memiliki arti sebagai kumpulan dari beberapa data yang terhubung antara satu dan lainnya, tersimpan didalam *hardware* atau biasa dikenal dengan perangkat lunak dan juga bisa digunakan didalam *software* atau biasa disebut perangkat lunak untuk memanipulasinya. Sistem manajemen basis data atau biasa disebut dengan DBMS ini bisa diartikan sebagai kumpulan data yang saling berhubungan dan kumpulan dari beberapa program untuk mengakses data tersebut.

Komponen utama dari sebuah sistem database ini terdiri dari beberapa bagian yang akan disebutkan seperti berikut:

1. *Data* : diutamakan data yang bersifat *integrity* atau biasa disebut kesatuan dan *share* atau biasa juga disebut pemakaian bersama.
2. *Hardware* : semua yang menyangkut dalam hal media penyimpanan baik berupa penyimpanan eksternal, kemudian piranti input dan juga output.
3. *Software* : berupa *database* manajemen system seperti SQL, yang merupakan penghubung antara alumni dengan data yang tersimpan didalam media penyimpanan secara fisik.
4. *Database* : merupakan kumpulan dari beberapa file yang memiliki hubungan antara satu file dengan file lainnya. *Database* secara fisik akan disimpan didalam media penyimpanan seperti system computer.
5. *Alumni*
 - *Programmer Aplikasi* : program aplikasi yang menyediakan keperluan fungsional untuk pemakai akhir yang harus diimplementasikan, ini adalah tanggung jawab dari *aplikasi*

programmer yang bekerja dari suatu spesifikasi yang dihasilkan dan analisis system. Setiap program berisi statemen yang diperlukan DBMS untuk melakukan operasi terhadap *database* yang terdiri dari penyisipan, penghapusan dan sebagainya.

- *End Alumni* : merupakan *client* terhadap *database* dimana *end alumni* juga dapat diklasifikasikan menurut cara mereka menggunakan system.

Database Administrator : memiliki tanggung jawab dalam hal merealisasikan fisik system *database* yang juga termasuk dalam rancangan *database* keamanan dan pengendalian kesatuan, implementasi, fisik, pemeliharaan dari sebuah system yang operasional dan keyakinan keputusan penampilan untuk aplikasi pemakai.

3.1.1.2 Database Yang Digunakan



Gambar 3.2 MySQL Logo

MySQL adalah Database Management System yang memiliki sifat open source atau berarti bisa diakses dan dikembangkan oleh

siapapun dan memiliki dua buah bentuk lisensi, yaitu Free Software atau dalam bahasa Indonesia berarti perangkat lunak bebas lalu kemudian Shareware atau dalam bahasa Indonesia berarti perangkat lunak berpemilik yang penggunaannya terbatas. Jadi MySQL ini sebenarnya database server yang dapat dikatakan gratis dengan diperlengkap oleh lisensi GNU General Public License atau disingkat GPL, sehingga dapat dipakai untuk sebuah keperluan baik pribadi maupun komersil tanpa adanya bayaran lisensi yang ada.

Seperti yang sudah dikatakan sebelumnya, MySQL ini masuk ke dalam jenis Relational Database Management System atau kepanjangan dari RDBMS . oleh karena itu istilah seperti tabel, kolom, baris juga dipakai didalam MySQL.

SQL ini juga merupakan sebuah bahasa yang digunakan di dalam mengambil suatu data pada database yang terstruktur. Jadi MySQL dapat disebut DBMS yang menggunakan bahasa SQL dalam pengaliksiannya sebagai bahasa yang menghubungkan antara perangkat lunak aplikasi dengan database server.

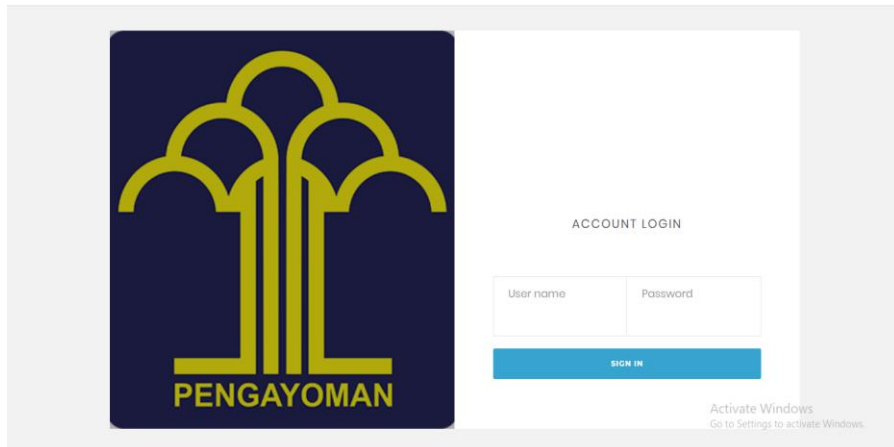
3.1.1.3 Perancangan Database Aplikasi

BAB IV

4.1 PENJELASAN TAMPILAN PROGRAM

4.1.1 Penjelasan Setiap Halaman Pada Aplikasi

4.1.1.1 Halaman *Login*



Gambar 1. Halaman *Login*

Halaman *login* atau pada gambar 1 ini adalah awal untuk menjalankan semua fungsi yang dimana pegawai keuangan di Rumah Tahanan Negara Klas 1 Bandung harus memasukkan *username* dan *password* terlebih dahulu.

4.1.1.2 Halaman Tabel Pegawai

Daftar Tabel / Tabel Pegawai

Data Pegawai (Gaji Pokok)

PDF Excel CSV Copy Print

Search:

No.	NIP	Nama Pegawai	Golongan	Rekening	Nama Rekening	Nama Bank	Email	Gaji Bersi
1	197501051998031001	HERI KUSRITA, A.Md.IP.SHM	IV/a	0089596254100	HERI KUSRITA	BPD JABAR BANTEN	farhanmaulana88@gmail.com	Rp5.333.3
2	196701031996031002	Drs. DENI DASMANA, MAP	IV/a	0080325622101	DENI DASMANA	BPD JABAR BANTEN	deni_dasmana@yahoo.com	Rp4.854.9
3	196710291993031001	Drs. JENAL ASIKIN, MAP	IV/a	0001692607100	JENAL ASIKIN	BPD JABAR BANTEN	jenal_asikin86@yahoo.com	Rp5.225.5
4	197007201992012001	DEDE WENDARI,	IV/a	0080254776100	DEDE WENDARI	BPD JABAR	dedewendari1970@gmail.com	Rp5.111.3

Gambar 2. Halaman Tabel Pegawai

Halaman tabel pegawai pada gambar 2 ini adalah halaman yang dituju setelah pegawai keuangan di Rumah Tahanan Negara Klas 1 Bandung selesai dan berhasil melakukan login. Halaman ini berisi beberapa data pilihan dari seluruh pegawai di Rumah Tahanan Negara Klas 1 Bandung yang memang dibutuhkan untuk pendataan didalam aplikasi.

4.1.1.3 Halaman Tabel Potongan

No.	Nama Pegawai	Bulan Tahun	Arisan Pengayoman	IDW	Koperasi	Tabungan DW	Olahraga	BRI	BJB	SKP	Lain-lain	Arit DW
1	HERI KUSRITA, A.Md.IP.SH.MH	Aug-19	120.000	25.000	200.000	0	1.000	0	0	1.500	0	105,-
2	Drs. DENI DASMANA, MAP	Aug-19	0	10.000	2.142.000	0	1.000	0	0	1.500	0	105,-
3	Drs. JENAL ASKIN, MAP	Aug-19	0	5.000	200.000	0	1.000	0	0	1.500	0	105,-
4	DEDE	Aug-19	0	5.000	700.000	0	1.000	0	0	1.500	4.015.573	105,-

Gambar 3. Halaman Tabel Potongan

Halaman tabel potongan pada gambar 3 ini adalah berisi seluruh data penggajian dari pegawai beserta potongan-potongannya yang menjadi alasan pegawai tersebut tidak menerima seluruh gaji kotornya. Seluruh data potongan yang sudah dimasukan kedalam program, maka akan ditampilkan disini dan berpatok pada bulan data tersebut dimasukan kedalam aplikasi.

4.1.1.4 Halaman Tabel Pinjaman

Daftar Tabel / Tabel Pinjaman

Data Pinjaman Pegawai (Bank BJB)

PDF Excel CSV Copy Print

Search:

No.	NIP	Nama Pegawai	Gaji Kotor	Bulan Peminjaman	Platfond	Lama Bulan	Angsuran Bulanan	Total	Status
1	197501051998031001	HERI KUSRIITA, A.Md.IP.SH.MH	Rp5.333.300	Aug-19	Rp10.000.000	12	888488	Rp10.661.856	BELUM LUNAS

Showing 1 to 1 of 1 entries

Previous 1 Next

Copyright © Your Website 2019

Activate Windows
Go to Settings to activate Windows.

Gambar 4. Halaman Tabel Pinjaman

Halaman tabel pinjaman pada gambar 4 ini berisi pendataan pinjaman dari pegawai yang telah melalui proses dari aplikasi. Data pinjaman pegawai Rumah Tahanan Klas 1 Bandung pada tabel pinjaman ini setiap bulannya akan selalu berkurang setiap ditambahkannya data potongan baru karena salah satu potongan pada gambar 4 sebelumnya adalah pinjaman ini, setelah potongan tersebut lunas maka data potongan tidak akan hilang namun akan berganti status menjadi LUNAS yang sebelumnya adalah BELUM LUNAS.

4.4.2 Halaman Detail Potongan Gaji Berdasarkan Bulan

PENDATAAN Gaji POKOK

Admin

Dashboard

INTERFACE

Daftar Tabel

Import Data

Rekap Total Potongan Gaji Bulan Sep-19

Total Potongan Rp.405,787,148

Total Gaji Diterima Rp.225,575,752

Data Potongan Gaji

PDF Excel CSV Copy Print

Search:

Select	No.	Nama Pegawai	Bulan Tahun	Arisan Pengayoman	IDW	Koperasi	Tabungan DW	Olahraga	BRI	BIB	SKP
<input type="checkbox"/>	1	HERI KUSRITA, A.Md.IP.SH.MH	Sep-19	120.000	25.000	200.000	0	1.000	0	4.707.347	1.500
<input type="checkbox"/>	2	Drs. DENI DASMANA, MAP	Sep-19	0	10.000	2.123.000	0	1.000	0	0	1.500
<input type="checkbox"/>	3	Drs. JENAL ASKIN, MAP	Sep-19	0	5.000	200.000	0	1.000	0	0	1.500
<input type="checkbox"/>	4	DEDE WENDARI, MAP	Sep-19	0	5.000	700.000	0	1.000	0	0	1.500

Gambar 5. Halaman Detail Potongan Gaji Berdasarkan Bulan

Halaman tabel detail potongan gaji berdasarkan bulan pada gambar 5 hampir sama seperti pada gambar 3 yaitu berisi data potongan dari gaji pegawai Rumah Tahanan Negara Klas 1 Bandung tapi perbedaannya adalah halaman ini menampilkan seluruh data potongan yang lebih spesifik pada salah satu bulan. Halaman pada gambar 5 ini juga adalah salah satu jalur akses untuk pegawai keuangan Rumah Tahanan Negara Klas 1 Bandung melakukan pengiriman data potongan gaji seluruh pegawai melalui *email* masing-masing pegawai.

4.4.3 Halaman Detail Potongan Gaji Pegawai

PENDATAAN GAJI POKOK

Admin

Detail Potongan Gaji Pegawai
 Nama : DADANG GUNAWAN
 NIP : 196209221998101001

Data Potongan Gaji

PDF Excel CSV Copy Print

Search:

Select All	No.	Bulan Tahun	Arisan Pengayoman	IDW	Koperasi	Tabungan DW	Olahraga	BRI	BIB	SKP	Lain-lain	Arisan DW	Aris 100
<input type="checkbox"/>	1	Aug-19	0	5.000	1.563.000	0	1.000	0	1.912.502	1.500	0	105.000	0

Kirim

Copyright © Your Website 2019

Activate Windows
Go to Settings to activate Windows.

Gambar 6. Halaman Detail Potongan Gaji Pegawai

Halaman tabel detail potongan gaji pegawai pada gambar 6 hampir sama seperti pada gambar 3 yaitu berisi data potongan dari gaji pegawai Rumah Tahanan Negara Klas 1 Bandung tapi perbedaannya adalah halaman ini menampilkan data potongan yang lebih spesifik pada salah satu pegawai Rumah Tahanan Negara Klas 1 Bandung sesuai dengan data pegawai Rumah Tahanan Negara Klas 1 Bandung yang dipilih. Halaman pada gambar 6 ini juga adalah jalur akses untuk pegawai keuangan Rumah Tahanan Negara Klas 1 Bandung melakukan pendataan pinjaman pada aplikasi kepada seluruh pegawai yang akan melakukan pinjaman dengan berpatok pada bulan pegawai tersebut melakukan pinjaman.

4.4.4 Halaman Pemberian Rekomendasi Pinjaman

Daftar Tabel / Tabel Pegawai

Data Pegawai (Gaji Pokok)

Gaji : Rp. 4.218.300

PDF Excel CSV Copy Print

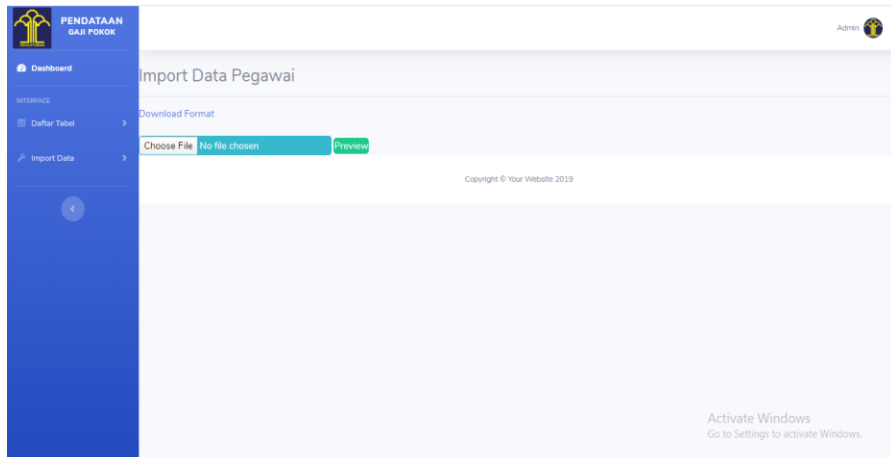
Search:

No.	Plafond	Lama Pinjaman (Bulan)	Angsuran Tiap Bulan	Value Rekomendasi	
1	Rp. 10.000.000	12	Rp. 888.488	0.28701600922016	Pinjam
2	Rp. 10.000.000	24	Rp. 470.735	0.21099132208143	Pinjam
3	Rp. 10.000.000	36	Rp. 332.143	0.19893493665881	Pinjam
4	Rp. 10.000.000	48	Rp. 263.338	0.20115053657277	Pinjam
5	Rp. 10.000.000	60	Rp. 222.444	0.20790760820701	Pinjam
6	Rp. 10.000.000	72	Rp. 195.502	0.21611161693146	Pinjam

Gambar 7. Halaman Pemberian Rekomendasi Pinjaman

Halaman pemberian rekomendasi pinjaman pada gambar 7 ini adalah tempat penyimpanan metode *Simple Additive Weighting* (SAW) yang akan memberikan pilihan terbaik sesuai dengan beberapa kriteria yang telah disetujui dalam melakukan pinjaman. Karena rekomendasi bersifat opsional maka pilihan yang paling direkomendasikan oleh aplikasi tidak harus selalu dipilih oleh pegawai Rumah Tahanan Klas 1 Bandung. Metode SAW disini adalah sebagai pembantu dalam mengambil keputusan pinjaman terbaik sesuai dengan kriteria dan gaji yang diterimanya.

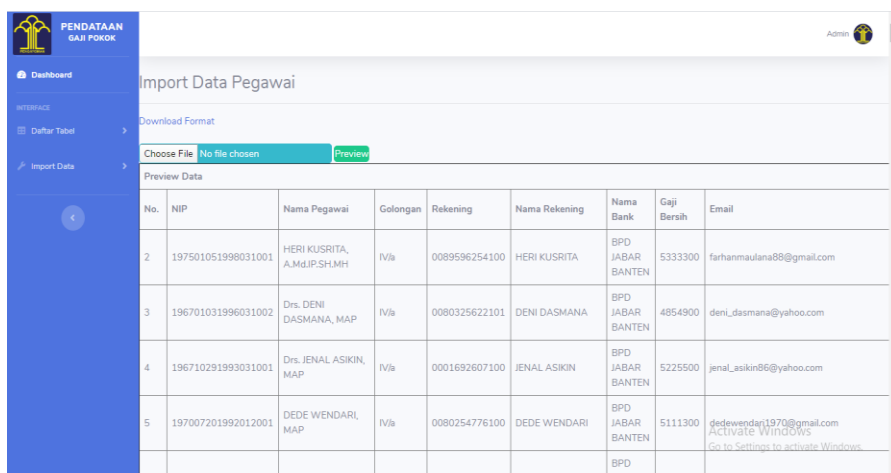
4.4.5 Halaman Pengambilan Data Pegawai (1)



Gambar 8. Halaman Pengambilan Data Pegawai (1)

Halaman pengambilan data pegawai pada gambar 8 adalah halaman yang akan mengambil data seluruh pegawai Rumah Tahanan Negara Klas 1 Bandung dari *excel* dengan *type* .xlsx yang sebelumnya sudah dibuatkan *template* khusus.

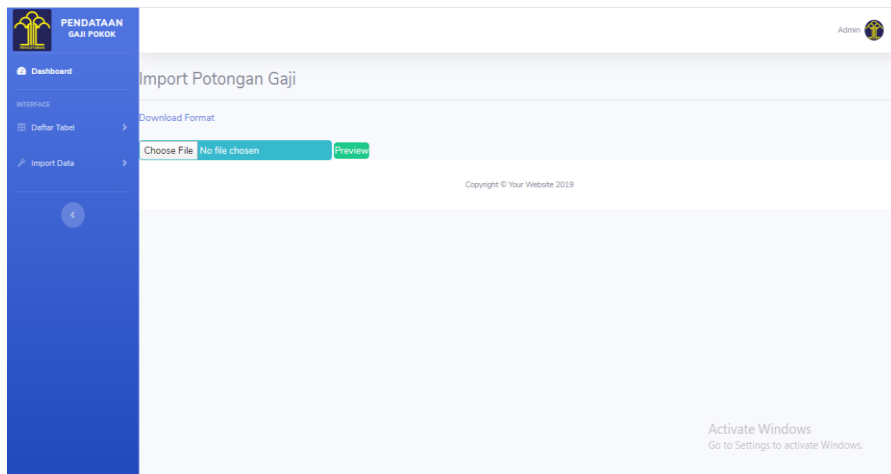
4.4.6 Halaman Pengambilan Data Pegawai (2)



Gambar 9. Halaman Pengambilan Data Pegawai (2)

Halaman pengambilan data pegawai pada gambar 9 adalah halaman yang akan menampilkan keseluruhan data yang sebelumnya sudah diambil dari file *excel* kemudian seluruh data pegawai tersebut akan dimasukkan kedalam aplikasi.

4.4.7 Halaman Pengambilan Data Potongan (1)



Gambar 10. Halaman Pengambilan Data Potongan (1)

Halaman pengambilan data potongan pada gambar 10 ini memiliki fungsi sama seperti gambar 8 tapi halaman ini akan mengambil seluruh data potongan dari gaji pegawai Rumah Tahanan Negara Klas 1 Bandung dari *excel* dengan *type* *.xlsx* yang sebelumnya sudah dibuatkan *template* khusus.

4.4.8 Halaman Pengambilan Data Potongan (2)

NIP	Bulan	Arisan Pengayoman	IDW	Koperasi	Tabungan DW	Olahraga	BRI	BJB	SKP	Lain-lain	Arisan DW
197501051998031001	Aug-19	120000	25000	200000		1000			1500		105000
196701031996031002	Aug-19		10000	2142000		1000			1500		105000
196710291993031001	Aug-19		5000	200000		1000			1500		105000
197007201992012001	Aug-19		5000	700000		1000			1500	4015573	105000
197306072008012001	Aug-19		5000	1753000		1000			1500		105000
198212142002121002	Aug-19	120000	15000	200000		1000			1500		105000
198805232007011005	Aug-19	120000	15000	393800		1000			1500		105000
196407271985032001	Aug-19	120000	10000	1795500		1000		2796261	1500		105000
198608062007031003	Aug-19	120000	10000	2585200		1000			1500		105000

Gambar 11. Halaman Pengambilan Data Potongan (2)

Halaman pengambilan data potongan pada gambar 11 memiliki fungsi sama seperti gambar 9 tapi halaman ini akan menampilkan keseluruhan data potongan gaji pegawai yang sebelumnya sudah diambil dari file *excel* yang nantinya seluruh data potongan pegawai tersebut akan dimasukkan kedalam aplikasi.

A. Penjelasan Fungsi Element HTML Disetiap Halaman

1. Halaman Login

a. *Textbox username dan password*

User name	Password
-----------	----------

Gambar 12. *Textbox username dan password*

Textbox pada gambar 12 adalah *textbox* yang akan diisi oleh pegawai keuangan Rumah Tahanan Klas 1

Bandung dan sekaligus menampung *username* dan *password* yang akan digunakan sebagai kunci masuk kedalam aplikasi.

b. *Button Sign In*



Gambar 13. *Button Sign In*

Button pada gambar 13 adalah sebagai akses permintaan masuk pada program yang nanti akan mengirimkan hasil yang sudah diisikan oleh pegawai keuangan di *textbox username* dan *password* kedalam aplikasi, apabila *username* dan *password* sesuai maka pegawai keuangan dapat menjalankan seluruh fungsi yang terdapat didalam aplikasi.

2. Halaman Tabel Pegawai

a. *Button Edit dan Update*



Gambar 14. *Button Edit*

Button pada gambar 14 memiliki fungsi untuk menampilkan *popup* yang akan berisi data pribadi dari pegawai Rumah Tahanan Klas 1 Bandung yang sebelumnya sudah dipilih seperti pada gambar 15.

Golongan	III/b
Rekening	0002330571100
Nama Rekening	YAN ROSBIN PURBA
Nama Bank	BPD JABAR BANTEN
Email	yanrosbinpurba50@gmail.com
Gaji Bersih	4805700
Status	AKTIF

Tutup Update

Gambar 15. *Pop-up Edit* dan *Button Update*

Gambar 15 adalah hasil *popup* dari ditekannya *button edit* gambar 14 yang menampilkan data pribadi pegawai Rumah Tahanan Klas 1 Bandung dan *button update* akan membawa seluruh data pribadi pegawai yang terdapat pada *textbox popup* untuk kemudian dilakukan fungsi *update* pada data tersebut.

b. *Button Delete All*



Gambar 16. *Button Delete All*

Button pada gambar 16 memiliki fungsi untuk menghapus seluruh data pegawai Rumah Tahanan Klas 1 Bandung, namun seluruh disini tidak termasuk dengan data potongan dan pinjaman.

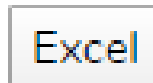
c. *Button Export PDF*



Gambar 17. *Button Export PDF*

Button pada gambar 17 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 22 didalam halaman tabel pegawai atau pada gambar 2 menjadi format PDF.

d. *Button Export Excel*



Gambar 18. *Button Export Excel*

Button pada gambar 18 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 22 didalam halaman tabel pegawai atau pada gambar 2 menjadi format *excel*.

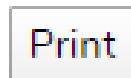
e. *Button Export CSV*



Gambar 19. *Button Export CSV*

Button pada gambar 19 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 22 didalam halaman tabel pegawai atau pada gambar 2 menjadi format CSV.

f. *Button Print*



Gambar 20. *Button Print*

Button pada gambar 20 memiliki fungsi untuk mencetak secara langsung pada aplikasi seluruh data pribadi pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 22.

g. *Button Copy*



Gambar 21. *Button Copy*

Button pada gambar 21 memiliki fungsi untuk menyalin seluruh data pribadi pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 22.

h. *Table Pegawai*

No.	NIP	Nama Pegawai	Golongan	Rekening	Nama Rekening	Nama Bank	Email	Gaji Br
1	197501051998031001	HERI KUSRITA, A.Md.IP.SH.MH	IV/a	0089596254100	HERI KUSRITA	BPD JABAR BANTEN	farhanmaulana88@gmail.com	Rp5.33:
2	196701031996031002	Drs. DENI DASMANA, MAP	IV/a	0080325622101	DENI DASMANA	BPD JABAR BANTEN	deni_dasmana@yahoo.com	Rp4.85:
3	196710291993031001	Drs. JENAL ASIKIN, MAP	IV/a	0001692607100	JENAL ASIKIN	BPD JABAR BANTEN	jena_lasikin86@yahoo.com	Rp5.22:
4	197007201992012001	DEDE WENDARI, MAP	IV/a	0080254776100	DEDE WENDARI	BPD JABAR BANTEN	dedewendari1970@gmail.com	Rp5.11:
5	197306072008012001	dr. TIKA KARTIKA	IV/a	0080325924100	TIKA KARTIKA	BPD JABAR BANTEN	tikapuang@gmail.com	Rp4.46:
6	198212142002121002	MASHURI ALWI, A.MD.IP	III/c	0080490151101	MASHURI ALWI	BPD JABAR BANTEN	mashurialwi@gmail.com	Rp4.10:

Gambar 22. *Table Pegawai*

Table pada gambar 22 memiliki fungsi untuk menampilkan seluruh data pribadi pegawai Rumah Tahanan Klas 1 Bandung, maksud dari seluruh data tersebut tidak mencakup data potongan gaji dan data pinjaman.

i. *Button Previous, Number, dan Next*



Gambar 23. *Button Previous, Number, dan Next*

Seluruh *button* pada gambar 23 umumnya memiliki fungsi yang sama yaitu sebagai navigasi menampilkan data pegawai pada tabel di gambar 22. Untuk lebih spesifik *button previous* digunakan untuk memundurkan halaman dari tabel di gambar 22, kemudian *button number* digunakan untuk memilih halaman dari tabel pada gambar 22 sesuai yang

diinginkan pegawai keuangan, dan terakhir *button next* digunakan untuk memajukan halaman dari tabel di gambar 22.

j. *Button Up*



Gambar 24. *Button Up*

Button pada gambar 24 memiliki fungsi untuk mengembalikan halaman tabel pegawai pada gambar 2 kembali ke bagian atas.

k. *Textbox Search*



Gambar 25. *Textbox Search*

Textbox pada gambar 25 memiliki fungsi untuk melakukan pencarian data secara menyeluruh kepada seluruh data didalam tabel gambar 22 sesuai dengan yang diisikan pada *textbox* tersebut.

3. Halaman Tabel Potongan

a. *Button Email*



Gambar 26. *Button Email*

Button email pada gambar 26 memiliki fungsi untuk mengirimkan data potongan gaji pegawai sesuai dengan data potongan yang dipilih kepada pegawai yang bersangkutan melalui email yang sebelumnya sudah di data pada tabel di gambar 22.

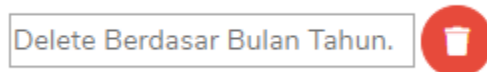
b. *Button Delete All*



Gambar 27. *Button Delete All*

Button pada gambar 27 memiliki fungsi untuk menghapus seluruh data potongan gaji pegawai Rumah Tahanan Klas 1 Bandung, namun seluruh disini tidak termasuk dengan data pribadi pegawai dan data pinjaman.

c. *Textbox dan Button Delete*



Gambar 27. *Textbox dan Button Delete*

Textbox dan *button delete* pada gambar 27 ini sangat berhubungan karena *button delete* memiliki fungsi untuk menghapus data potongan gaji pegawai Rumah Tahanan Klas 1 Bandung sesuai dengan bulan yang telah diisikan didalam *textbox*.

d. *Button* Pemilihan Bulan



Gambar 27. *Button* Pemilihan Bulan

Button pemilihan bulan pada gambar 27 memiliki fungsi menampilkan halaman potongan gaji pegawai berdasarkan bulan pilhan seperti pada gambar 5. Sebagai contohnya pada gambar 27 terdapat 2 *button* pemilihan bulan yang mana apabila memilih *button* Aug-19 maka seluruh data potongan gaji pegawai pada bulan Agustus 2019 akan ditampilkan.

e. *Button* Export PDF

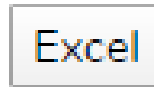


Gambar 28. *Button* Export PDF

Button pada gambar 28 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di

gambar 33 didalam halaman tabel potongan atau pada gambar 3 menjadi format PDF.

f. Button Export Excel



Gambar 29. *Button Export Excel*

Button pada gambar 29 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 33 didalam halaman tabel potongan atau pada gambar 3 menjadi format *excel*.

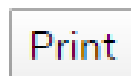
g. Button Export CSV



Gambar 30. *Button Export CSV*

Button pada gambar 30 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 33 didalam halaman tabel potongan atau pada gambar 3 menjadi format CSV.

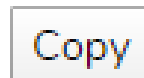
h. Button Print



Gambar 31. *Button Print*

Button pada gambar 31 memiliki fungsi untuk mencetak secara langsung pada aplikasi seluruh data potongan gaji pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 33.

i. *Button Copy*



Gambar 32. *Button Copy*

Button pada gambar 32 memiliki fungsi untuk menyalin seluruh data potongan gaji pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 33.

j. *Table Potongan*

No.	Nama Pegawai	Bulan Tahun	Arisan Pengayoman	IDW	Koperasi	Tabungan DW	Olahraga	BRI	BJB	SKP	Lain-lain	Aris DW
1	HERI KUSRITA, A.Md.IP.SH.MH	Aug-19	120.000	25.000	200.000	0	1.000	0	0	1.500	0	105.000
2	Drs. DENI DASMANA, MAP	Aug-19	0	10.000	2.142.000	0	1.000	0	0	1.500	0	105.000
3	Drs. JENAL ASIKIN, MAP	Aug-19	0	5.000	200.000	0	1.000	0	0	1.500	0	105.000
4	DEDE WENDARI, MAP	Aug-19	0	5.000	700.000	0	1.000	0	0	1.500	4.015.573	105.000
5	dr. TIKA KARTIKA	Aug-19	0	5.000	1.753.000	0	1.000	0	0	1.500	0	105.000
6	MASHURI ALWI, A.MD.IP	Aug-19	120.000	15.000	200.000	0	1.000	0	0	1.500	0	105.000

Gambar 33. *Table Potongan*

Table pada gambar 33 memiliki fungsi untuk menampilkan seluruh data pribadi pegawai Rumah Tahanan

Klas 1 Bandung, maksud dari seluruh data tersebut tidak mencakup data potongan gaji dan data pinjaman.

k. *Button Previous, Number, dan Next*



Gambar 34. *Button Previous, Number, dan Next*

Seluruh *button* pada gambar 34 umumnya memiliki fungsi yang sama yaitu sebagai navigasi menampilkan data potongan pegawai pada tabel di gambar 33. Untuk lebih spesifik *button previous* digunakan untuk memundurkan halaman dari tabel di gambar 33, kemudian *button number* digunakan untuk memilih halaman dari tabel pada gambar 33 sesuai yang diinginkan pegawai keuangan, dan terakhir *button next* digunakan untuk memajukan halaman dari tabel di gambar 33.

l. *Button Up*



Gambar 35. *Button Up*

Button pada gambar 35 memiliki fungsi untuk mengembalikan halaman tabel potongan pada gambar 3 kembali ke bagian atas.

m. Textbox Search

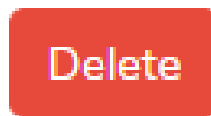


Gambar 36. *Textbox Search*

Textbox pada gambar 36 memiliki fungsi untuk melakukan pencarian data secara menyeluruh kepada seluruh data didalam tabel gambar 33 sesuai dengan yang diisikan pada *textbox* tersebut.

4. Halaman Tabel Pinjaman

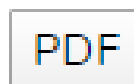
a. Button Delete



Gambar 37. *Button Delete*

Button delete pada gambar 37 memiliki fungsi untuk menghapus data pinjaman pegawai Rumah Tahanan Klas 1 Bandung sesuai dengan data pinjaman yang sebelumnya sudah dipilih.

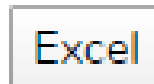
b. Button Export PDF



Gambar 38. *Button Export PDF*

Button pada gambar 38 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 43 didalam halaman tabel pinjaman atau pada gambar 4 menjadi format PDF.

c. *Button Export Excel*



Gambar 39. *Button Export Excel*

Button pada gambar 39 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 43 didalam halaman tabel pinjaman atau pada gambar 4 menjadi format *excel*.

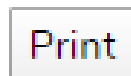
d. *Button Export CSV*



Gambar 40. *Button Export CSV*

Button pada gambar 40 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 43 didalam halaman tabel pinjaman atau pada gambar 4 menjadi format CSV.

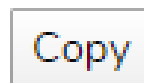
e. *Button Print*



Gambar 41. *Button Print*

Button pada gambar 41 memiliki fungsi untuk mencetak secara langsung pada aplikasi seluruh data pinjaman gaji pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 43.

f. *Button Copy*



Gambar 42. *Button Copy*

Button pada gambar 42 memiliki fungsi untuk menyalin seluruh data pinjaman gaji pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 43.

g. *Table Pinjaman*

No.	NIP	Nama Pegawai	Gaji Kotor	Bulan Peminjaman	Plafond	Lama Bulan	Angsuran Bulanan	Total	Status
1	197501051998031001	HERI KUSRITA, A.Md.IP.SH.MH	Rp5.333.300	Aug-19	Rp10.000.000	11	888488	Rp9.773.368	BELUM LUNAS
2	196701031996031002	Drs. DENI DASMANA, MAP	Rp4.854.900	Sep-19	Rp10.000.000	36	332143	Rp11.957.148	BELUM LUNAS
3	196710291993031001	Drs. JENAL ASIKIN, MAP	Rp5.225.500	Sep-19	Rp20.000.000	72	391004	Rp28.152.288	BELUM LUNAS
4	197007201992012001	DEDE WENDARI, MAP	Rp5.111.300	Sep-19	Rp60.000.000	156	761200	Rp118.747.200	BELUM LUNAS
5	197306072008012001	dr. TIKA KARTIKA	Rp4.461.900	Sep-19	Rp40.000.000	72	782008	Rp56.304.576	BELUM LUNAS

Gambar 43. *Table Pinjaman*

Table pada gambar 43 memiliki fungsi untuk menampilkan seluruh data pinjaman pegawai Rumah Tahanan

Klas 1 Bandung, maksud dari seluruh data tersebut tidak mencakup data potongan gaji dan data pegawai.

h. *Button Previous, Number, dan Next*



Gambar 44. *Button Previous, Number, dan Next*

Seluruh *button* pada gambar 44 umumnya memiliki fungsi yang sama yaitu sebagai navigasi menampilkan data pinjaman pegawai pada tabel di gambar 33. Untuk lebih spesifik *button previous* digunakan untuk memundurkan halaman dari tabel di gambar 43, kemudian *button number* digunakan untuk memilih halaman dari tabel pada gambar 43 sesuai yang diinginkan pegawai keuangan, dan terakhir *button next* digunakan untuk memajukan halaman dari tabel di gambar 43.

i. *Button Up*



Gambar 45. *Button Up*

Button pada gambar 55 memiliki fungsi untuk mengembalikan halaman tabel pinjaman pada gambar 4 kembali ke bagian atas.

j. *Textbox Search*



Gambar 46. *Textbox Search*

Textbox pada gambar 46 memiliki fungsi untuk melakukan pencarian data secara menyeluruh kepada seluruh data didalam tabel gambar 33 sesuai dengan yang diisikan pada *textbox* tersebut.

5. Halaman Detail Potongan Berdasarkan Bulan

a. *Button Email*



Gambar 47. *Button Email*

Button email pada gambar 47 memiliki fungsi untuk mengirimkan data potongan gaji pegawai sesuai dengan data potongan yang dipilih melalui *combo box* pilihan pada gambar 48 kepada pegawai melalui *email* yang sebelumnya sudah di data pada tabel di gambar 54.

b. *Combo Box Pilihan*



Select
All

Gambar 48. *Combo Box* Pilihan

Combo box pilihan pada gambar 48 memiliki fungsi untuk memilih beberapa atau seluruh data pinjaman berdasarkan bulan yang akan dikirimkan ke masing-masing pegawai melalui *email*.

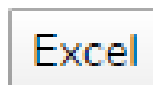
c. *Button Export PDF*



Gambar 49. *Button Export PDF*

Button pada gambar 49 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 54 didalam halaman tabel potongan berdasarkan bulan atau pada gambar 5 menjadi format PDF.

d. *Button Export Excel*



Gambar 50. *Button Export Excel*

Button pada gambar 50 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 54 didalam halaman tabel potongan berdasarkan bulan atau pada gambar 5 menjadi format *excel*.

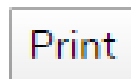
e. *Button Export CSV*



Gambar 51. *Button Export CSV*

Button pada gambar 51 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 54 didalam halaman tabel potongan berdasarkan bulan atau pada gambar 5 menjadi format CSV.

f. *Button Print*



Gambar 52. *Button Print*

Button pada gambar 52 memiliki fungsi untuk mencetak secara langsung pada aplikasi seluruh data potongan gaji berdasarkan bulan pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 54.

g. *Button Copy*



Gambar 53. *Button Copy*

Button pada gambar 53 memiliki fungsi untuk menyalin seluruh data potongan gaji berdasarkan bulan pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 54.

h. Table Potongan Berdasarkan Bulan

<input type="checkbox"/> Select All	No.	Nama Pegawai	Bulan Tahun	Arisan Pengayoman	IDW	Koperasi	Tabungan DW	Olahraga	BRI	BJB	SKP
<input type="checkbox"/>	1	HERI KUSRITA, A.Md.IP.SHLMH	Aug-19	120.000	25.000	200.000	0	1.000	0	0	1.500
<input type="checkbox"/>	2	Drs. DENI DASMANA, MAP	Aug-19	0	10.000	2.142.000	0	1.000	0	0	1.500
<input type="checkbox"/>	3	Drs. JENAL ASIKIN, MAP	Aug-19	0	5.000	200.000	0	1.000	0	0	1.500
<input type="checkbox"/>	4	DEDE WENDARI, MAP	Aug-19	0	5.000	700.000	0	1.000	0	0	1.500
<input type="checkbox"/>	5	dr. TIKA KARTIKA	Aug-19	0	5.000	1.753.000	0	1.000	0	0	1.500
<input type="checkbox"/>	6	MASHURI ALWI, A.MD.IP	Aug-19	120.000	15.000	200.000	0	1.000	0	0	1.500

Gambar 54. *Table Potongan Berdasarkan Bulan*

Table pada gambar 54 memiliki fungsi untuk menampilkan data potongan berdasarkan bulan pegawai Rumah Tahanan Klas 1 Bandung.

i. Button Previous, Number, dan Next



Gambar 55. *Button Previous, Number, dan Next*

Seluruh *button* pada gambar 55 umumnya memiliki fungsi yang sama yaitu sebagai navigasi menampilkan data potongan berdasarkan bulan pegawai pada tabel di gambar 54. Untuk lebih spesifik *button previous* digunakan untuk memundurkan halaman dari tabel di gambar 54, kemudian *button number* digunakan untuk memilih halaman dari tabel pada gambar 54 sesuai yang diinginkan pegawai keuangan, dan terakhir *button next* digunakan untuk memajukan halaman dari tabel di gambar 54.

j. *Button Up*



Gambar 56. *Button Up*

Button pada gambar 56 memiliki fungsi untuk mengembalikan halaman tabel potongan berdasarkan bulan pada gambar 5 kembali ke bagian atas.

k. *Textbox Search*



Gambar 57. *Textbox Search*

Textbox pada gambar 57 memiliki fungsi untuk melakukan pencarian data secara menyeluruh kepada seluruh

data didalam tabel gambar 54 sesuai dengan yang diisikan pada *textbox* tersebut.

6. Halaman Detail Potongan Gaji Pegawai

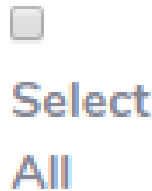
a. *Button Email*



Gambar 58. *Button Email*

Button email pada gambar 58 memiliki fungsi untuk mengirimkan data potongan gaji pegawai sesuai dengan data potongan yang dipilih melalui *combo box* pilihan pada gambar 59 kepada pegawai melalui *email* yang sebelumnya sudah di data pada tabel di gambar 68.

b. *Combo Box Pilihan*



Gambar 59. *Combo Box Pilihan*

Combo box pilihan pada gambar 59 memiliki fungsi untuk memilih beberapa atau seluruh data detail potongan pegawai yang akan dikirimkan melalui *email*.

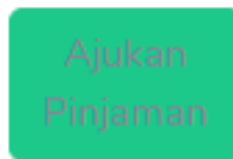
c. *Button Pemilihan Bulan*



Gambar 60. *Button* Pemilihan Bulan

Button pemilihan bulan pada gambar 60 memiliki fungsi menampilkan halaman potongan gaji pegawai berdasarkan bulan pilhan seperti pada gambar 5. Sebagai contohnya pada gambar 60 terdapat 2 *button* pemilihan bulan yang mana apabila memilih *button* Aug-19 maka seluruh data potongan gaji pegawai pada bulan Agustus 2019 akan ditampilkan.

d. *Button* Ajukan Pinjaman dan Cek Data



Gambar 61. *Button* Ajukan Pinjaman

Button ajukan pinjaman pada gambar 61 memiliki fungsi untuk menampilkan *popup* yang akan berisi data kalkulasi dari gaji yang diterima dan beberapa potongan khusus pegawai Rumah Tahanan Klas 1 Bandung yang sebelumnya sudah dipilih seperti pada gambar 62.

Keterangan

X

NIP	Nama Pegawai
198805232007011005	ALVIANTINO RISKI SATRIY
Koperasi	BRI
Rp. 393.800	Rp. 0
BJB	Gaji Diterima
Rp. 0	Rp. 3.035.200
Lain-lain	Hasil
Rp. 0	Rp. 3.229.000

Tutup

Cek Data

Gambar 62. *Popup* dan *Button* Cek Data

Gambar 62 adalah hasil *popup* dari ditekannya *button* ajukan pinjaman gambar 61 yang menampilkan data kalkulasi dari gaji yang diterima dan beberapa potongan khusus pegawai Rumah Tahanan Klas 1 Bandung dan *button* cek data akan membawa data di *textbox* hasil untuk mulai dimasukan kedalam perhitungan metode *simple additive weighting* (SAW).

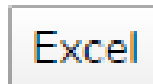
e. *Button Export PDF*



Gambar 63. *Button Export PDF*

Button pada gambar 63 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 68 didalam halaman tabel detail potongan pegawai atau pada gambar 6 menjadi format PDF.

f. *Button Export Excel*



Gambar 64. *Button Export Excel*

Button pada gambar 64 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 68 didalam halaman tabel detail potongan pegawai atau pada gambar 6 menjadi format *excel*.

g. *Button Export CSV*



Gambar 65. *Button Export CSV*

Button pada gambar 65 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di

gambar 68 didalam halaman tabel detail potongan pegawai atau pada gambar 6 menjadi format CSV.

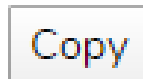
h. Button Print



Gambar 66. *Button Print*

Button pada gambar 66 memiliki fungsi untuk mencetak secara langsung pada aplikasi seluruh data detail potongan gaji pegawai berdasarkan bulan pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 68.

i. Button Copy



Gambar 67. *Button Copy*

Button pada gambar 67 memiliki fungsi untuk menyalin seluruh data detail potongan gaji pegawai pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 68.

j. Table Detail Potongan Gaji Pegawai

Select All	No.	Bulan Tahun	Arisan Pengayoman	IDW	Koperasi	Tabungan DW	Olahraga	BRI	BJB	SKP	Lain-lain	Arisan DW
<input type="checkbox"/>	1	Aug-19	120.000	25.000	200.000	0	1.000	0	0	1.500	0	105.000
<input type="checkbox"/>	2	Sep-19	120.000	25.000	200.000	0	1.000	0	4.707,347	1.500	0	105.000
<input type="checkbox"/>	3	Oct-19	120.000	25.000	200.000	0	1.000	0	4.707,347	1.500	0	105.000

Gambar 68. *Table Detail Potongan Gaji Pegawai*

Table pada gambar 68 memiliki fungsi untuk menampilkan data detail potongan gaji pegawai berdasarkan bulan pegawai Rumah Tahanan Klas 1 Bandung.

k. *Button Previous, Number, dan Next*



Gambar 69. *Button Previous, Number, dan Next*

Seluruh *button* pada gambar 69 umumnya memiliki fungsi yang sama yaitu sebagai navigasi menampilkan data detail potongan gaji pegawai pegawai pada tabel di gambar 68. Untuk lebih spesifik *button previous* digunakan untuk memundurkan halaman dari tabel di gambar 68, kemudian *button number* digunakan untuk memilih halaman dari tabel pada gambar 68 sesuai yang diinginkan pegawai keuangan, dan terakhir *button next* digunakan untuk memajukan halaman dari tabel di gambar 68.

l. Button Up



Gambar 70. *Button Up*

Button pada gambar 70 memiliki fungsi untuk mengembalikan halaman tabel detail potongan gaji pegawai pada gambar 6 kembali ke bagian atas.

m. Textbox Search



Gambar 71. *Textbox Search*

Textbox pada gambar 71 memiliki fungsi untuk melakukan pencarian data secara menyeluruh kepada seluruh data didalam tabel gambar 68 sesuai dengan yang diisikan pada *textbox* tersebut.

7. Halaman Pemberian Rekomendasi Pinjaman

a. Button Pinjam



Gambar 72. *Button Pinjam*

Button pinjam pada gambar 72 memiliki fungsi untuk menyimpan data pinjaman sesuai dengan pilihan pegawai Rumah Tahanan Negara Klas 1 Bandung yang akan melakukan pinjaman, data yang bisa dipilih pegawai Rumah Tahanan Negara Klas 1 Bandung ada pada tabel di gambar 78.

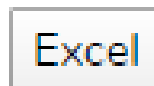
b. *Button Export PDF*



Gambar 73. *Button Export PDF*

Button pada gambar 73 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 78 didalam halaman tabel rekomendasi pinjaman atau pada gambar 7 menjadi format PDF.

c. *Button Export Excel*



Gambar 74. *Button Export Excel*

Button pada gambar 74 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 78 didalam halaman tabel rekomendasi pinjaman atau pada gambar 7 menjadi format *excel*.

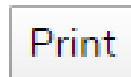
d. *Button Export CSV*



Gambar 75. *Button Export CSV*

Button pada gambar 75 memiliki fungsi untuk melakukan *export* seluruh data yang ditampilkan pada tabel di gambar 78 didalam halaman tabel rekomendasi pinjaman atau pada gambar 7 menjadi format CSV.

e. *Button Print*



Gambar 76. *Button Print*

Button pada gambar 76 memiliki fungsi untuk mencetak secara langsung pada aplikasi seluruh tabel rekomendasi pinjaman pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 78.

f. *Button Copy*



Gambar 77. *Button Copy*

Button pada gambar 77 memiliki fungsi untuk menyalin seluruh data detail tabel rekomendasi pinjaman pegawai pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 78.

g. *Table Rekomendasi Pinjaman*

Select All	No.	Bulan Tahun	Arisan Pengayoman	IDW	Koperasi	Tabungan DW	Olahraga	BRI	BJB	SKP	Lain-lain	Arisan DW
<input type="checkbox"/>	1	Aug-19	120.000	25.000	200.000	0	1.000	0	0	1.500	0	105.000
<input type="checkbox"/>	2	Sep-19	120.000	25.000	200.000	0	1.000	0	4.707,347	1.500	0	105.000
<input type="checkbox"/>	3	Oct-19	120.000	25.000	200.000	0	1.000	0	4.707,347	1.500	0	105.000

Gambar 78. *Table Rekomendasi Pinjaman*

Table pada gambar 78 memiliki fungsi untuk menampilkan data detail potongan gaji pegawai berdasarkan bulan pegawai Rumah Tahanan Klas 1 Bandung.

h. *Button Previous, Number, dan Next*



Gambar 79. *Button Previous, Number, dan Next*

Seluruh *button* pada gambar 79 umumnya memiliki fungsi yang sama yaitu sebagai navigasi menampilkan data tabel rekomendasi pinjaman pegawai pada tabel di gambar 78. Untuk lebih spesifik *button previous* digunakan untuk memundurkan halaman dari tabel di gambar 78, kemudian *button number* digunakan untuk memilih halaman dari tabel pada gambar 78 sesuai yang diinginkan pegawai keuangan, dan terakhir *button next* digunakan untuk memajukan halaman dari tabel di gambar 78.

i. *Button Up*



Gambar 79. *Button Up*

Button pada gambar 79 memiliki fungsi untuk mengembalikan halaman tabel detail potongan gaji pegawai pada gambar 7 kembali ke bagian atas.

j. *Textbox Search*



Gambar 80. *Textbox Search*

Textbox pada gambar 80 memiliki fungsi untuk melakukan pencarian data secara menyeluruh kepada seluruh data didalam tabel gambar 78 sesuai dengan yang diisikan pada *textbox* tersebut.

8. Halaman Pengambilan Data Pegawai (1)

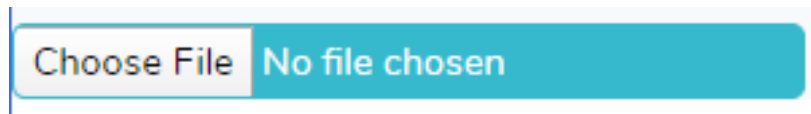
a. *Button Download Format*



Gambar 81. *Button Download Format*

Button download format pada gambar 81 memiliki fungsi untuk menyimpan file yang akan di *import* kedalam aplikasi sesuai direktori yang telah ditentukan pada kodingan.

b. *Button Choose File*



Gambar 81. *Button Choose File*

Button pada gambar 81 memiliki fungsi untuk memilih file *excel* yang akan di *import* kan pada aplikasi dari direktori yang ada didalam *device*.

c. *Button Preview*



Gambar 82. *Button Preview*

Button pada gambar 82 memiliki fungsi untuk menampilkan seluruh data pegawai dari file *excel* yang telah dipilih pada tabel di halaman pengambilan data pegawai (2) atau pada gambar 9.

9. Halaman Pengambilan Data Pegawai (2)

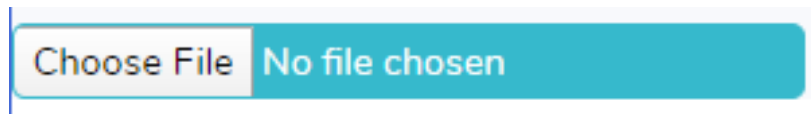
a. *Button Download Format*



Gambar 83. *Button Download Format*

Button download format pada gambar 83 memiliki fungsi untuk menyimpan file yang akan di *import* kedalam aplikasi sesuai direktori yang telah ditentukan pada kodingan.

b. *Button Choose File*



Gambar 84. *Button Choose File*

Button pada gambar 84 memiliki fungsi untuk memilih file *excel* yang akan di *import* kan pada aplikasi dari direktori yang ada didalam *device*.

c. *Button Preview*



Gambar 85. *Button Preview*

Button pada gambar 85 memiliki fungsi untuk menampilkan seluruh data pegawai dari file *excel* yang telah dipilih pada tabel di halaman pengambilan data pegawai (2) atau pada gambar 9.

d. *Table Data Pegawai Import*

Preview Data							
No.	NIP	Nama Pegawai	Golongan	Rekening	Nama Rekening	Nama Bank	Gaji Bersih
2	197501051998031001	HERI KUSRITA, A.Md.IP.SH.MH	IV/a	0089596254100	HERI KUSRITA	BPD JABAR BANTEN	5333300 farhanmaulana88@gmail.com
3	196701031996031002	Drs. DENI DASMANA, MAP	IV/a	0080325622101	DENI DASMANA	BPD JABAR BANTEN	4854900 deni_dasmana@yahoo.com
4	196710291993031001	Drs. JENAL ASIKIN, MAP	IV/a	0001692607100	JENAL ASIKIN	BPD JABAR BANTEN	5225500 jenal_asikin86@yahoo.com
5	197007201992012001	DEDE WENDARI, MAP	IV/a	0080254776100	DEDE WENDARI	BPD JABAR BANTEN	5111300 dedewendari1970@gmail.com

Gambar 86. *Table Data Pegawai Import*

Table data pegawai import pada gambar 86 memiliki fungsi untuk menampilkan seluruh data pegawai dari file *excel* setelah *button preview* pada gambar 85 yang telah dipilih pada tabel di halaman pengambilan data pegawai (2) atau pada gambar 9.

e. *Button Import*



Gambar 87. *Button Import*

Button import pada gambar 87 adalah untuk memasukan seluruh data pribadi pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 86 yang selanjutnya data pribadi pegawai Rumah Tahanan Klas 1 Bandung akan ditampilkan pada halaman tabel pegawai pada gambar 2.

f. Button Cancel



Gambar 88. *Button Cancel*

Button pada gambar 88 memiliki fungsi untuk menggagalkan *import* dari tabel pada gambar 86 dan setelah itu akan di arahkan ke halaman tabel pegawai pada gambar 2.

g. Button Up



Gambar 89. *Button Up*

Button pada gambar 89 memiliki fungsi untuk mengembalikan halaman tabel data pribadi yang akan di *import* dari pegawai pada gambar 9 kembali ke bagian atas.

10. Halaman Pengambilan Data Potongan (1)

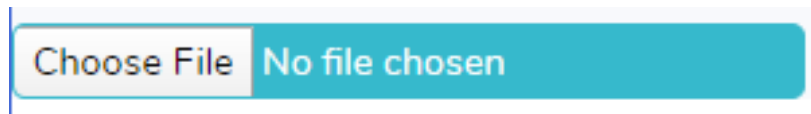
a. Button Download Format



Gambar 90. *Button Download Format*

Button download format pada gambar 90 memiliki fungsi untuk menyimpan file yang akan di *import* kedalam aplikasi sesuai direktori yang telah ditentukan pada kodingan.

b. *Button Choose File*



Gambar 91. *Button Choose File*

Button pada gambar 91 memiliki fungsi untuk memilih file *excel* yang akan di *import* kan pada aplikasi dari direktori yang ada didalam *device*.

c. *Button Preview*



Gambar 92. *Button Preview*

Button pada gambar 92 memiliki fungsi untuk menampilkan seluruh data potongan dari file *excel* yang telah dipilih pada tabel di halaman pengambilan data potongan (2) atau pada gambar 10.

11. Halaman Pengambilan Data Potongan (2)

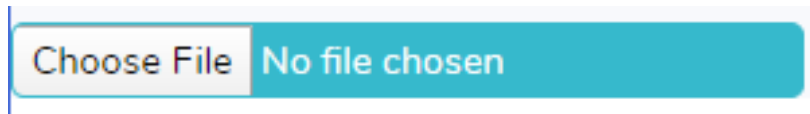
a. *Button Download Format*



Gambar 93. *Button Download Format*

Button download format pada gambar 93 memiliki fungsi untuk menyimpan file yang akan di *import* kedalam aplikasi sesuai direktori yang telah ditentukan pada kodingan.

b. *Button Choose File*



Gambar 94. *Button Choose File*

Button pada gambar 94 memiliki fungsi untuk memilih file *excel* yang akan di *import* kan pada aplikasi dari direktori yang ada didalam *device*.

c. *Button Preview*



Gambar 95. *Button Preview*

Button pada gambar 95 memiliki fungsi untuk menampilkan seluruh data pegawai dari file *excel* yang telah dipilih pada tabel di halaman pengambilan data potongan (2) atau pada gambar 11.

d. *Table Data Potongan Import*

Preview Data								
No.	NIP	Nama Pegawai	Golongan	Rekening	Nama Rekening	Nama Bank	Gaji Bersih	Email
2	197501051998031001	HERI KUSRITA, A.Md.IP.SH.MH	IV/a	0089596254100	HERI KUSRITA	BPD JABAR BANTEN	5333300	farhanmaulana88@gmail.com
3	196701031996031002	Drs. DENI DASMANA, MAP	IV/a	0080325622101	DENI DASMANA	BPD JABAR BANTEN	4854900	deni_dasmana@yahoo.com
4	196710291993031001	Drs. JENAL ASIKIN, MAP	IV/a	0001692607100	JENAL ASIKIN	BPD JABAR BANTEN	5225500	jenal_asikin86@yahoo.com
5	197007201992012001	DEDE WENDARI, MAP	IV/a	0080254776100	DEDE WENDARI	BPD JABAR BANTEN	5111300	dedewendari1970@gmail.com

Gambar 96. *Table Data Potongan Import*

Table data pegawai import pada gambar 96 memiliki fungsi untuk menampilkan seluruh data potongan dari file *excel* setelah *button preview* pada gambar 95 yang telah dipilih pada tabel di halaman pengambilan data potongan (2) atau pada gambar 11.

e. *Button Import*



Gambar 97. *Button Import*

Button import pada gambar 97 adalah untuk memasukan seluruh data potongan pegawai Rumah Tahanan Klas 1 Bandung pada tabel di gambar 96 yang selanjutnya data potongan pegawai Rumah Tahanan Klas 1 Bandung akan ditampilkan pada halaman tabel potongan pada gambar 3.

f. Button Cancel



Gambar 98. *Button Cancel*

Button pada gambar 98 memiliki fungsi untuk menggagalkan *import* dari tabel pada gambar 96 dan setelah itu akan di arahkan ke halaman tabel potongan pegawai pada gambar 3.

g. Button Up



Gambar 99. *Button Up*

Button pada gambar 99 memiliki fungsi untuk mengembalikan halaman tabel data potongan pegawai yang akan di *import* dari pegawai pada gambar 9 kembali ke bagian atas.

B. Penjelasan Kodingan *Controller* Perbaris

1. *Controller Login*

a. Controller Class Login

```

1 <?php
2 Class Login extends CI_Controller{
3     private $filename = "import_data";
4
5     function __construct() {
6         parent::__construct();
7         $this->load->library('session');
8         $this->load->library('pagination');
9         $this->load->helper('form');
10        $this->load->helper('url');
11        $this->load->model('ModelLogin');
12        $this->load->model('ModelAdmin');
13    }

```

Gambar 100. *Controller Class Login*

- Baris 1

Pada baris 1 adalah membuka bahasa php pada halaman tersebut dimulai dari baris 1 yang kemudian baris-baris selanjutnya dapat juga menggunakan seluruh elemen dari bahasa php.

- Baris 2

Pada baris 2 berisi pendeklarasian *class* yang bernama *Login* dan hasil *extends* dari *CI_Controller*, yang berarti *class Login* ini mendapatkan sifat-sifat/atribut dari class induk yaitu *CI_Controller*.

- Baris 3

Pada baris 3 berisi pembuatan *variable* atau bisa juga disebut pendeklarasian dengan nama “\$filename” yang isinya adalah “import_data” dengan tipe data *string* kemudian *variable* ini bersifat *private* yang berarti hanya *controller* dengan *class Login* saja yang dapat menggunakannya.

- Baris 4

Pada Baris 4 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 5

Pada baris 5 ini berisi *function construct* yang berarti setiap ada *link* yang mengarah kepada *controller class Login* ini yang dijalankan pertama kali adalah beberapa fungsi tertentu yang sudah disiapkan sesuai kebutuhan *controller class Login* yang dipanggil didalam *function construct*.

- Baris 6

Pada baris 6 ini sebenarnya hampir sama seperti pada baris 5 namun perbedaannya adalah yaitu setiap ada *link* yang mengarah kepada *controller class Login* ini *construct parent* akan menjalankan seluruh *function construct* yang terdapat pada *CI_Controller* karena *class Login* ini adalah hasil *extends* dari *class CI_Controller*.

- Baris 7

Pada Baris 7 ini berisi pemanggilan salah satu sumber pustaka dengan nama *session* didalam *framework codeigniter* yang memiliki fungsi sebagai wadah penyimpanan data sementara sampai fungsi *session destroy* dipanggil nantinya.

- Baris 8

Pada Baris 8 ini berisi pemanggilan salah satu sumber pustaka dengan nama *pagination* didalam *framework codeigniter* yang memiliki fungsi sebagai pemberi batasan untuk menampilkan data sesuai dengan yang nanti akan diatur.

- Baris 9

- Baris 10

- Baris 11

Pada Baris 11 ini berisi pemanggilan model bernama *ModelLogin* yang sebelumnya sudah dibuat terlebih dahulu dan kemudian model *ModelLogin* ini dapat dipakai oleh setiap fungsi di *controller class Login*.

- Baris 12

- Pada Baris 12 ini berisi pemanggilan model bernama *ModelAdmin* yang sebelumnya sudah dibuat terlebih

dahulu dan kemudian model *ModelAdmin* ini dapat dipakai oleh setiap fungsi di *controller class Login*.

- Baris 13

Pada baris 13 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh fungsi yang akan dijalankan pada *function construct*.

b. *Function Index*

```
15     function index(){  
16         $this->load->view('login');  
17     }
```

Gambar 101. *Function Index*

- Baris 15

Pada Baris 15 ini berisi pendeklarasian fungsi yang bernama *index* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 16

Pada baris 16 ini adalah isi dari *function index* yang berisi pemanggilan *view* bernama *login*.

- Baris 17

Pada baris 17 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function index*.

c. *Function cekLogin*

```
19 function cekLogin(){
20     $data = array('username' => $this->input->post('username'),
21                   'password' => $this->input->post('password')
22                   );
23     $hasil = $this->ModelLogin->cekLogin($data);
24     if ($hasil->num_rows() == 1) {
25         foreach ($hasil->result() as $sess) {
26             $sess_data['id'] = $sess->id;
27             $sess_data['username'] = $sess->username;
28             $sess_data['level'] = $sess->level;
29             $sess_data['status'] = 'LOGIN';
30             $this->session->set_userdata($sess_data);
31         }
32         if ($this->session->userdata('level')=='GAJI') {
33             redirect('admin');
34         }
35         elseif ($this->session->userdata('level')=='TUNJANGAN') {
36             redirect('AdminTunjangan');
37         }
38     }else{
39         echo "<script>alert('Gagal login: Cek username,
40             password!');history.go(-1);</script>";
41     }
```

Gambar 102. *Function cekLogin*

- Baris 19

Pada Baris 19 ini berisi pendeklarasian fungsi yang bernama *cekLogin* yang dapat dipanggil setiap *view* tertentu apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 20

Pada baris 20 ini berisi pendeklarasian *variable array* dengan nama *\$data* yang memiliki isi hasil dari elemen html pada *view* yang sudah diberi nama *username*.

- Baris 21

Pada baris 21 ini adalah lanjutan dari baris 20 yang memiliki hasil dari elemen html pada *view* yang sudah diberi nama *password*.

- Baris 22

Pada baris 22 ini berisi penutup menggunakan kurung tutup dari *variable* *\$data*.

- Baris 23

Pada baris 23 ini berisi pendeklarasian *variable* dengan nama *\$hasil* yang memiliki isi pemanggilan fungsi *cekLogin* pada model *ModelLogin* dengan membawa parameter *\$data* yang sudah dibuat pada baris 21.

- Baris 24

Pada baris 24 ini berisi jika *variable* *\$hasil* pada baris 23 memiliki nomor baris sama dengan 1 atau bisa diartikan data yang dilempar pada *variable* *\$data* baris 21 itu ada yang sama seperti pada tabel database, maka yang terjadi adalah

- Baris 25

Pada baris 25 ini adalah pemanggilan seluruh data yang didapat dari *variable* *\$hasil* yang di inisialkan menjadi *\$sess*.

- Baris 26

Pada baris 26 ini adalah pembuatan *variable* dengan nama `$sess_data['id']` yang berisi `$sess->id` dimana ini adalah id yang didapat dari hasil pemanggilan *variable* `$hasil` pada baris 25.

- Baris 27

Pada baris 27 ini adalah pembuatan *variable* dengan `$sess_data['username']` yang berisi `$sess->username` dimana ini adalah *username* yang didapat dari hasil pemanggilan *variable* `$hasil` pada baris 25.

- Baris 28

Pada baris 28 ini adalah pembuatan *variable* dengan `$sess_data['level']` yang berisi `$sess->level` dimana ini adalah *level* yang didapat dari hasil pemanggilan *variable* `$hasil` pada baris 25.

- Baris 29

Pada baris 29 ini adalah pembuatan *variable* dengan `$sess_data['status']` yang berisi LOGIN dimana isinya dibuat secara bebas sesuai kebutuhan.

- Baris 30

Pada baris 30 ini semua data yang sudah dimasukan ke dalam *variable* masing-masing mulai dari baris 26 sampai baris 29 akan dimasukan kedalam *session* sebagai wadah penyimpanan sementara data tersebut.

- Baris 31

Pada baris 31 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *foreach* yang ada pada baris 24 sebelumnya.

- Baris 32

Pada baris 32 ini berisi jika *level* didalam *session* dari *user* yang sudah melakukan *login* adalah GAJI, maka yang terjadi adalah

- Baris 33

Pada baris 33 ini berisi tujuan fungsi dari *controller admin*, fungsi yang akan diambil adalah fungsi *index* karena tidak ada keterangan fungsi yang lain maka yang diambil adalah fungsi *index*.

- Baris 34

Pada baris 34 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *if* yang ada pada baris 32 sebelumnya.

- Baris 35

Pada baris 35 ini berisi selain itu jika *level* didalam *session* dari *user* yang sudah melakukan *login* adalah TUNJANGAN, maka yang terjadi adalah.

- Baris 36

Pada baris 36 ini berisi tujuan fungsi dari *controller AdminTunjangan*, fungsi yang akan diambil adalah fungsi *index* karena tidak ada keterangan fungsi yang lain maka yang diambil adalah fungsi *index*.

- Baris 37

Pada baris 37 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *elseif* yang ada pada baris 35 sebelumnya.

- Baris 38

Pada baris 38 ini berisi apabila kondisi pada baris 24 tidak terpenuhi dengan benar, maka yang terjadi adalah.

- Baris 39

Pada baris 39 ini berisi menampilkan *popup alert javascript* pada halaman *website* yang berisi “Gagal Login, Cek Username, password” kemudian halaman akan kembali lagi ke halaman *login*.

- Baris 40

Pada baris 40 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *else* yang ada pada baris 38 sebelumnya.

- Baris 41

Pada baris 41 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi fungsi *cekLogin* yang ada pada baris 19 sebelumnya.

d. *Function Logout*

```
44     function logout(){
45         $this->session->sess_destroy();
46         redirect('Login');
47     }
48
49 }
```

Gambar 103. *Function Logout*

- Baris 44

Pada Baris 44 ini berisi pendeklarasian fungsi yang bernama *logout* yang dapat dipanggil oleh *view* apabila *view* tersebut akan melakukan *logout* akun dan mengarahkannya pada fungsi ini.

- Baris 45

Pada baris 45 ini adalah menghapus seluruh data yang terdapat pada *session* atau penyimpanan sementara yang sebelumnya sudah dideklarasikan pada baris 26 sampai baris 30.

- Baris 46

Pada baris 46 ini berisi tujuan fungsi dari *controller Login*, fungsi yang akan diambil adalah fungsi *index* karena tidak ada keterangan fungsi yang lain maka yang diambil adalah fungsi *index*.

- Baris 47

Pada baris 47 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function logout*.

- Baris 48

Pada baris 48 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *controller class Login* berisi fungsi-fungsi yang akan dijalankan.

2. *Controller Admin*

a. *Controller Class Admin*

```
1 <?php
2
3 class Admin extends CI_Controller {
4     private $filename = "import_data";
5
6     public function __construct(){
7         parent::__construct();
8         $this->load->library('session');
9         $this->load->model('ModelAdmin');
10
11         if($this->session->userdata('status') != "LOGIN"
12            and $this->session->userdata('status') != "GAJI"){
13             redirect('Login');
14         }
15     }
```

Gambar 104. *Controller Class Admin*

- Baris 1

Pada baris 1 adalah membuka bahasa php pada halaman tersebut dimulai dari baris 1 yang kemudian baris-baris selanjutnya dapat juga menggunakan seluruh elemen dari bahasa php.

- Baris 2

Pada Baris 2 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 3

Pada baris 3 berisi pendeklarasian *class* yang bernama *Admin* dan hasil *extends* dari *CI_Controller*, yang berarti *class Admin* ini mendapatkan sifat-sifat/atribut dari class induk yaitu *CI_Controller*.

- Baris 4

Pada baris 4 berisi pembuatan *variable* atau bisa juga disebut pendeklarasian dengan nama “\$filename” yang isinya adalah “import_data” dengan tipe data *string* kemudian *variable* ini bersifat *private* yang berarti hanya *controller* dengan *class Admin* saja yang dapat menggunakannya.

- Baris 5

Pada Baris 5 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 6

Pada baris 5 ini berisi *function construct* yang berarti setiap ada *link* yang mengarah kepada *controller class Admin* ini yang dijalankan pertama kali adalah beberapa fungsi tertentu yang sudah disiapkan sesuai kebutuhan *controller class Admin* yang dipanggil didalam *function construct*.

- Baris 7

Pada baris 7 ini sebenarnya hampir sama seperti pada baris 6 namun perbedaannya adalah yaitu setiap ada *link* yang mengarah kepada *controller class Admin* ini *construct parent* akan menjalankan seluruh *function construct* yang terdapat pada *CI_Controller* karena *class Admin* ini adalah hasil *extends* dari *class CI_Controller*.

- Baris 8

Pada Baris 8 ini berisi pemanggilan salah satu sumber pustaka dengan nama *session* didalam *framework codeigniter* yang memiliki fungsi sebagai wadah penyimpanan data sementara sampai fungsi *session destroy* dipanggil nantinya.

- Baris 9

Pada Baris 9 ini berisi pemanggilan model bernama *ModelAdmin* yang sebelumnya sudah dibuat terlebih

dahulu dan kemudian model *ModelAdmin* ini dapat dipakai oleh setiap fungsi di *controller class Admin*.

- Baris 10
Pada Baris 10 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.
- Baris 11
Pada baris 11 ini berisi jika *status* didalam *session* dari *user* yang sudah melakukan *login* adalah bukan LOGIN.
- Baris 12
Pada baris 12 ini berisi lanjutan dari baris 11 yaitu dan jika *level* didalam *session* dari *user* yang sudah melakukan *login* adalah bukan GAJI.
- Baris 13
Pada baris 13 ini berisi tujuan fungsi dari *controller Login*, fungsi yang akan diambil adalah fungsi *index* karena tidak ada keterangan fungsi yang lain maka yang diambil adalah fungsi *index*.
- Baris 14
Pada baris 14 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *if* yang akan dijalankan pada baris 11 sebelumnya.

- Baris 15

Pada baris 15 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh fungsi yang akan dijalankan pada *function construct*.

b. *Function Index*

```
17     public function index(){
18         $data['pegawai'] = $this->ModelAdmin->view();
19         $this->load->view('komponen/header');
20         $this->load->view('komponen/sidebar');
21         $this->load->view('Slip/TampilPegawai', $data);
22         $this->load->view('komponen/footer');
23     }
```

Gambar 105. *Function Index*

- Baris 17

Pada Baris 17 ini berisi pendeklarasian fungsi yang bernama *index* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 18

Pada Baris 18 ini berisi pendeklarasian *variable* *\$data* *array* dengan nama ‘pegawai’ yang memiliki isi pemanggilan fungsi *view* pada model *ModelAdmin*.

- Baris 19

Pada baris 19 ini adalah isi dari *function index* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *header*.

- Baris 20

Pada baris 20 ini adalah isi dari *function index* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *sidebar*.

- Baris 21

Pada baris 21 ini adalah isi dari *function index* yang berisi pemanggilan *view* yang berada pada folder *slip* dan bernama *tampilPegawai* dengan membawa *variable \$data array*.

- Baris 22

Pada baris 22 ini adalah isi dari *function index* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *footer*.

- Baris 23

Pada baris 23 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function index*

c. *Function getDataBulan*

```
25 public function getDataBulan($bulan_tahun){
26     $data['total_potongan'] = $this->ModelAdmin->sumPotongan($
        bulan_tahun);
27     $data['total_gaji_bersih'] = $this->ModelAdmin->
        sumGajiDiterima($bulan_tahun);
28     $data['potongan'] = $this->ModelAdmin->getPotonganBulan($
        bulan_tahun);
29     $data['bulan_tahun'] = $bulan_tahun;
30     $this->load->view('komponen/header');
31     $this->load->view('komponen/sidebar');
32     $this->load->view('Slip/TampilPotonganBulan', $data);
33     $this->load->view('komponen/footer');
34 }
```

Gambar 106. *Function getDataBulan*

- Baris 25

Pada Baris 25 ini berisi pendeklarasian fungsi yang bernama *getDataBulan* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini sambil melempar parameter, lalu fungsi ini akan menangkap parameter tersebut dan diberi penamaan *\$bulan_tahun*.

- Baris 26

Pada Baris 26 ini berisi pendeklarasian *variable* *\$data* *array* dengan nama 'total_potongan' yang memiliki isi pemanggilan fungsi *sumPotongan* pada model *ModelAdmin* dengan membawa parameter *\$bulan_tahun*.

- Baris 27

Pada Baris 27 ini berisi pendeklarasian *variable* *\$data* *array* dengan nama ‘total_gaji_bersih’ yang memiliki isi pemanggilan fungsi *sumGajiDiterima* pada model *ModelAdmin* dengan membawa parameter *\$bulan_tahun*.

- Baris 28

Pada Baris 28 ini berisi pendeklarasian *variable* *\$data* *array* dengan nama ‘potongan’ yang memiliki isi pemanggilan fungsi *getPotonganBulan* pada model *ModelAdmin* dengan membawa parameter *\$bulan_tahun*.

- Baris 29

Pada Baris 29 ini berisi pendeklarasian *variable* *\$data* *array* dengan nama ‘bulan_tahun’ yang memiliki isi dari parameter yang didapat oleh fungsi *getDataBulan*.

- Baris 30

Pada baris 30 ini adalah isi dari *function index* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *header*.

- Baris 31

Pada baris 31 ini adalah isi dari *function index* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *sidebar*.

- Baris 32

Pada baris 32 ini adalah isi dari *function index* yang berisi pemanggilan *view* yang berada pada folder slip dan bernama *tampilPotonganBulan* dengan membawa *variable \$data array*.

- Baris 33

Pada baris 33 ini adalah isi dari *function index* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *footer*.

- Baris 34

Pada baris 34 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function index*

d. *Function editPegawai*

```
36 public function editPegawai(){
37     $nip = $this->input->post('nip');
38     $data = array(
39         'nip' => $nip,
40         'nama_pegawai' => $this->input->post('nama_pegawai'),
41         'golongan' => $this->input->post('golongan'),
42         'rekening' => $this->input->post('rekening'),
43         'nama_rekening' => $this->input->post('nama_rekening'),
44         'nama_bank' => $this->input->post('nama_bank'),
45         'email' => $this->input->post('email'),
46         'gaji_bersih' => $this->input->post('gaji_bersih'),
47         'status' => $this->input->post('status'),
48     );
49     $this->ModelAdmin->editPegawai($data,$nip);
50     redirect('admin');
51 }
```

Gambar 107. *Function editPegawai*

- Baris 36

Pada Baris 36 ini berisi pendeklarasian fungsi yang bernama *getDataBulan* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 37

Pada baris 37 ini berisi pendeklarasian *variable* dengan nama *\$nip* yang memiliki isi hasil dari elemen *html* pada *view* yang sudah diberi nama *nip*.

- Baris 38

Pada baris 38 ini berisi pendeklarasian *variable array* dengan nama *\$data*

- Baris 39

Pada baris 39 ini ditetapkan untuk nama 'nip' akan di isi dengan \$nip yaitu hasil dari elemen html pada *view* yang sudah diberi nama nip.

- Baris 40

Pada baris 40 ini ditetapkan untuk nama 'nama_pegawai' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama nama_pegawai.

- Baris 41

Pada baris 41 ini ditetapkan untuk nama 'golongan' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama golongan.

- Baris 42

Pada baris 42 ini ditetapkan untuk nama 'rekening' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama rekening.

- Baris 43

Pada baris 43 ini ditetapkan untuk nama 'nama_rekening' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama nama_rekening.

- Baris 44

Pada baris 44 ini ditetapkan untuk nama 'nama_bank' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama nama_bank.

- Baris 45

Pada baris 45 ini ditetapkan untuk nama 'email' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama email

- Baris 46

Pada baris 46 ini ditetapkan untuk nama 'gaji_bersih' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama gaji_bersih.

- Baris 47

Pada baris 47 ini ditetapkan untuk nama 'status' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama status.

- Baris 48

Pada baris 48 ini berisi penutup menggunakan kurung tutup dari seluruh isi *variable array* pada baris 38.

- Baris 49

Pada Baris 49 ini berisi pemanggilan fungsi *editPegawai* pada model *ModelAdmin* dengan membawa parameter \$data dan \$nip.

- Baris 50

Pada baris 50 ini berisi tujuan fungsi dari *controller Admin*, fungsi yang akan diambil adalah fungsi *editPegawai* karena tidak ada keterangan fungsi yang lain maka yang diambil adalah fungsi *index*.

- Baris 51

Pada baris 51 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function editPegawai*.

e. *Function Potongan*

```
53 public function potongan(){
54     $data['potongan'] = $this->ModelAdmin->getPotongan();
55     $this->load->view('komponen/header');
56     $this->load->view('komponen/sidebar');
57     $this->load->view('Slip/TampilPotongan', $data);
58     $this->load->view('komponen/footer');
59 }
```

Gambar 108. *Function Potongan*

- Baris 53

Pada Baris 53 ini berisi pendeklarasian fungsi yang bernama *Potongan* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 54

Pada Baris 54 ini berisi pendeklarasian *variable* *\$data array* dengan nama ‘potongan’ yang memiliki isi pemanggilan fungsi *getPotongan* pada model *ModelAdmin*.

- Baris 55

Pada baris 55 ini adalah isi dari *function potongan* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *header*.

- Baris 56

Pada baris 56 ini adalah isi dari *function potongan* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *sidebar*.

- Baris 57

Pada baris 57 ini adalah isi dari *function potongan* yang berisi pemanggilan *view* yang berada pada folder *slip* dan bernama *tampilPotongan* dengan membawa *variable \$data array*.

- Baris 58

Pada baris 58 ini adalah isi dari *function potongan* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *footer*.

- Baris 59

Pada baris 59 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function potongan*.

f. *Function getDetail Pegawai*

```
61     public function getDetailPegawai($nip){
62         $data['pegawai'] = $this->ModelAdmin->getDetailPegawai($nip);
63         $data['pegawai2'] = $this->ModelAdmin->getNamaPegawai($nip);
64         $data['pinjaman'] = $this->ModelAdmin->getPinjaman();
65
66         $this->load->view('komponen/header');
67         $this->load->view('komponen/sidebar');
68         $this->load->view('Slip/TampilanDetailPegawai', $data);
69         $this->load->view('komponen/footer');
70     }
```

Gambar 109. *Function getDetailPegawai*

- Baris 61

Pada Baris 61 ini berisi pendeklarasian fungsi yang bernama *getDetailPegawai* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini sambil melempar parameter, lalu fungsi ini akan menangkap parameter tersebut dan diberi penamaan *\$nip*.

- Baris 62

Pada Baris 62 ini berisi pendeklarasian *variable* *\$data* *array* dengan nama 'pegawai' yang memiliki isi pemanggilan fungsi *getDetailPegawai* pada model *ModelAdmin* dengan membawa parameter *\$nip*.

- Baris 63

Pada Baris 63 ini berisi pendeklarasian *variable* `$data` *array* dengan nama ‘pegawai2’ yang memiliki isi pemanggilan fungsi *getNamaPegawai* pada model *ModelAdmin* dengan membawa parameter `$nip`.

- Baris 64

Pada Baris 64 ini berisi pendeklarasian *variable* `$data` *array* dengan nama ‘pinjaman’ yang memiliki isi pemanggilan fungsi *getPinjaman* pada model *ModelAdmin*.

- Baris 65

Pada Baris 65 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 66

Pada baris 66 ini adalah isi dari *function* *getDetailPegawai* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *header*.

- Baris 67

Pada baris 67 ini adalah isi dari *function* *getDetailPegawai* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *sidebar*.

- Baris 68

Pada baris 68 ini adalah isi dari *function* *getDetailPegawai* yang berisi pemanggilan *view* yang berada pada folder *slip* dan bernama *tampilanDetailPegawai* dengan membawa *variable* *\$data array*.

- Baris 69

Pada baris 69 ini adalah isi dari *function* *getDetailPegawai* yang berisi pemanggilan *view* yang berada pada folder *komponen* dan bernama *footer*.

- Baris 70

Pada baris 70 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function* *getDetailPegawai*.

g. *Function* *getDataPinjaman*

```
72 ▾ public function getDataPinjaman(){  
73     $data['pinjaman'] = $this->ModelAdmin->getDataPinjaman();  
74     $this->load->view('komponen/header');  
75     $this->load->view('komponen/sidebar');  
76     $this->load->view('Slip/TampilPinjaman', $data);  
77     $this->load->view('komponen/footer');  
78 }
```

Gambar 110. *Function* *getDataPinjaman*

- Baris 72

Pada Baris 72 ini berisi pendeklarasian fungsi yang bernama *getDataPinjaman* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 73

Pada Baris 73 ini berisi pendeklarasian *variable* *\$data* *array* dengan nama 'pinjaman' yang memiliki isi pemanggilan fungsi *getDataPinjaman* pada model *ModelAdmin*.

- Baris 74

Pada baris 74 ini adalah isi dari *function* *getDataPinjaman* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *header*.

- Baris 75

Pada baris 75 ini adalah isi dari *function* *getDataPinjaman* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *sidebar*.

- Baris 76

Pada baris 76 ini adalah isi dari *function* *getDataPinjaman* yang berisi pemanggilan *view* yang berada pada folder slip dan bernama *tampilPinjaman* dengan membawa *variable* *\$data* *array*.

- Baris 77

Pada baris 77 ini adalah isi dari *function* *getDataPinjaman* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *footer*.

- Baris 78

Pada baris 78 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function* *getDataPinjaman*.

h. *Function* *getDataRekomendasiPinjaman*

```
80 public function getDataRekomendasiPinjaman(){
81     $hasil = $this->input->post('hasil');
82     $nip = $this->input->post('nip');
83     $nama_pegawai = $this->input->post('nama_pegawai');
84     $bulan_tahun = $this->input->post('bulan_tahun');
85     $id = $this->input->post('id');
86     $data['pinjaman'] = $this->ModelAdmin->getRekomendasiPinjaman($hasil);
87     $data['max_value'] = $this->ModelAdmin->getMaxValue($hasil);
88     $data['min_value1'] = $this->ModelAdmin->getMinValue1($hasil);
89     $data['min_value2'] = $this->ModelAdmin->getMinValue2($hasil);
90     $data['hasil'] = $hasil;
91     $data['yaww'] = $this->input->post('id');
92     $data['nip'] = $nip;
93     $data['nama_pegawai'] = $nama_pegawai;
94     $data['bulan_tahun'] = $bulan_tahun;
95
96     $this->load->view('komponen/header');
97     $this->load->view('komponen/sidebar');
98     $this->load->view('Slip/TampilDataPinjaman', $data);
99     $this->load->view('komponen/footer');
100 }
```

Gambar 111. *Function* *getDataRekomendasiPinjaman*

- Baris 80

Pada Baris 80 ini berisi pendeklarasian fungsi yang bernama *getDataRekomendasiPinjaman* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 81

Pada baris 81 ini berisi pendeklarasian *variable* dengan nama *\$hasil* yang memiliki isi hasil dari elemen html pada *view* yang sudah diberi nama hasil.

- Baris 82

Pada baris 82 ini berisi pendeklarasian *variable* dengan nama *\$nip* yang memiliki isi hasil dari elemen html pada *view* yang sudah diberi nama nip.

- Baris 83

Pada baris 83 ini berisi pendeklarasian *variable* dengan nama *\$nama_pegawai* yang memiliki isi hasil dari elemen html pada *view* yang sudah diberi nama nama_pegawai.

- Baris 84

Pada baris 84 ini berisi pendeklarasian *variable* dengan nama *\$bulan_tahun* yang memiliki isi hasil dari elemen html pada *view* yang sudah diberi nama bulan_tahun.

- Baris 85

Pada baris 85 ini berisi pendeklarasian *variable* dengan nama \$id yang memiliki isi hasil dari elemen html pada *view* yang sudah diberi nama id.

- Baris 86

Pada Baris 86 ini berisi pendeklarasian *variable* \$data *array* dengan nama 'pinjaman' yang memiliki isi pemanggilan fungsi *getRekomendasiPinjaman* pada model *ModelAdmin* dengan membawa parameter \$hasil pada baris 81.

- Baris 87

Pada Baris 87 ini berisi pendeklarasian *variable* \$data *array* dengan nama 'max_value' yang memiliki isi pemanggilan fungsi *getMaxValue* pada model *ModelAdmin* dengan membawa parameter \$hasil pada baris 81.

- Baris 88

Pada Baris 88 ini berisi pendeklarasian *variable* \$data *array* dengan nama 'min_value1' yang memiliki isi pemanggilan fungsi *getMinValue1* pada model *ModelAdmin* dengan membawa parameter \$hasil pada baris 81.

- Baris 89

Pada Baris 89 ini berisi pendeklarasian *variable* \$data array dengan nama 'min_value2' yang memiliki isi pemanggilan fungsi *getMinValue2* pada model *ModelAdmin* dengan membawa parameter \$hasil pada baris 81.

- Baris 90

Pada Baris 90 ini berisi pendeklarasian *variable* \$data array dengan nama 'hasil' yang memiliki isi \$hasil pada baris 81 yaitu hasil dari elemen html pada *view* yang sudah diberi nama hasil.

- Baris 91

Pada Baris 91 ini berisi pendeklarasian *variable* \$data array dengan nama 'yaww' yang memiliki isi hasil dari elemen html pada *view* yang sudah diberi nama id.

- Baris 92

Pada Baris 92 ini berisi pendeklarasian *variable* \$data array dengan nama 'nip' yang memiliki isi \$nip pada baris 82 yaitu hasil dari elemen html pada *view* yang sudah diberi nama nip.

- Baris 93

Pada Baris 93 ini berisi pendeklarasian *variable* \$data array dengan nama 'nama_pegawai' yang memiliki isi \$nama_pegawai pada baris 83 yaitu hasil dari elemen

html pada *view* yang sudah diberi nama `nama_pegawai`.

- Baris 94

Pada Baris 94 ini berisi pendeklarasian *variable* `$data` *array* dengan nama `'bulan_tahun'` yang memiliki isi `$nama_pegawai` pada baris 84 yaitu hasil dari elemen html pada *view* yang sudah diberi nama `bulan_tahun`.

- Baris 95

Pada Baris 95 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 96

Pada baris 96 ini adalah isi dari *function* `getDataRekomendasiPinjaman` yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *header*.

- Baris 97

Pada baris 97 ini adalah isi dari *function* `getDataRekomendasiPinjaman` yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *sidebar*.

- Baris 98

Pada baris 98 ini adalah isi dari *function* `getDataRekomendasiPinjaman` yang berisi

pemanggilan *view* yang berada pada folder *slip* dan bernama *tampilDataPinjaman* dengan membawa *variable \$data array*.

- Baris 99

Pada baris 99 ini adalah isi dari *function getDataRekomendasiPinjaman* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *footer*.

- Baris 100

Pada baris 78 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function getDataRekomendasiPinjaman*.

i. *Function formPegawai*

```
102 public function formPegawai(){
103     $data = array();
104     if(isset($_POST['preview'])){
105         $upload = $this->ModelAdmin->upload_file($this->filename);
106
107         if($upload['result'] == "success"){
108             include APPPATH.'third_party/PHPExcel/PHPExcel.php';
109
110             $excelreader = new PHPExcel_Reader_Excel2007();
111             $loadexcel = $excelreader->load('excel/'.$this->filename.'.xlsx');
112             $sheet = $loadexcel->getActiveSheet()->toArray(null, true, true, true);
113
114             $data['sheet'] = $sheet;
115         }else{
116             $data['upload_error'] = $upload['error'];
117         }
118     }
119
120     $this->load->view('komponen/header');
121     $this->load->view('komponen/sidebar');
122     $this->load->view('Slip/ImportPegawai', $data);
123     $this->load->view('komponen/footer');
124 }
```

Gambar 112. *Function formPegawai*

- Baris 102

Pada Baris 102 ini berisi pendeklarasian fungsi yang bernama *formPegawai* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 103

Pada baris 103 ini berisi pendeklarasian *variable* dengan nama *\$data* sebagai *array* yang akan digunakan pada baris selanjutnya.

- Baris 104

Pada baris 104 ini berisi jika user menekan tombol pada *view* yang sudah diberi nama *preview*, maka yang terjadi adalah.

- Baris 105

Pada baris 105 ini berisi pendeklarasian *variable* dengan nama *\$upload* yang berisi melakukan upload file dengan memanggil fungsi *upload_file* pada model *ModelAdmin* dengan membawa parameter *\$this->filename* yaitu nama *file* yang akan di *upload*.

- Baris 106

Pada Baris 106 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 107

Pada Baris 107 berisi jika proses *\$upload* atau pada baris 105 memiliki hasil sama dengan “success”, maka yang terjadi selanjutnya adalah.

- Baris 108

Pada Baris 108 berisi pemanggilan *plugin* *PHPExcel* yang sebelumnya sudah dimasukan kedalam *folder* *'third_party/PHPExcel/PHPExcel.php'*.

- Baris 109

Pada Baris 109 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 110

Pada Baris 110 berisi pendeklarasian *variable* dengan nama \$excelreader yang berisi pembuatan fungsi baru yang isinya didapat dari fungsi *PHPExcel_Reader_Excel2007* yang didapat dari *plugin PHPExcel*.

- Baris 111

Pada Baris 111 berisi pendeklarasian *variable* dengan nama \$loadexcel yang berisi *variable* \$excelreader pada baris 110 akan melakukan pemanggilan *file* yang sebelumnya di *upload* kedalam *folder* 'excel/nama_file.xlsx'

- Baris 112

Pada Baris 112 berisi pendeklarasian *variable* dengan nama \$sheet yang berisi *variable* \$sheet tersebut akan dimasukan ke dalam *array* data yang nantinya akan di kirim ke *view*.

- Baris 113

Pada Baris 113 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 114

Pada Baris 104 ini berisi pendeklarasian *variable* `$data array` dengan nama 'sheet' yang memiliki isi `$sheet` pada baris 112 yaitu *variable* `$sheet` tersebut berisi data-data yang sudah diinput di dalam *excel* yang sudah di *upload* sebelumnya.

- Baris 115

Pada baris 115 ini berisi apabila kondisi pada baris 107 tidak terpenuhi dengan benar, maka yang terjadi adalah.

- Baris 116

Pada Baris 116 ini berisi pendeklarasian *variable* `$data array` dengan nama 'upload_error' yang memiliki isi `$upload['error']` yaitu mengambil pesan *error upload* pada baris 105 untuk dikirim ke *view* dan ditampilkan nantinya.

- Baris 117

Pada baris 117 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *else* pada baris 115 yang akan dijalankan.

- Baris 118

Pada baris 118 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *if* yang dimulai pada baris 104 dan akan dijalankan.

- Baris 119

Pada Baris 119 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 120

Pada baris 120 ini adalah isi dari *function formPegawai* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *header*.

- Baris 121

Pada baris 121 ini adalah isi dari *function formPegawai* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *sidebar*.

- Baris 122

Pada baris 122 ini adalah isi dari *function formPegawai* yang berisi pemanggilan *view* yang berada pada folder slip dan bernama *ImportPegawai* dengan membawa *variable \$data array*.

- Baris 123

Pada baris 123 ini adalah isi dari *function formPegawai* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *footer*.

- Baris 124

Pada baris 124 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function formPegawai*.

j. *Function inputPinjaman*

```
126 public function inputPinjaman(){
127     $data = array(
128         'id_potongan' => $this->input->post('id_potongan'),
129         'nip' => $this->input->post('nip'),
130         'bulan_tahun' => $this->input->post('bulan_tahun'),
131         'plafond' => $this->input->post('plafond'),
132         'bulan' => $this->input->post('bulan'),
133         'angsuran' => $this->input->post('angsuran'),
134         'total' => $this->input->post('total'),
135     );
136
137     $this->ModelAdmin->inputPinjaman($data);
138     redirect('admin/getDataPinjaman');
139 }
```

Gambar 113. *Function inputPinjaman*

- Baris 126

Pada Baris 126 ini berisi pendeklarasian fungsi yang bernama *inputPinjaman* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 127

Pada baris 127 ini berisi pendeklarasian *variable array* dengan nama \$data

- Baris 128

Pada baris 128 ini ditetapkan untuk nama 'id_potongan' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama id_potongan.

- Baris 129

Pada baris 129 ini ditetapkan untuk nama 'nip' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama nip.

- Baris 130

Pada baris 130 ini ditetapkan untuk nama 'bulan_tahun' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama bulan_tahun.

- Baris 131

Pada baris 131 ini ditetapkan untuk nama 'plafond' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama plafond.

- Baris 132

Pada baris 132 ini ditetapkan untuk nama 'bulan' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama bulan.

- Baris 133
Pada baris 133 ini ditetapkan untuk nama 'angsuran' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama angsuran.
- Baris 134
Pada baris 134 ini ditetapkan untuk nama 'total' akan di isi dengan hasil dari elemen html pada *view* yang sudah diberi nama total.
- Baris 135
Pada baris 135 ini berisi penutup menggunakan kurung tutup dari seluruh isi *variable array* pada baris 127.
- Baris 136
Pada Baris 136 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.
- Baris 137
Pada Baris 137 ini berisi pemanggilan fungsi *inputPinjaman* pada model *ModelAdmin* dengan membawa parameter \$data.
- Baris 138
Pada baris 138 ini berisi tujuan fungsi dari *controller Admin*, fungsi yang akan diambil adalah fungsi

getDataPinjaman karena setelah nama *controller* kemudian nama fungsinya.

- Baris 139

Pada baris 139 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function inputPinjaman*.

k. *Function formPotongan*

```
141 public function formPotongan(){
142     $data = array();
143     if(isset($_POST['preview'])){
144         $upload = $this->ModelAdmin->upload_file_gaji($this->filename);
145
146         if($upload['result'] == "success"){
147             include APPPATH.'third_party/PHPExcel/PHPExcel.php';
148
149             $excelreader = new PHPExcel_Reader_Excel2007();
150             $loadexcel = $excelreader->load('excel/'.$this->filename.'.xlsx');
151             $sheet = $loadexcel->getActiveSheet()->toArray(null, true, true ,
152                 true);
153
154             $data['sheet'] = $sheet;
155         }else{
156             $data['upload_error'] = $upload['error'];
157         }
158     }
159 }
```

Gambar 114. *Function formPotongan*

- Baris 141

Pada Baris 141 ini berisi pendeklarasian fungsi yang bernama *formPotongan* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 142

Pada baris 142 ini berisi pendeklarasian *variable* dengan nama *\$data* sebagai *array* yang akan digunakan pada baris selanjutnya.

- Baris 143

Pada baris 143 ini berisi jika user menekan tombol pada *view* yang sudah diberi nama *preview*, maka yang terjadi adalah.

- Baris 144

Pada baris 144 ini berisi pendeklarasian *variable* dengan nama *\$upload* yang berisi melakukan upload file dengan memanggil fungsi *upload_file* pada model *ModelAdmin* dengan membawa parameter *\$this->filename* yaitu nama *file* yang akan di *upload*.

- Baris 145

Pada Baris 145 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 146

Pada Baris 146 berisi jika proses \$upload atau pada baris 143 memiliki hasil sama dengan “success”, maka yang terjadi selanjutnya adalah.

- Baris 147

Pada Baris 147 berisi pemanggilan *plugin* PHPExcel yang sebelumnya sudah dimasukan kedalam *folder* ‘third_party/PHPExcel/PHPExcel.php’.

- Baris 148

Pada Baris 148 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 149

Pada Baris 149 berisi pendeklarasian *variable* dengan nama \$excelreader yang berisi pembuatan fungsi baru yang isinya didapat dari fungsi *PHPExcel_Reader_Excel2007* yang didapat dari *plugin PHPExcel*.

- Baris 150

Pada Baris 150 berisi pendeklarasian *variable* dengan nama \$loadexcel yang berisi *variable* \$excelreader pada baris 149 akan melakukan pemanggilan *file* yang sebelumnya di *upload* kedalam *folder* ‘excel/nama_file.xlsx’

- Baris 151

Pada Baris 151 berisi pendeklarasian *variable* dengan nama `$sheet` yang berisi *variable* `$sheet` tersebut akan dimasukan ke dalam *array* data yang nantinya akan di kirim ke *view*.

- Baris 152

Pada Baris 152 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 153

Pada Baris 153 ini berisi pendeklarasian *variable* `$data array` dengan nama `'sheet'` yang memiliki isi `$sheet` pada baris 151 yaitu *variable* `$sheet` tersebut berisi data-data yang sudah diinput di dalam *excel* yang sudah di *upload* sebelumnya.

- Baris 154

Pada baris 154 ini berisi apabila kondisi pada baris 146 tidak terpenuhi dengan benar, maka yang terjadi adalah.

- Baris 155

Pada Baris 155 ini berisi pendeklarasian *variable* `$data array` dengan nama `'upload_error'` yang memiliki isi `$upload['error']` yaitu mengambil pesan *error upload* pada baris 144 untuk dikirim ke *view* dan ditampilkan nantinya.

- Baris 156
Pada baris 156 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *else* pada baris 153 yang akan dijalankan.
- Baris 157
Pada baris 157 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *if* yang dimulai pada baris 143 dan akan dijalankan.
- Baris 158
Pada Baris 158 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.
- Baris 159
Pada baris 159 ini adalah isi dari *function formPotongan* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *header*.
- Baris 160
Pada baris 160 ini adalah isi dari *function formPotongan* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *sidebar*.
- Baris 161
Pada baris 161 ini adalah isi dari *function formPotongan* yang berisi pemanggilan *view* yang

berada pada folder slip dan bernama ImportPotongan dengan membawa *variable* \$data array.

- Baris 162

Pada baris 162 ini adalah isi dari *function formPotongan* yang berisi pemanggilan *view* yang berada pada folder komponen dan bernama *footer*.

- Baris 163

Pada baris 163 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function formPotongan*.

1. Function Import (1)

```
165     public function import(){
166         include APPPATH.'third_party/PHPExcel/PHPExcel.php';
167
168         $excelreader = new PHPExcel_Reader_Excel2007();
169         $loadexcel = $excelreader->load('excel/'.$this->filename.'.xlsx');
170         $sheet = $loadexcel->getActiveSheet()->toArray(null, true, true, true);
171
172         $data = array();
173         $numrow = 1;
```

Gambar 115. Function Import (1)

- Baris 165

Pada Baris 165 ini berisi pendeklarasian fungsi yang bernama *import* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 166

Pada Baris 166 berisi pemanggilan *plugin* PHPExcel yang sebelumnya sudah dimasukan kedalam *folder* 'third_party/PHPExcel/PHPExcel.php'.

- Baris 167

Pada Baris 167 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 168

Pada Baris 168 berisi pendeklarasian *variable* dengan nama \$excelreader yang berisi pembuatan fungsi baru yang isinya didapat dari fungsi *PHPExcel_Reader_Excel2007* yang didapat dari *plugin PHPExcel*.

- Baris 169

Pada Baris 169 berisi pendeklarasian *variable* dengan nama \$loadexcel yang berisi *variable* \$excelreader pada baris 168 akan melakukan pemanggilan *file* yang sebelumnya di *upload* kedalam *folder* 'excel/nama_file.xlsx'

- Baris 170

Pada Baris 170 berisi pendeklarasian *variable* dengan nama \$sheet yang berisi *variable* \$sheet tersebut akan dimasukan ke dalam *array* data yang nantinya akan di kirim ke *view*.

- Baris 171

Pada Baris 171 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 172

Pada baris 172 ini berisi pendeklarasian *variable* dengan nama \$data sebagai *array* yang akan digunakan pada baris selanjutnya.

- Baris 173

Pada baris 173 ini berisi pendeklarasian *variable* dengan nama \$numrow dengan tipe data *number* yang berisi '1' kemudian akan digunakan pada baris selanjutnya.

m. *Function Import (2)*

```
174         foreach($sheet as $row){
175             if($numrow > 1){
176
177                 array_push($data, array(
178                     'id'=>$row['A'],
179                     'nip'=>$row['C'],
180                     'nama_pegawai'=>$row['D'],
181                     'golongan'=>$row['E'],
182                     'rekening'=>$row['F'],
183                     'nama_rekening'=>$row['G'],
184                     'nama_bank'=>$row['H'],
185                     'gaji_bersih'=>$row['I'], |
186                     'email'=>$row['AB'],
187                     'status'=>'AKTIF',
188                 ));
189             }
190             $numrow++;
191         }
192
193         $this->ModelAdmin->insert_multiple($data);
194         redirect("admin");
195     }
196 }
```

Gambar 116. *Function Import (2)*

- Baris 174

Pada Baris 174 ini berisi untuk setiap data yang didapat pada *variable* \$sheet yang sebelumnya sudah

dibuat di baris 170 maka akan dialiaskan semuanya menjadi \$row.

- Baris 175

Pada baris 175 ini berisi jika setelah melakukan pengecekan pada *variable* \$numrow dan apakah hasilnya lebih dari 1, artinya karena baris pertama pada *file excel* adalah nama-nama kolom jadi cukup dilewat tidak perlu diimport.

- Baris 176

Pada Baris 176 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 177

Pada Baris 177 berisi 1 melakukan *push (add) array* data kedalam *variable* \$data.

- Baris 178

Pada baris 178 ini ditetapkan untuk nama 'id' akan di isi dengan hasil dari kolom A yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 179

Pada baris 179 ini ditetapkan untuk nama 'nip' akan di isi dengan hasil dari kolom C yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 180

Pada baris 180 ini ditetapkan untuk nama ‘nama_pegawai’ akan di isi dengan hasil dari kolom D yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 181

Pada baris 181 ini ditetapkan untuk nama ‘golongan’ akan di isi dengan hasil dari kolom E yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 182

Pada baris 182 ini ditetapkan untuk nama ‘golongan’ akan di isi dengan hasil dari kolom F yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 183

Pada baris 183 ini ditetapkan untuk nama ‘nama_rekening’ akan di isi dengan hasil dari kolom G yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 184

Pada baris 184 ini ditetapkan untuk nama ‘nama_bank’ akan di isi dengan hasil dari kolom H yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 185

Pada baris 185 ini ditetapkan untuk nama ‘gaji_bersih’ akan di isi dengan hasil dari kolom I yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 186

Pada baris 186 ini ditetapkan untuk nama ‘email’ akan di isi dengan hasil dari kolom AB yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 187

Pada baris 187 ini ditetapkan untuk nama ‘status’ akan di isi dengan ‘AKTIF’ yang memiliki tipe data *string*.

- Baris 188

Pada baris 188 ini berisi penutup menggunakan kurung tutup dari seluruh isi *variable array* yang dimulai pada baris 177.

- Baris 189

Pada Baris 189 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 190

Pada baris 190 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *if* yang dimulai pada baris 175 dan akan dijalankan.

- Baris 191

Pada baris 191 ini *variable* \$numrow akan ditambah 1 setiap kalo data melakukan *looping*.

- Baris 192

Pada baris 192 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *foreach* yang dimulai pada baris 174 dan akan dijalankan.

- Baris 193

Pada Baris 193 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 194

Pada Baris 194 ini berisi pemanggilan fungsi *insert_multiple* pada model *ModelAdmin* dengan membawa parameter \$data.

- Baris 195

Pada baris 195 ini berisi tujuan fungsi dari *controller Admin*, fungsi yang akan diambil adalah fungsi *index* karena tidak ada keterangan fungsi yang lain maka yang diambil adalah fungsi *index*.

- Baris 196

Pada baris 196 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function import*.

n. *Function Import2* (1)

```
198 public function import2(){
199     include APPPATH.'third_party/PHPExcel/PHPExcel.php';
200
201     $excelreader = new PHPExcel_Reader_Excel2007();
202     $loadexcel = $excelreader->load('excel/'.$this->filename.'.xlsx');
203     $sheet = $loadexcel->getActiveSheet()->toArray(null, true, true, true);
204
205     $data = array();
206     $numrow = 1;
```

Gambar 117. *Function Import2* (1)

- Baris 198

Pada Baris 198 ini berisi pendeklarasian fungsi yang bernama *import2* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 199

Pada Baris 199 berisi pemanggilan *plugin* *PHPExcel* yang sebelumnya sudah dimasukan kedalam *folder* 'third_party/PHPExcel/PHPExcel.php'.

- Baris 200

Pada Baris 200 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 201

Pada Baris 201 berisi pendeklarasian *variable* dengan nama \$excelreader yang berisi pembuatan fungsi baru yang isinya didapat dari fungsi *PHPExcel_Reader_Excel2007* yang didapat dari *plugin PHPExcel*.

- Baris 202

Pada Baris 202 berisi pendeklarasian *variable* dengan nama \$loadexcel yang berisi *variable* \$excelreader pada baris 201 akan melakukan pemanggilan *file* yang sebelumnya di *upload* kedalam *folder* 'excel/nama_file.xlsx'

- Baris 203

Pada Baris 203 berisi pendeklarasian *variable* dengan nama \$sheet yang berisi *variable* \$sheet tersebut akan dimasukan ke dalam *array* data yang nantinya akan di kirim ke *view*.

- Baris 204

Pada Baris 204 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 205

Pada baris 205 ini berisi pendeklarasian *variable* dengan nama \$data sebagai *array* yang akan digunakan pada baris selanjutnya.

- Baris 206

Pada baris 206 ini berisi pendeklarasian *variable* dengan nama \$numrow dengan tipe data *number* yang berisi '1' kemudian akan digunakan pada baris selanjutnya.

o. *Function Import2 (2)*

```

207         foreach($sheet as $row){
208             if($numrow > 1){
209
210                 array_push($data, array(
211                     'id_pegawai'=>$row['C'],
212                     'arisan_pengayoman'=>$row['J'],
213                     'idw'=>$row['K'],
214                     'koperasi'=>$row['L'],
215                     'tabungan_dw'=>$row['M'],
216                     'olahraga'=>$row['N'],
217                     'bri'=>$row['O'],
218                     'bjb'=>$row['P'],
219                     'skp'=>$row['Q'],
220                     'lain_lain'=>$row['R'],
221                     'arisan_dw'=>$row['S'],
222                     'arisan_100'=>$row['T'],
223                     'total'=>$row['Y'],
224                     'gaji_diterima'=>$row['Z'],
225                     'bulan_tahun'=>$row['AA']
226                 ));
227
228             }
229             $numrow++;
230         }
231
232         $this->ModelAdmin->insert_multiple_potongan($data);
233         redirect("admin/potongan");
234     }

```

Gambar 118. *Function Import2 (2)*

- Baris 207

Pada Baris 207 ini berisi untuk setiap data yang didapat pada *variable* \$sheet yang sebelumnya sudah dibuat di baris 203 maka akan dialiaskan semuanya menjadi \$row.

- Baris 208

Pada baris 208 ini berisi jika setelah melakukan pengecekan pada *variable* \$numrow dan apakah hasilnya lebih dari 1, artinya karena baris pertama pada *file excel* adalah nama-nama kolom jadi cukup dilewat tidak perlu diimport.

- Baris 209

Pada Baris 209 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 210

Pada Baris 210 berisi 1 melakukan *push (add) array* data kedalam *variable* \$data.

- Baris 211

Pada baris 178 ini ditetapkan untuk nama 'id_pegawai' akan di isi dengan hasil dari kolom C yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 212

Pada baris 212 ini ditetapkan untuk nama ‘arisan_pengayoman’ akan di isi dengan hasil dari kolom J yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 213

Pada baris 213 ini ditetapkan untuk nama ‘idw’ akan di isi dengan hasil dari kolom K yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 214

Pada baris 214 ini ditetapkan untuk nama ‘koperasi’ akan di isi dengan hasil dari kolom L yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 215

Pada baris 215 ini ditetapkan untuk nama ‘tabungan_dw’ akan di isi dengan hasil dari kolom M yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 216

Pada baris 216 ini ditetapkan untuk nama ‘olahraga’ akan di isi dengan hasil dari kolom N yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 217

Pada baris 217 ini ditetapkan untuk nama ‘bri’ akan di isi dengan hasil dari kolom O yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 218

Pada baris 218 ini ditetapkan untuk nama ‘bjb’ akan di isi dengan hasil dari kolom P yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 219

Pada baris 219 ini ditetapkan untuk nama ‘skp’ akan di isi dengan hasil dari kolom Q yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 220

Pada baris 220 ini ditetapkan untuk nama ‘lain_lain’ akan di isi dengan hasil dari kolom R yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 221

Pada baris 221 ini ditetapkan untuk nama ‘arisan_dw’ akan di isi dengan hasil dari kolom S yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 222

Pada baris 222 ini ditetapkan untuk nama ‘arisan_100’ akan di isi dengan hasil dari kolom T yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 223

Pada baris 223 ini ditetapkan untuk nama 'total' akan di isi dengan hasil dari kolom Y yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 224

Pada baris 224 ini ditetapkan untuk nama 'gaji_diterima' akan di isi dengan hasil dari kolom Z yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 225

Pada baris 220 ini ditetapkan untuk nama 'bulan_tahun' akan di isi dengan hasil dari kolom AA yang terdapat pada *file excel* yang sudah di *upload*.

- Baris 226

Pada baris 226 ini berisi penutup menggunakan kurung tutup dari seluruh isi *variable array* yang dimulai pada baris 210.

- Baris 227

Pada Baris 227 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 228

Pada baris 228 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *if* yang dimulai pada baris 208 dan akan dijalankan.

- Baris 229

Pada baris 229 ini *variable* \$numrow akan ditambah 1 setiap kalo data melakukan *looping*.

- Baris 230

Pada baris 230 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi *foreach* yang dimulai pada baris 207 dan akan dijalankan.

- Baris 231

Pada baris 231 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 232

Pada baris 232 ini berisi pemanggilan fungsi *insert_multiple_potongan* pada model *ModelAdmin* dengan membawa parameter \$data.

- Baris 233

Pada baris 233 ini berisi tujuan fungsi dari *controller Admin*, fungsi yang akan diambil adalah fungsi *potongan* karena setelah nama *controller* kemudian nama fungsinya.

- Baris 234

Pada baris 234 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function import2*.

p. *Function deletePinjaman*

```
236 public function deletePinjaman($id){  
237     $this->db->where('id', $id);  
238     $this->db->delete('tb_pinjaman');  
239     redirect('admin/getDataPinjaman');  
240 }
```

Gambar 4.119 *Function deletePinjaman*

- Baris 236

Pada Baris 236 ini berisi pendeklarasian fungsi yang bernama *deletePinjaman* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini sambil melempar parameter, lalu fungsi ini akan menangkap parameter tersebut dan diberi penamaan *\$id*.

- Baris 237

Pada Baris 237 berisi pemfilteran atau penyaringan data yang akan dipilih adalah data dengan nama 'id'

pada tabel akan disamakan dengan parameter \$id yang ditangkap oleh fungsi *deletePinjaman*.

- Baris 238

Pada Baris 238 berisi fungsi menghapus data pada tabel yang bernama 'tb_pinjaman' dengan kriteria yang sudah disebutkan pada baris nomer 237 yaitu data dengan nama 'id' pada tabel akan disamakan dengan parameter \$id yang ditangkap oleh fungsi *deletePinjaman*.

- Baris 239

Pada baris 239 ini berisi tujuan fungsi dari *controller Admin*, fungsi yang akan diambil adalah fungsi *getDataPinjaman* karena setelah nama *controller* kemudian nama fungsinya.

- Baris 240

Pada baris 240 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function deletePinjaman*.

q. *Function deletePegawai*

```
242     public function deletePegawai(){  
243         $this->db->empty_table('tb_pegawai');  
244         $this->ModelAdmin->autoIncrement();  
245         redirect('admin');  
246     }
```

Gambar 4.120 *Function deletePegawai*

- Baris 242

Pada Baris 242 ini berisi pendeklarasian fungsi yang bernama *deletePegawai* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 243

Pada Baris 243 berisi fungsi menghapus seluruh data atau juga bisa disebut melakukan pengosongan tabel pada tabel yang bernama 'tb_pinjaman'.

- Baris 244

Pada Baris 244 ini berisi pemanggilan sebuah fungsi dengan nama *autoIncrement* pada sebuah model yaitu model *ModelAdmin*.

- Baris 245

Pada baris 245 ini berisi tujuan fungsi dari *controller Admin*, fungsi yang akan diambil adalah fungsi *index* karena tidak ada keterangan fungsi yang lain maka yang diambil adalah fungsi *index*.

- Baris 246

Pada baris 246 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function deletePegawai*.

r. *Function deleteAllPotongan*

```
248 public function deleteAllPotongan(){  
249     $this->db->empty_table('tb_potongan');  
250     $this->ModelAdmin->autoIncrementPotongan();  
251     redirect('admin/potongan');  
252 }
```

Gambar 4.121 *Function deleteAllPotongan*

- Baris 248

Pada Baris 248 ini berisi pendeklarasian fungsi yang bernama *deleteAllPotongan* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 249

Pada Baris 249 berisi fungsi menghapus seluruh data atau juga bisa disebut melakukan pengosongan tabel pada tabel yang bernama 'tb_potongan.

- Baris 250

Pada Baris 250 ini berisi pemanggilan sebuah fungsi dengan nama *autoIncrementPotongan* pada sebuah model yaitu model *ModelAdmin*.

- Baris 251

Pada baris 251 ini berisi tujuan fungsi dari *controller Admin*, fungsi yang akan diambil adalah fungsi *potongan* karena setelah nama *controller* kemudian nama fungsinya.

- Baris 251

Pada baris 251 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function deleteAllPotongan*.

s. *Function deletePotongan*

```
254 ▼ public function deletePotongan(){  
255     $bulan_tahun = $this->input->post('bulan_tahun');  
256     $this->ModelAdmin->deletePotongan($bulan_tahun);  
257     redirect('admin/potongan');  
258 }
```

Gambar 4.122 *Function deletePotongan*

- Baris 254

Pada Baris 254 ini berisi pendeklarasian fungsi yang bernama *deletePotongan* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 255

Pada baris 255 ini berisi pendeklarasian *variable* dengan nama *\$bulan_tahun* yang memiliki isi hasil dari elemen html pada *view* yang sudah diberi nama *bulan_tahun*.

- Baris 256

Pada Baris 256 ini berisi pemanggilan sebuah fungsi dengan nama *autoIncrementPotongan* pada sebuah model yaitu model *ModelAdmin* dengan membawa

parameter `$bulan_tahun` yang sebelumnya sudah dideklarasikan pada baris 254.

- Baris 257

Pada baris 257 ini berisi tujuan fungsi dari *controller Admin*, fungsi yang akan diambil adalah fungsi *potongan* karena setelah nama *controller* kemudian nama fungsinya.

- Baris 258

Pada baris 258 ini berisi penutup menggunakan kurung tutup kurawal dari seluruh isi yang akan dijalankan pada *function deleteAllPotongan*.

t. *Function send_email_all* (1)

```
260 ▼ public function send_email_all(){
261     $ArrData = $this->input->post('check_list');
262     $jumlahData = count($ArrData);
263
264 ▼     $config = [
265         'mailtype' => 'html',
266         'charset' => 'utf-8',
267         'protocol' => 'smtp',
268         'smtp_host' => 'ssl://smtp.gmail.com',
269         'smtp_user' => 'pdg.rutanbandung@gmail.com',
270         'smtp_pass' => 'rutan526801',
271         'smtp_port' => 465,
272         'crlf' => "\r\n",
273         'newline' => "\r\n"
274     ];
```

Gambar 4.123 *Function send_email_all* (2)

- Baris 260

Pada Baris 260 ini berisi pendeklarasian fungsi yang bernama *send_email_all* yang dapat dipanggil oleh setiap *view* apabila *view* tersebut mengarahkannya pada fungsi ini.

- Baris 261

Pada baris 261 ini berisi pendeklarasian *variable* dengan nama *\$ArrData* yang memiliki isi hasil dari elemen html pada *view* yang sudah diberi nama *check_list*.

- Baris 262

Pada baris 262 ini berisi pendeklarasian *variable* dengan nama *\$jumlahData* yang memiliki isi menghitung ada berapa banyak checkbox dengan nama *check_list* yang dicentang.

- Baris 263

Pada Baris 263 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 264

Pada baris 264 ini berisi pendeklarasian *variable* dengan nama *\$config* yang memiliki isi data berbentuk array.

- Baris 265
Pada baris 265 ini ditetapkan untuk nama 'mailtype' akan di isi dengan 'html' yang memiliki tipe data string.
- Baris 266
Pada baris 266 ini ditetapkan untuk nama 'charset' akan di isi dengan 'utf-8' yang memiliki tipe data string.
- Baris 267
Pada baris 267 ini ditetapkan untuk nama 'protocol' akan di isi dengan 'smtp' yang memiliki tipe data string.
- Baris 268
Pada baris 268 ini ditetapkan untuk nama 'smtp_host' akan di isi dengan 'ssl://smtp.gmail.com' yang memiliki tipe data string.
- Baris 269
Pada baris 269 ini ditetapkan untuk nama 'smtp_user' akan di isi dengan 'pdg.rutanbandung@gmail.com' yang memiliki tipe data string.
- Baris 270

Pada baris 270 ini ditetapkan untuk nama 'smtp_pass' akan di isi dengan 'rutan526801' yang memiliki tipe data string.

- Baris 271

Pada baris 271 ini ditetapkan untuk nama 'smtp_port' akan di isi dengan 465 yang memiliki tipe data number.

- Baris 272

Pada baris 272 ini ditetapkan untuk nama 'crlf' akan di isi dengan '\r\n' yang memiliki tipe data string.

- Baris 273

Pada baris 273 ini ditetapkan untuk nama 'newline' akan di isi dengan '\r\n' yang memiliki tipe data string.

- Baris 274

Pada baris 274 ini berisi penutup menggunakan kurung tutup dari seluruh isi *variable array* yang dimulai pada baris 264.

u. *Function send_email_all* (2)

```
276     for ($i=0; $i < $jumlahData ; $i++) {  
277         $id123 = $ArrData[$i];  
278  
279         $this->db->join('tb_pegawai','tb_potongan.id_pegawai = tb_pegawai.nip');  
280         $this->db->select('tb_pegawai.id as id, tb_pegawai.nip as nip,  
            tb_pegawai.email as email, tb_pegawai.rekening as rekening,  
            tb_pegawai.gaji_bersih as gaji_bersih, tb_pegawai.nama_pegawai as  
            nama_pegawai, tb_potongan.bulan_tahun as bulan_tahun,  
            tb_potongan.arisan_pengayoman as arisan_pengayoman, tb_potongan.idw as  
            idw, tb_potongan.koperasi as koperasi, tb_potongan.tabungan_dw as  
            tabungan_dw, tb_potongan.olahraga as olahraga, tb_potongan.bri as bri,  
            tb_potongan.bjb as bjb, tb_potongan.skp as skp, tb_potongan.lain_lain  
            as lain_lain, tb_potongan.arisan_dw as arisan_dw,  
            tb_potongan.arisan_100 as arisan_100, tb_potongan.total as total,  
            tb_potongan.gaji_diterima as gaji_diterima');  
281  
282         $data = $this->db->get_where('tb_potongan', array('tb_potongan.id' => $id123))->  
            row();
```

Gambar 4.124 *Function send_email_all* (2)

- Baris 276

Pada Baris 276 ini berisi *for* yang berisi variable \$i adalah 0, selagi \$i kurang dari \$jumlahData seperti pada baris 262 maka \$i akan selalu bertambah 1.

- Baris 277

Pada baris 277 ini berisi pendeklarasian *variable* dengan nama \$id123 yang memiliki isi hasil dari

\$ArrData yang menjadi array dengan berisi variable \$i pada baris 276.

- Baris 278

Pada Baris 278 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 279

Pada baris 279 ini berisi deklarasi *join table* antara tabel 'tb_pegawai' dan tabel 'tb_potongan', persamaannya diatur untuk tb_pegawai adalah nip dan untuk 'tb_potongan' adalah id_pegawai.

- Baris 280

Pada Baris 263 berisi pemanggilan data pada tabel 'tb_potongan' dan 'tb_pegawai', daftarnya adalah sebagai berikut:

1. Tb_pegawai.id menjadi id
2. Tb_pegawai.nip menjadi nip
3. Tb_pegawai.email menjadi email
4. Tb_pegawai.rekening menjadi rekening
5. Tb_pegawai.gaji_bersih menjadi gaji bersih
6. Tb_pegawai.nama_pegawai menjadi nama pegawai
7. Tb_potongan.bulan_tahun menjadi bulan tahun
8. Tb_potongan.arisan_pengayoman menjadi arisan pengayoman
9. Tb_potongan.idw menjadi idw

10. Tb_potongan.koperasi menjadi koperasi
11. Tb_potongan.tabungan_dw menjadi tabungan
dw
12. Tb_potongan.olahraga menjadi olahraga
13. Tb_potongan.bri menjadi bri
14. Tb_potongan.bjb menjadi bjb
15. Tb_potongan.skp menjadi skp
16. Tb_potongan.lain_lain menjadi lain lain
17. Tb_potongan.arisan_dw menjadi arisan dw
18. Tb_potongan.arisan_100 menjadi arisan 100
19. Tb_potongan.total menjadi total
20. Tb_potongan.gaji_diterima menjadi gaji
diterima

- Baris 281

Pada Baris 281 berisi 1 enter untuk membuat baris koding menjadi lebih rapi.

- Baris 282

Pada baris 282 ini berisi pendeklarasian *variable* dengan nama \$data yang memiliki isi hasil pemanggilan seluruh data yang sudah didaftarkan pada baris 280 di tabel 'tb_potongan' dengan filter yaitu 'tb_potongan.id' harus sama dengan hasil yang didapat dari \$id123 di baris 277.

v. *Function send_email_all* (3)[illegible]

Gambar 4.125 *Function send_email_all* (3)

- Baris 284

Pada Baris 284 ini berisi pendeklarasian *variable* dengan nama \$a1 dengan tipe data string yang memiliki isi ‘Slip Gaji Pegawai Rutan Klas I Bandung Bulan’ disambung dengan \$data_bulan hasil pemanggilan data di baris 280.

- Baris 285

Pada Baris 285 ini berisi pendeklarasian *variable* dengan nama \$a2 dengan tipe data string yang

memiliki isi 'Pemberitahuan Gaji Pegawai Rutan' disambung dengan 2 enter.

- Baris 286

Pada Baris 286 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'Nama Pegawai :.' disambung dengan \$nama_pegawai hasil pemanggilan data di baris 280 dan ditambah 1 enter.

- Baris 287

Pada Baris 287 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'Bulan :.' disambung dengan \$bulan_tahun hasil pemanggilan data di baris 280 dan ditambah 1 enter.

- Baris 288

Pada Baris 288 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'Rekening :.' disambung dengan \$rekening hasil pemanggilan data di baris 280 dan ditambah 1 enter.

- Baris 289

Pada Baris 289 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya

adalah 'Gaji :' disambung dengan \$gaji_bersih hasil pemanggilan data di baris 280 dan ditambah 2 enter.

- Baris 289

Pada Baris 289 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'Gaji :' disambung dengan \$gaji_bersih hasil pemanggilan data di baris 280 dan ditambah 2 enter.

- Baris 290

Pada Baris 290 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'Daftar Potongan :' disambung dengan 1 enter.

- Baris 291

Pada Baris 291 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'Arisan Pengayoman : Rp. ' disambung dengan \$arisan_pengayoman hasil pemanggilan data di baris 280 dan ditambah 1 enter.

- Baris 292

Pada Baris 292 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'IDW: Rp. ' disambung dengan \$idw hasil pemanggilan data di baris 280 dan ditambah 1 enter.

- Baris 293

Pada Baris 293 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'Koperasi : Rp. ' disambung dengan \$koperasi hasil pemanggilan data di baris 280 dan ditambah 1 enter.

- Baris 294

Pada Baris 294 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'Tabungan DW: Rp. ' disambung dengan \$tabungan_dw hasil pemanggilan data di baris 280 dan ditambah 1 enter.

- Baris 295

Pada Baris 295 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'Olahraga: Rp. ' disambung dengan \$olahraga hasil pemanggilan data di baris 280 dan ditambah 1 enter.

- Baris 296

Pada Baris 296 ini berisi lanjutan isi dari *variable* dengan nama \$a2 dengan tipe data string dan isinya adalah 'BRI: Rp. ' disambung dengan \$bri hasil pemanggilan data di baris 280 dan ditambah 1 enter.