

SISTEM APLIKASI TOTALAN



MANUAL BOOK DEVELOPMENT

PT.SIANTARTOP, Tbk

BEKASI

DAFTAR ISI

DAFTAR ISI	0
PENDAHULUAN.....	1
BAB I FORM LOGIN.....	2
BAB II DATA MASTER.....	3
2.1 Data Pelaksana.....	3
2.2 Jenis Packing	4
2.3 Master Masalah.....	7
BAB III TRANSAKSI BALANCING.....	8
3.1 Transaksi Balancing	8
3.2 Detail Transaksi Balancing	9
3.4 Data Pendukung Kesimpulan Akhir Shift.....	17
3.5 Data Raport	17
4.1 Monitoring Progress.....	21
4.2 Insentif	23
4.3 Monitoring BS Bandroll	26
4.4 Monitoring Performance.....	27

PENDAHULUAN

Sistem adalah kumpulan hal atau komponen yang saling terhubung dengan prosedur tertentu sehingga membentuk kesatuan guna mencapai tujuan tertentu. Aplikasi merupakan suatu perangkat lunak (*software*) yang memiliki fitur-fitur yang dapat diakses pengguna. Sehingga sistem aplikasi dapat diartikan sebagai sekumpulan komponen-komponen yang saling terhubung dengan prosedur tertentu sehingga membentuk fitur-fitur yang dapat diakses pengguna guna mencapai tujuan Perusahaan. Sistem aplikasi dibuat untuk memudahkan penggunaannya dalam mengolah data sehingga menghasilkan informasi yang dapat membantu manajemen mengambil keputusan guna mencapai tujuan Perusahaan dengan efektif dan efisien.

Sistem Aplikasi Totalan merupakan sistem aplikasi yang dibuat untuk memudahkan tim totalan mencatat dan mengolah data harian produksi, seperti output dan bs produksi untuk dikalkulasi nilai balancing antara planning, standar dan hasil real produksi. Sistem aplikasi totalan ini merupakan aplikasi berbasis web yang dibuat menggunakan Bahasa pemrograman PHP dengan framework codeigniter 3 dan database MySql yang dapat digunakan user menggunakan browser seperti firefox atau chrome.

Manual book ini berisi informasi tentang perancangan sistem aplikasi Totalan sehingga dapat membantu developer selanjutnya mengembangkan sistem yang sudah ada.

BAB I FORM LOGIN

Form login diambil dari table login. Di proses di controller LogUser function ceklogin2.

```
public function ceklogin2(){
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $login = $this->m->level_login($username, $password);
    if($login){
        foreach ($login as $row);
            // $this->session->set_userdata('iduser',$row->nip);
            $this->session->set_userdata('iduser',$row->niplog);
            $this->session->set_userdata('leveluser',$row->level);
            $this->session->set_userdata('approve',$row->approve);
            $this->session->set_userdata('nama',$row->nama);
            $this->session->set_userdata('divisi',$row->divisi);
            $this->session->set_userdata('gambar',$row->gambar);
            $this->session->set_userdata('bagian',$row->bagian);
            $this->session->set_userdata('alamat',$row->alamat);
            $this->session->set_userdata('tlp',$row->no_tlp);
            if($this->session->userdata('leveluser')=="Totalan"){
                redirect(base_url("Home"));
            }else if($this->session->userdata('leveluser')=="admin"){
                redirect(base_url("Home"));
            }else if($this->session->userdata('leveluser')=="Produksi"){
                redirect(base_url("Downtime"));
            }else if($this->session->userdata('leveluser')=="Manager"){
                redirect(base_url("Home"));
            }
        }
    }else{
        $this->session->set_flashdata('error_msg','Username atau Password tidak terdaftar');
        redirect(base_url("LogUser"));
    }
}
```

Kolom level pada table login ini merupakan hardcode yang digunakan untuk memisahkan hak akses aplikasi. Secara keseluruhan ada 4 level hak akses, namun aplikasi totalan dasarnya hanya memiliki 3 level hak akses, karena level akses 'Produksi' ada setelah pengembangan bahwa data downtime diinput oleh tim produksi. Sehingga user yang memiliki level akses 'Produksi' hanya dapat menginputkan data downtime, tidak dengan fitur lainnya pada aplikasi totalan.

level
<input type="checkbox"/> Manager
<input type="checkbox"/> admin
<input type="checkbox"/> Produksi
<input type="checkbox"/> Totalan

BAB II DATA MASTER

2.1 Data Pelaksana

Data pelaksana diambil dari table pelaksana yang berelasi dengan table insentif_premi dan jabatan.

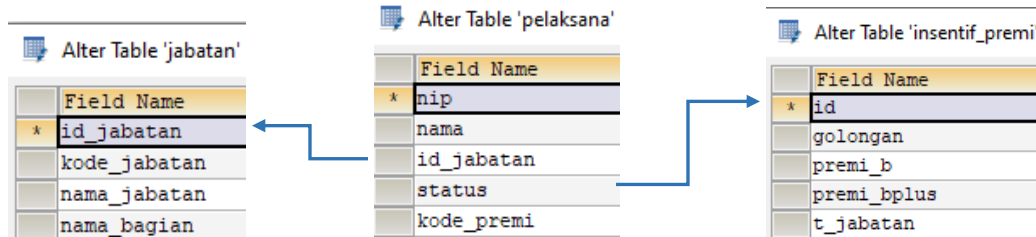


Table pelaksana memiliki 2 jenis status yang digunakan sebagai hardcode untuk memisahkan pelaksana yang masih aktif dan sudah nonaktif.

status
<input type="checkbox"/> aktif
<input type="checkbox"/> nonaktif

Table insentif_premi merupakan besaran nominal premi yang didapat pelaksana tersebut. Berikut contoh data besaran premi yang digunakan untuk pemrosesan nilai premi/insentif perbulannya. Masing-masing kolom dan besarnya digunakan sesuai aturan kebijakan manajemen (customize).

	id	golongan	premi_b	premi_bplus	t_jabatan
<input type="checkbox"/>	Astdiv	(NULL)	800000	(NULL)	600000
<input type="checkbox"/>	Kabag	(NULL)	600000	(NULL)	400000
<input type="checkbox"/>	Karu Produksi A	A	400000	(NULL)	250000
<input type="checkbox"/>	Karu Produksi B	B	300000	(NULL)	200000
<input type="checkbox"/>	Karu Teknik	(NULL)	250000	350000	(NULL)
<input type="checkbox"/>	Staff Produksi	(NULL)	550000	(NULL)	400000
<input type="checkbox"/>	Staff Teknik	(NULL)	550000	(NULL)	400000

Table jabatan berisi hardcode id_jabatan yang digunakan untuk pemrosesan nilai kuantitatif transaksi perjam pada totalan sehingga perubahan data pada table ini perlu diperbaharui perubahan pada proses report

id_jabatan	kode_jabatan	nama_jabatan	nama_bagian
1	J-0419-00001	KADIV	(NULL)
2	J-0419-00002	ASTDIV PRODUKSI	BISKUIT & MIE
3	J-0419-00003	STAFF TEHNIK	TEHNIK
4	J-0822-00004	STAFF PRODUKSI	KERUPUK
5	J-0822-00005	KARU M-UP	KERUPUK
6	J-0819-00006	KARU TEHNIK M-UP	KERUPUK
7	J-0819-00007	KARU M-LOW	KERUPUK
9	J-0722-00009	STAFF PRODUKSI	MIE
10	J-0722-00010	STAFF PRODUKSI	BISKUIT
11	J-0722-00011	KARU PACKING MIE HAN	MIE
12	J-0722-00012	KARU PACKING MIE KOT	MIE
13	J-0722-00013	KARU OLAH	KERUPUK
14	J-0722-00014	KARU OLAH	MIE
16	J-0722-00016	KARU OLAH	BISKUIT
17	J-0722-00017	KARU PACKING GORIO	BISKUIT
18	J-0722-00018	KARU PACKING POTATO	BISKUIT
20	J-0822-00020	KARU TEHNIK M-LOW	KERUPUK
21	J-0722-00021	TEHNIK PROD MK	MIE

22	J-0822-00001	ASTDIV PRODUKSI	KERUPUK
24	J-0822-00024	ASTDIV PRODUKSI	TEHNIK
25	J-0822-00025	TEHNIK PROD M-LOW	KERUPUK
26	J-0822-00026	TEHNIK PROD M-UP	KERUPUK
27	J-0822-00027	TEHNIK PROD MH	MIE
28	J-0822-00028	KARU TEHNIK OLAH	KERUPUK
29	J-0822-00029	KARU TEHNIK OLAH	BISKUIT
30	J-0822-00030	KARU TEHNIK PACK	BISKUIT
31	J-0822-00031	KARU TEHNIK OLAH	MIE
32	J-0822-00032	KARU TEHNIK PACK MK	MIE
33	J-0822-00033	OPERATOR PACK GORIO	BISKUIT
34	J-0822-00034	OPERATOR PACK POTATO	BISKUIT
35	J-0822-00035	OPERATOR MH	MIE
36	J-0822-00036	OPERATOR MK	MIE
37	J-0822-00037	OPERATOR PACK M-UP	KERUPUK
38	J-0822-00038	OPERATOR PACK M-LOW	KERUPUK
39	J-1022-00039	KARU TEHNIK PACK MH	MIE

Pengelolaan data pelaksana diproses di controller Pelaksana dan model Pelaksana_m. berikut sample source codenya.

Controller	Model
<pre> public function index(){ \$data= array("konten" => "Pelaksana/index"); \$data['Pelaksana'] = \$this->m->getPelaksana(); \$data['Bagian'] = \$this->m->getBagian(); \$this->load->view('template/header'); \$this->load->view(\$this->menu); \$this->load->view("Pelaksana/index",\$data); } public function Tambah(){ \$data= array("konten" => "Pelaksana/tambah"); \$data['Pelaksana'] = \$this->m->getJabatan(); \$this->load->view('template/header'); \$this->load->view(\$this->menu); \$this->load->view("Pelaksana/tambah",\$data); \$this->load->view('template/footer'); } public function Ubah(\$id){ \$data['Pelaksana'] = \$this->m->getUbah(\$id); \$data['Jabatan'] = \$this->m->getJabatan(); \$data['Premi'] = \$this->db->query("SELECT * FROM insentif_premi")->result(); \$this->load->view('template/header'); \$this->load->view(\$this->menu); \$this->load->view("Pelaksana/ubah",\$data); \$this->load->view('template/footer'); } </pre>	<pre> public function getPelaksana() { \$this->db->select('nip,nama,jabatan.nama_jabatan as nama_jabatan, nama_bagian, jabatan.nilai_premi as nilai_premi, status, premi_b'); \$this->db->from('pelaksana'); \$this->db->join('jabatan', 'jabatan.id_jabatan = pelaksana.id_jabatan'); \$this->db->join('insentif_premi', 'insentif_premi.id = pelaksana.kode_premi','left'); \$query = \$this->db->get(); if(\$query->num_rows() > 0){ return \$query->result(); }else{ return false; } } public function getJabatan(){ \$query = \$this->db->get('jabatan'); if(\$query->num_rows() > 0){ return \$query->result(); }else{ return false; } } public function getUbah(\$id){ \$this->db->where('nip',\$id); \$this->db->join('jabatan','jabatan.id_jabatan=pe laksana.id_jabatan'); \$query = \$this->db->get('pelaksana'); if(\$query->num_rows() > 0){ return \$query->row(); }else{ return false; } } public function getBagian(){ \$query = \$this->db->get('bagian'); if(\$query->num_rows() > 0){ return \$query->result(); }else{ return false; } } </pre>

2.2 Jenis Packing

Jenis packing menggunakan data pada table jenis yang berelasi ke table bagian, jenis_extruder (bagian mie hancur dan pengolahan pufftwist), jenis_gorengan (bagian gorengan), jenis_standar (standar persentase bagian lainnya), dan mesin (standar mesin bagian lainnya).

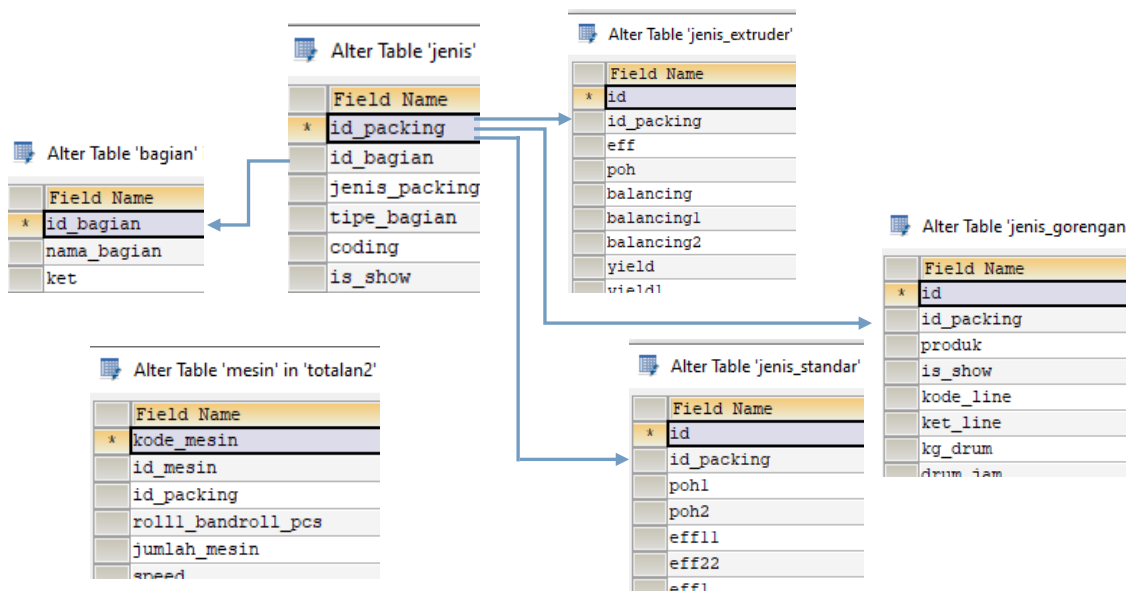


Table bagian berisi hardcode yang digunakan untuk menentukan format sheet totalan. Sheet yang tampil pada transaksi tergantung id_bagian pada table ini.

	id_bagian	nama_bagian	ket
<input type="checkbox"/>	1	INLINE MIE KOTAK	(NULL)
<input type="checkbox"/>	2	MIE HANCUR	(NULL)
<input type="checkbox"/>	3	PACKING KRUPUK M-UP	(NULL)
<input type="checkbox"/>	4	PACKING KRUPUK M-LOW	(NULL)
<input type="checkbox"/>	5	INLINE BISKUIT GORIORIO	(NULL)
<input type="checkbox"/>	6	INLINE BISKUIT POTATO	(NULL)
<input type="checkbox"/>	7	NONINLINE BISKUIT GORIORIO	(NULL)
<input type="checkbox"/>	8	GORENGAN	(NULL)
<input type="checkbox"/>	9	PENGOLAHAN PUFF TWIST	(NULL)
<input type="checkbox"/>	10	PENGOLAHAN MIE HANCUR	pengolahan mie hancur, sheet linv
<input type="checkbox"/>	11	NONINLINE BISKUIT POTATO	(NULL)

Table jenis berisi nama item produk, tipe_bagian pada id_bagian 9 memiliki 2 tipe berbeda tergantung kebutuhannya.

id_packing	id_bagian	jenis_packing	tipe_bagian	coding	is_show
P-0823-00277	1	MIE GEMEZ ENAAK CBP 2000	(NULL)	(NULL)	1
P-0723-00275	2	SPIX MIE GORENG SAMBAL BALADO PREMIUM	(NULL)	(NULL)	1
P-0722-00239	3	TIC-TIC BAWANG CBP 2000	(NULL)	1NLTCEWGBE	1
P-0722-00237	4	TWISTKO LOKAL	(NULL)	1FTWKJBQBE	1
P-0722-00250	5	DEO GORIORIO VANILA ISI 2	(NULL)		1
P-0722-00252	6	DEO GO POTATO	(NULL)		1
P-1122-00260	7	DEO GORIORIO VANILLA ISI 4	(NULL)	(NULL)	1
P-1222-00264	8	GORENGAN	(NULL)	(NULL)	1
P-1222-00265	9	JARI-JARI	1	1FJJRJBQBC	1
P-1222-00267	9	TWISTKO LOKAL	2	1FTWKJBQBE	1
P-1222-00269	10	PENGOLAHAN MIE HANCUR	(NULL)	(NULL)	1
P-0823-00276	11	BRIO GO POTATO	(NULL)	(NULL)	1

Setiap bagian memiliki format standar yang berbeda. Table jenis_extruder merupakan table yang berisi standar-standar untuk bagian pengolahan puff twist, baik standar kapasitas mesin maupun standar persentase balancing. Table jenis_gorengan merupakan table yang berisi standar-standar untuk bagian gorengan, standar kapasitas mesin dan standar persentase balancing. Sedangkan untuk bagian lainnya dapat dilihat pada table jenis_standar untuk standar persentase balancing dan table mesin untuk standar kapasitas

mesin. Pemrosesan data master jenis packing dan standar-standarnya diproses di controller Jenis dan model Jenis_m. Berikut sample codenya.

```
public function index()
{
    $data = array(
        "konten" => "Jenis/index"
    );
    $this->db->select('id_packing,bagian.nama_bagian,
        bagian.id_bagian,jenis_packing, is_show');
    $this->db->from('jenis');
    $this->db->join('bagian', 'bagian.id_bagian =
        jenis.id_bagian');
    $this->db->order_by('jenis.is_show DESC,
        bagian.id_bagian,jenis.jenis_packing');
    $query = $this->db->get();
    $data['Jenis'] = $query->result();
    $this->load->view('template/header');
    $this->load->view($this->menu);
    $this->load->view("Jenis/index", $data);
    $this->load->view('template/footer');
}

public function StandarExtruder($id)
{
    $data = array(
        "konten" => "Jenis/index"
    );
    $this->db->select('*');
    $this->db->from('jenis_extruder');
    $this->db->where('id_packing', $id);
    $this->db->order_by('id', 'DESC');
    $query = $this->db->get();
    $data['Standar'] = $query->result();
    $this->db->select('*');
    $this->db->from('jenis_extruder');
    $this->db->where('id_packing', $id);
    $this->db->order_by('id', 'DESC');
    $query = $this->db->get();
    $data['master'] = $query->row();
    $data['id_packing'] = $id;
    $this->db->select('*');
    $this->db->from('jenis');
    $this->db->where('id_packing', $id);
    $query = $this->db->get();
    $data['jenis'] = $query->row();
    $data['id_bagian'] = $data['jenis']->id_bagian;
    $data['id_bag22'] = $data['jenis']->id_bagian;
    $this->load->view('template/header');
    $this->load->view($this->menu);
    $this->load->view("Jenis/standarExtru", $data);
    $this->load->view('template/footer');
}
```

```
public function StandarGorengan($id)
{
    $data = array(
        "konten" => "Jenis/index"
    );
    $this->db->select('*');
    $this->db->from('jenis_gorengan');
    $this->db->where('id_packing', $id);
    $this->db->order_by('is_show', 'DESC');
    $this->db->order_by('produk', 'DESC');
    $this->db->order_by('id', 'DESC');
    $query = $this->db->get();
    $data['Standar'] = $query->result();
    $this->db->select('*');
    $this->db->from('jenis_gorengan');
    $this->db->where('id_packing', $id);
    $this->db->order_by('id', 'DESC');
    $query = $this->db->get();
    $data['master'] = $query->row();
    $data['id_packing'] = $id;
    $this->db->select('*');
    $this->db->from('jenis');
    $this->db->where('id_packing', $id);
    $query = $this->db->get();
    $data['jenis'] = $query->row();
    $data['id_bagian'] = $data['jenis']->id_bagian;
    $this->load->view('template/header');
    $this->load->view($this->menu);
    $this->load->view("Jenis/standarGorengan",
        $data);
    $this->load->view('template/footer');
}

public function Tambah()
{
    $data['Jenis'] = $this->m->getBagian();
    $data['Kode'] = $this->m->getkodejenis();
    $this->load->view('template/header');
    $this->load->view($this->menu);
    $this->load->view("Jenis/tambah", $data);
    $this->load->view('template/footer');
}
```

2.3 Master Masalah

Alter Table 'sumber_masalah'

Field Name
* kode_masalah
id_bagian
id_mesin
kategori
kategori2
permasalahan
sumber_masalah
penyelesaian
user_input
tanggal_input

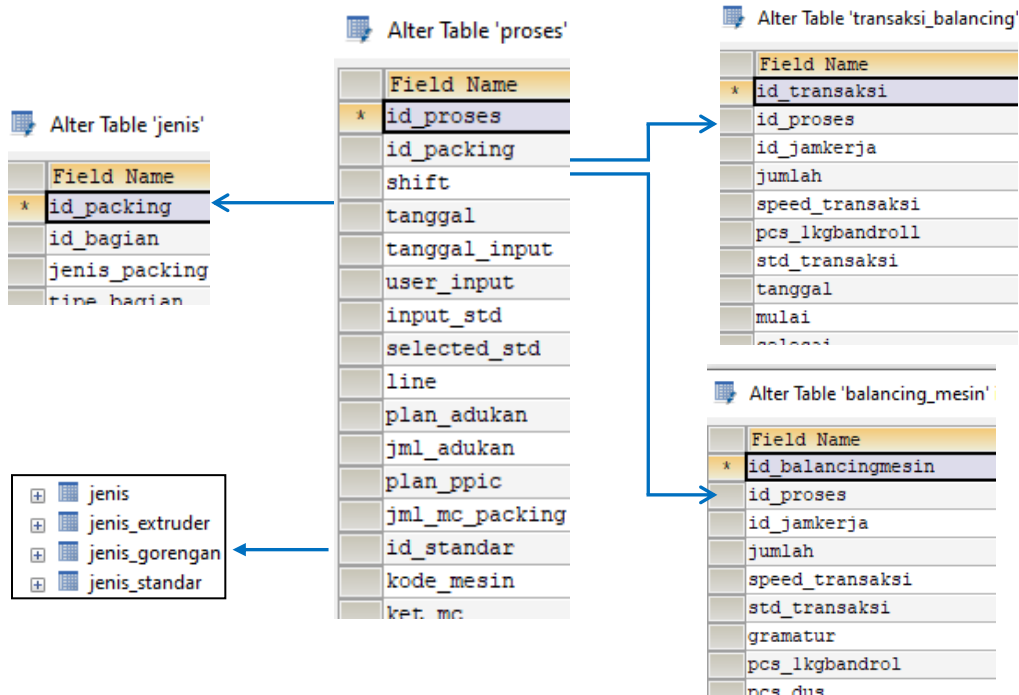
Table sumber_masalah berelasi dengan table bagian. Data master sumber masalah akan muncul di transaksi balancing sesuai dengan bagiannya. Data master downtime di proses pada controller MasterMasalah dan model MasterMasalah_m. Berikut sample codenya,

Controller	Model
<pre> public function index(){ \$data= array("konten" => "MasterDowntime/index"); \$data['Masalah'] = \$this->m->getMasalah(); \$this->load->view('template/header'); \$this->load->view(\$this->menu); \$this->load->view("MasterDowntime/index",\$data); \$this->load->view('template/footer'); } public function Tambah(){ \$data['Bagian']=\$this->m->getBagian(); \$data['Mesin'] = \$this->m->getMesin(); \$this->load->view('template/header'); \$this->load->view(\$this->menu); \$this->load->view("MasterDowntime/tambah", \$data); \$this->load->view('template/footer'); } public function Simpan(){ \$result=\$this->m->getSimpan(); if(\$result){ \$this->session->set_flashdata('success_msg','Data berhasil ditambahkan'); }else{ \$this->session->set_flashdata('error_msg','Data gagal ditambahkan'); } redirect(base_url('MasterMasalah')); } </pre>	<pre> public function getMasalah(){ \$this->db->select('*'); \$this->db->from('sumber_masalah s'); \$this->db->join('bagian b','b.id_bagian = s.id_bagian'); \$this->db->join('master_mesin m','m.id_mesin = s.id_mesin','left'); \$this->db->order_by('s.id_bagian'); \$this->db->order_by('s.kategori'); \$query = \$this->db->get(); if(\$query->num_rows() > 0){ return \$query->result(); }else{ return false; } } public function getBagian(){ \$query = \$this->db->get('bagian'); if(\$query->num_rows() > 0){ return \$query->result(); }else{ return false; } } </pre>

BAB III TRANSAKSI BALANCING

3.1 Transaksi Balancing

Transaksi balancing di proses di controller Balancing dan model Balancing_m. Table yang digunakan pada transaksi ini adalah table proses yang berelasi dengan table jenis, table ini juga menjadi foreign key di table balancing_mesin, transaksi_balancing,



Berikut function-function pada Controller Balancing yang berkaitan dengan transaksi balancing,

```

public function index($tgl = null)
{
    ...
}

public function Bagian($id_bagian)
{
    ...
}

public function Tambah()
{
    ...
}

public function Simpan()
{
    ...
}

public function Ubah($id)
{
    ...
}

public function Update()
{
    ...
}

public function Hapus($id)
{
    ...
}

```

} Untuk menampilkan list semua transaksi diurutkan dari yang terbaru

} Untuk menampilkan list transaksi per bagian (menu transaksi per bagian) diurutkan dari yang terbaru.

Untuk function lain memiliki fungsi sesuai nama functionnya, function ini untuk pengolahan data header transaksi balancing.

Untuk function simpan setelah informasi header transaksi balancing disimpan (table proses), user akan diarahkan ke detail header transaksi seperti informasi mesin yang digunakan (table balancing_mesin) dan standar yang digunakan.

Berikut sample codenya,

<pre> public function Tambah() { \$sql = "SELECT * FROM jenis p left join bagian b ON b.id_bagian=p.id_bagian order by p.id_bagian, p.jenis_packing "; \$query = \$this->db->query(\$sql); \$data['Proses'] = \$query->result(); \$this->load->view('template/header'); \$this->load->view(\$this->menu); \$this->load->view("Balancing/Tambah", \$data); \$this->load->view('template/footer'); } </pre>	<pre> public function Simpan() { \$field = array('id_packing' => \$this->input->post('jenis'), 'shift' => \$this->input->post('shift'), 'tanggal' => \$this->input->post('tanggal'), 'user_input' => \$this->session->userdata('iduser')); \$this->db->set('tanggal_input', 'NOW()', FALSE); \$this->db->insert('proses', \$field); \$id_proses = \$this->db->insert_id(); \$proses = \$this->db->query("SELECT id_packing,tanggal from proses where id_proses='\$id_proses'")->row(); \$id_packing= \$proses->id_packing; \$tanggal = \$proses->tanggal; \$jenis = \$this->db->query("SELECT id_bagian from jenis where id_packing='\$id_packing'")->row(); \$id_bagian = \$jenis->id_bagian; \$versi=''; if (\$id_bagian==1 && \$tanggal>='2023-07-01') { \$versi='&versi=2'; } redirect(base_url() . 'Balancing/Detail/?id_bagian=' . \$id_bagian . '&id_proses=' . \$id_proses . '&id_packing=' . \$id_packing.\$versi); } </pre>
---	---

3.2 Detail Transaksi Balancing

Untuk pemrosesan detail header seperti pemilihan standard dan detail mesin diproses pada controller Balancing function SimpanProsesNew.

<pre> public function SimpanProsesNew() { \$id = \$this->input->post('id_proses'); \$id2 = \$this->input->post('id_packing'); \$id3 = \$this->input->post('id_bagian'); \$tipe_std=null; if (\$this->input->post('id_bagian') == 5 or \$this->input->post('id_bagian') == 7) { ... } else if (\$this->input->post('id_bagian') == 6 OR \$this->input->post('id_bagian')==11) { ... } else if (\$this->input->post('id_bagian')==1){ ... } else { ... } // redirect(base_url('Balancing/Detail/?id_bagian=' . \$id3 . '&id_proses=' . \$id . '&id_packing=' . \$id2 . '&std=' . \$std)); redirect(base_url('Balancing/Detail/?id_bagian=' . \$id3 . '&id_proses=' . \$id . '&id_packing=' . \$id2 . '&std=' . \$std.'&versi=2')); // redirect(base_url('Balancing/Detail/?id_bagian=' . \$id3 . '&id_proses=' . \$id . '&id_packing=' . \$id2 . '&std=' . \$std.(\$Proses- } </pre>

Pada function ini akan ditentukan standar perjam dan detail perhitungan standar perjam tersebut, serta standar-standar lain yang akan menjadi acuan untuk data transaksi perjamnya. Berikut sample codenya,

<p>Proses detail header transaksi gorio</p> <pre> \$dd = \$this->input->post('std1check'); if (isset(\$dd)) { \$std = \$this->input->post('std1'); if (\$this->input->post('id_bagian') == 7) { \$tipe_standar = 'pengolahan'; } else { \$tipe_standar = null; } } else { \$std = \$this->input->post('std2'); } </pre>	<p>Proses detail header transaksi krupuk</p> <pre> \$Mesin = \$this->input->post('kode_mesin'); \$std = 0; \$jumlah = 0; \$field = array(); if (\$Mesin) { foreach (\$Mesin as \$k) { \$d = \$this->db->query("SELECT * FROM mesin where kode_mesin='\$k'")->row(); //update 220919 if (\$this->input->post('id_bagian') == 1) { </pre>
---	--

<pre> if (\$this->input->post('id_bagian') == 7) { \$tipe_standar = 'packing'; } else { \$tipe_standar = null; } } if(\$this->input->post('id_bagian')==7 && \$tipe_standar=='packing') { \$Mesin = \$this->input->post('kode_mesin'); \$std = 0; \$jumlah = 0; \$field = array(); if (\$Mesin) { foreach (\$Mesin as \$k) { \$d = \$this->db->query("SELECT * FROM mesin where kode_mesin='\$k'")->row(); \$std += (float)\$this->input->post('std-' . \$d->id_mesin); \$jumlah += (float) \$this->input->post('jmlmc-' . \$d->id_mesin); //insert balancing_mesin if (\$this->input->post('jmlmc-' . \$d->id_mesin) > 0) { \$field = array('id_proses'=> \$this->input->post('id_proses'), 'jumlah'=> \$this->input->post('jmlmc-' . \$d->id_mesin), 'speed_transaksi' => \$this->input->post('speed-' . \$d->id_mesin), 'std_transaksi'=> \$this->input->post('std-' . \$d->id_mesin), 'nama_mesin'=> \$this->input->post('nama_mesin-' . \$d->id_mesin), 'pcs_1kgbandrol'=> \$this->input->post('pcs_1kgbandrol-' . \$d->id_mesin), 'pcs_dus'=> \$this->input->post('pack_dus-' . \$d->id_mesin), 'id_mesin'=> \$d->id_mesin, 'kode_mesin'=> \$k, 'tanggal_input' => date('Y-m-d H:i:s'), 'user_input' =>\$this->session->userdata('iduser')); \$this->db->insert('balancing_mesin', \$field); } } } \$field = array('selected_std' => \$std, 'tipe_standar' => \$tipe_standar, 'line'=> \$this->input->post('line'), 'jml_adukan'=> \$this->input->post('jml_adukan'), 'plan_ppic'=> \$this->input->post('plan_ppic'), 'jml_mc_packing'=> \$this->input->post('jml_mc'), 'id_standar'=> \$this->input->post('id_standar'), 'kode_mesin'=> \$this->input->post('id_mesin') </pre>	<pre> /pilih dulu pake std pengolahan atau packing if (\$this->input->post('pilihstd-' . \$d->id_mesin) == 'packing') { \$std += (float)\$this->input->post('std-' . \$d->id_mesin); } else { \$std = (float)\$this->input->post('std_olah-' . \$d->id_mesin); } } else { \$std += (float)\$this->input->post('std-' . \$d->id_mesin); } \$jumlah += (float)\$this->input->post('jmlmc-' . \$d->id_mesin); //insert balancing_mesin if (\$this->input->post('jmlmc-' . \$d->id_mesin) > 0) { \$field = array('id_proses'=> \$this->input->post('id_proses'), 'jumlah'=> \$this->input->post('jmlmc-' . \$d->id_mesin), 'speed_transaksi'=> \$this->input->post('speed-' . \$d->id_mesin), 'std_transaksi'=> \$this->input->post('std-' . \$d->id_mesin), 'nama_mesin'=> \$this->input->post('nama_mesin-' . \$d->id_mesin), 'gramatur'=> \$this->input->post('gramatur-' . \$d->id_mesin), 'pcs_1kgbandrol'=> \$this->input->post('pcs_1kgbandrol-' . \$d->id_mesin), 'pcs_dus'=> \$this->input->post('pcs_dus-' . \$d->id_mesin), 'id_mesin'=> \$d->id_mesin, 'kode_mesin'=> \$k, 'tanggal_input' => date('Y-m-d H:i:s'), 'user_input' => \$this->session->userdata('iduser')); \$this->db->insert('balancing_mesin', \$field); } //update selected std \$this->db->where('id_proses', \$this->input->post('id_proses')); \$this->db->update('proses', ["selected_std" => \$std, 'id_standar'=> \$this->input->post('id_standar')]); \$cek = \$this->db->affected_rows(); } } </pre>
---	--

```
);
$this->db->where('id_proses',
$this->input->post('id_proses'));
$this->db->update('proses', $field);
```

Sedangkan untuk data transaksi perjamnya, di proses dalam beberapa function berbeda sesuai dengan bagiannya.

<pre>public function simpanTransaksiPuffTwist() { ... }</pre>	<p>Untuk memproses transaksi perjam bagian pufftwist</p>
<pre>public function simpanTransaksiLinev() { ... }</pre>	<p>Untuk memproses transaksi perjam bagian pengolahan mie hancur</p>
<pre>public function simpanTransaksi() { ... }</pre>	<p>Untuk memproses transaksi perjam bagian lainnya, seperti Krupuk, Mie hancur, Biskuit</p>
<pre>//=====GORENGAN===== public function simpanTransaksiGorengan() { ... }</pre>	<p>Untuk memproses transaksi perjam bagian gorengan</p>

Berikut sample codenya,

```
simpanTransaksi()
$this->input->get('id_bagian');
$id = $this->input->post('id_proses');
$proses = $this->db->query("SELECT * FROM proses m where
m.id_proses='$id' limit 1")->row();
$id2 = $proses->id_packing;
$id_standar = $proses->id_standar;
if ($id_standar == null or $id_standar == '') {
    $standar = $this->db->query("SELECT * FROM
jenis_standar m where m.id_packing='$id2' ORDER BY id
DESC limit 1")->row();
} else {
    $standar = $this->db->query("SELECT * FROM
jenis_standar m where m.id='$id_standar' ORDER BY id
DESC limit 1")->row();
}
$tipejam = $this->input->post('tipe_jam');
$cekJamIstirahat = $this->input->post('is_istirahat');
if ($cekJamIstirahat) {
    $field = array(
        'id_proses'=> $this->input->post('id_proses'),
        'tanggal' => $this->input->post('tanggal'),
        'mulai'=> $this->input->post('mulai'),
        'selesai' => $this->input->post('selesai'),
        'tipe_jam'=> $tipejam,
        'tanggal_input'=> date('Y-m-d'),
        'ket_istirahat'=>
        $this->input->post('ket_istirahat'),
        'user_input'=> $this->session->userdata('iduser')
    );
} else {
    $jml_menit=(strtotime($this->input->post('selesai'))
- strtotime($this->input->post('mulai'))) / 60;
if ($jml_menit < 0
&&(strtotime($this->input->post('selesai')) >=
strtotime('00:00:00') &&
strtotime($this->input->post('selesai')) <
strtotime('01:00:00')) {
```

```
simpanTransaksiGorengan()
if (strtotime($this->input->post('mulai')) >
strtotime($this->input->post('selesai'))) {
    $datetime1 = new DateTime('2022-01-01 ' .
$this->input->post('mulai'));
    $datetime2 = new DateTime('2022-01-02 ' .
$this->input->post('selesai'));
} else {
    $datetime1 = new DateTime('2022-01-01 ' .
$this->input->post('mulai'));
    $datetime2 = new DateTime('2022-01-01 ' .
$this->input->post('selesai'));
}
$diff = $datetime1->diff($datetime2);
$jamgoreng = ($diff->h + ($diff->i / 60));
$jml_menit = $diff->i;
if ($diff->h > 0) {
    $jml_menit += $diff->h * 60;
}
$std = $jamgoreng * ($this->input->post('kg_drum') *
$this->input->post('drum_jam'));
$real_dus = (float)$this->input->post('real_dus');
$bs_gorengan =
(float)$this->input->post('kg_bandrol1') +
(float)$this->input->post('kg_bandrol2');
$bs_detail = $this->input->post('kg_bandrol1') . "+"
. $this->input->post('kg_bandrol2');
if ($real_dus == 0) {
    $pcg_bs = 0;
    $eff = 0;
} else {
    $pcg_bs = round($bs_gorengan / $std * 100, 2);
    $eff = round($real_dus / $std * 100, 2);
}
$menit_normatif =
(float)$this->input->post('menitnormatif');
if ($this->input->post('produk') == "AGING TIME"
```

```

$jml_menit = $jml_menit + 1440;
}
if ($jml_menit == 0) {
    redirect($_SERVER['HTTP_REFERER']);
}
$std = $this->input->post('std') * $jml_menit / 60;
$std_round= round($this->input->post('std') *
    $jml_menit / 60, 0);
$real_dus =
    (float)$this->input->post('real_dus1') +
    (float)$this->input->post('real_dus2') +
    (float)$this->input->post('real_dus');
$hasil = round($real_dus / $std, 4) * 100;
$dus = $this->input->post('pack_dus');
$standar_output =
    $this->input->post('standar_output');

if ($id_bagian == 6 or $id_bagian == 11) { //gopotato
    $total_waste_bandrol=
        round(($this->input->post('mc_pack') +
            $this->input->post('bs_teknik') +
            $this->input->post('mc_kospack')) /
            $this->input->post('pack_dus') *
            $this->input->post('pcs_1kg_bandrol'), 2);
    $total_waste_bandrol_pcg =
        round($total_waste_bandrol / $std * 100, 2);
    $bs_basah_dus =
        $this->input->post('bs_basah_dus');
    $bs_recgaido_dus =
        $this->input->post('bs_recgaido_dus');
    $bs_recpack_dus =
        $this->input->post('bs_recpack_dus');
    $bs_basah_pcg = round($bs_basah_dus / $std * 100, 2);
    $bs_recgaido_pcg = round($bs_recgaido_dus / $std *
        100, 2);
    $bs_recpack_pcg = round($bs_recpack_dus / $std * 100,
        2);
    $total_dus_bsolah = round($bs_recgaido_dus +
        $bs_basah_dus, 2);
    $total_pcg_bsolah = round($total_dus_bsolah / $std *
        100, 2);
    $dus_waste = round($total_waste_bandrol +
        $bs_recpack_dus, 2);
    $total_waste_bandrol=
        round(($this->input->post('mc_pack') +
            $this->input->post('bs_teknik') +
            $this->input->post('mc_kospack')) /
            $this->input->post('pack_dus') *
            $this->input->post('pcs_1kg_bandrol'), 2);
    $total_waste_bandrol_pcg=
        round($total_waste_bandrol / $std * 100, 2);
    if ($proses->tipe_standar == 'pengolahan') {
        $total_kg_bsolah =
            round($this->input->post('moulding') +
                $this->input->post('oven') +
                $this->input->post('recycle_gaido') +
                $this->input->post('sapuan'), 2);
        $total_dus_bsolah = round($total_kg_bsolah /
            $this->input->post('kg_dus'), 2);
        $total_pcg_bsolah = round($total_kg_bsolah / $std
            * 100, 2);
    } else {
        $total_kg_bsolah = $bs_basah_dus +
            $bs_recgaido_dus;
        $total_dus_bsolah = $bs_basah_dus +
            $bs_recgaido_dus;
    }
    OR $this->input->post('produk') == "PENAMBAHAN
    MINYAK"
    OR $this->input->post('produk') == "PERSIAPAN AWAL"
    OR $this->input->post('produk') == "OFF (ISTIRAHAT)"
    OR $this->input->post('produk') == "GANTI PRODUK"
    OR $this->input->post('produk') == "TRIAL PRODUK"
    OR $this->input->post('produk') == "WAKTU TUNGGU") {
        $menit_normatif = $jml_menit;
    }
    if ($menit_normatif == '' or $menit_normatif == 0) {
        $kg_normatif = 0;
        $pcg_normatif = 0;
    } else {
        $kg_normatif = ($menit_normatif / 60) * $std;
        if ($real_dus == 0) {
            $pcg_normatif = 0;
        } else {
            $pcg_normatif = round($kg_normatif / $std *
            100, 2);
        }
    }
    $menit_nonnormatif =
        (float)$this->input->post('menitnon');
    if ($menit_nonnormatif == '' or $menit_nonnormatif ==
    0) {
        $kg_nonnormatif = 0;
        $pcg_nonnormatif = 0;
    } else {
        $kg_nonnormatif = ($menit_nonnormatif / 60) *
        $std;
        if ($real_dus == 0) {
            $pcg_nonnormatif = 0;
        } else {
            $pcg_nonnormatif = round($kg_nonnormatif / $std
            * 100, 2);
        }
    }
    if ($real_dus == 0) {
        $balancing = 0;
    } else {
        $balancing = round(($eff + $pcg_bs +
        $pcg_normatif + $pcg_nonnormatif), 2);
    }
    $idstd = $this->input->post('id_standar');
    $standar = $this->db->query("SELECT * FROM
    jenis_gorengan m where m.id='$idstd' ORDER BY id DESC
    limit 1")->row();
    if ($eff >= $standar->std_eff) {
        $eff_grade = 'B';
        $eff_point = 7;
    } else if ($eff >= $standar->std_eff1 && $eff <
    $standar->std_eff) {
        $eff_grade = 'C';
        $eff_point = 6;
    } else {
        $eff_grade = 'K';
        $eff_point = 5;
    }
    if ($pcg_nonnormatif <= 0) {
        $dt_grade = 'B';
        $dt_point = 7;
    } else if ($pcg_nonnormatif <= 15 && $pcg_nonnormatif
    > 0) {
        $dt_grade = 'C';
        $dt_point = 6;
    } else {

```

<pre> \$total_pcg_bsolah = round(\$total_dus_bsolah / \$std * 100, 2); } \$dus_recycle = round(\$this->input->post('kg_recycle') / \$this->input->post('kg_dus_cream'), 2); \$dus_recycle_pcg = round(\$dus_recycle / \$std * 100, 2); \$dus_waste = round(\$total_waste_bandrol + \$dus_recycle, 2); if (\$proses->tipe_standar == 'packing') { if (\$total_waste_bandrol_pcg < (\$standar->wastebandrol1)) { \$bs_grade = 'B'; \$bs_point = 7; } else if (\$total_waste_bandrol_pcg >= (\$standar->wastebandrol1) && \$total_waste_bandrol_pcg <= (\$standar->wastebandrol2)) { \$bs_grade = 'C'; \$bs_point = 6; } else { \$bs_grade = 'K'; \$bs_point = 5; } if (\$dus_recycle_pcg < \$standar->wasterec) { \$bs_olah_grade = 'B'; \$bs_olah_point = 7; } else if (\$dus_recycle_pcg >= \$standar->wasterec && \$dus_recycle_pcg <= (\$standar->wasterec+0.5)) { \$bs_olah_grade = 'C'; \$bs_olah_point = 6; } else { \$bs_olah_grade = 'K'; \$bs_olah_point = 5; } } else { if (\$total_pcg_bsolah < \$standar->bs_olah1) { \$bs_olah_grade = 'B'; \$bs_olah_point = 7; } else if (\$total_pcg_bsolah >= \$standar->bs_olah1 && \$total_pcg_bsolah <= \$standar->bs_olah2) { \$bs_olah_grade = 'C'; \$bs_olah_point = 6; } else { \$bs_olah_grade = 'K'; \$bs_olah_point = 5; } } } } Kerupuk } else { \$bs_bandroll = \$this->input->post('kg_bandroll1') + (int)\$this->input->post('kg_bandroll2'); \$dus_bandroll = round(\$bs_bandroll * \$this->input->post('pcs_1kgbandroll') / \$this->input->post('pcs_dus'), 2); \$standar_bandroll = \$this->input->post('standar_bandroll'); \$hasil_bs = round(\$dus_bandroll / \$std, 4) * 100; \$kg_bandroll_detail = \$this->input->post('real_dus1') . '+' . (int)\$this->input->post('real_dus2'); \$real_dus_detail = \$this->input->post('kg_bandroll1') . '+' . </pre>	<pre> \$dt_grade = 'K'; \$dt_point = 5; } if (\$pcg_bs <= \$standar->std_bs) { \$bs_grade = 'B'; \$bs_point = 7; } else if (\$pcg_bs >= \$standar->std_bs && \$pcg_bs <= \$standar->std_bs + 0.5) { \$bs_grade = 'C'; \$bs_point = 6; } else { \$bs_grade = 'K'; \$bs_point = 5; } } if (\$this->input->post('produk') == "AGING TIME" OR \$this->input->post('produk') == "PENAMBAHAN MINYAK" or \$this->input->post('produk') == "PERSIAPAN AWAL" or \$this->input->post('produk') == "PEMANASAN MINYAK ULANG" or \$this->input->post('produk') == "TAP MINYAK" or \$this->input->post('produk') == "MIXING MINYAK" or \$this->input->post('produk') == "OFF" or \$this->input->post('produk') == "GANTI PRODUK" or \$this->input->post('produk') == "TRIAL PRODUK" or \$this->input->post('produk') == "WAKTU TUNGGU" OR \$this->input->post('produk') == "OFF (ISTIRAHAT)") { \$field = array('id_proses'=> \$this->input->post('id_proses'), 'tanggal'=> \$this->input->post('tanggal'), 'produk'=> \$this->input->post('produk'), 'mulai'=> \$this->input->post('mulai'), 'seleasai'=> \$this->input->post('seleasai'), 'tipe_jam'=> \$this->input->post('tipe_jam'), 'menit_normatif'=> \$menit_normatif, 'menit_nonnormatif'=> \$menit_nonnormatif, 'dus_normatif'=> \$kg_normatif, 'persen_normatif'=> \$pcg_normatif, 'tanggal_input'=> date('Y-m-d'), 'user_input' => \$this->session->userdata('iduser')); } else { \$field = array('id_proses'=> \$this->input->post('id_proses'), 'tanggal'=> \$this->input->post('tanggal'), 'produk'=> \$this->input->post('produk'), 'mulai'=> \$this->input->post('mulai'), 'seleasai'=> \$this->input->post('seleasai'), 'tipe_jam'=> \$this->input->post('tipe_jam'), 'jumlah_menit'=> \$jml_menit, 'real_dus'=> \$real_dus, 'real_persen_dus'=> \$eff, 'standar_output'=> \$std, 'kg_bandroll' => \$bs_gorengan, 'kg_bandroll_detail'=> \$bs_detail, 'real_bandroll'=> \$pcg_bs, 'menit_normatif'=> \$menit_normatif, 'dus_normatif'=> \$kg_normatif, 'persen_normatif'=> \$pcg_normatif, 'menit_nonnormatif'=> \$menit_nonnormatif, 'dus_nonnormatif' => \$kg_nonnormatif, 'persen_nonnormatif'=> \$pcg_nonnormatif, 'persen_balancing' => \$balancing, 'eff_grade'=> \$eff_grade, </pre>
--	--

<pre> \$this->input->post('kg_bandrol2'); \$dus = \$this->input->post('pcs_dus'); \$total_dus_bsolah = 0; \$dus_waste = \$dus_bandroll; } \$dus_waste_pcg= round(\$dus_waste / \$std * 100, 2); if (\$proses->tipe_standar == 'pengolahan') { \$dus_nonnormatif= round(\$this->input->post('menitnon') * (\$this->input->post('std')/60), 2); \$dus_normatif = round(\$this->input->post('menitnormatif') * (\$this->input->post('std')/60), 2); }else{ \$dus_nonnormatif = round(\$this->input->post('menitnon') * \$this->input->post('speed') / \$dus, 2); \$dus_normatif = round(\$this->input->post('menitnormatif') * \$this->input->post('speed') / \$dus, 2); } if (\$proses->tipe_standar == 'packing') { if (\$hasil >= \$standar->eff22) { \$eff_grade = 'B'; \$eff_point = 7; } else if (\$hasil >= \$standar->eff11 && \$hasil < \$standar->eff22) { \$eff_grade = 'C'; \$eff_point = 6; } else { \$eff_grade = 'K'; \$eff_point = 5; } } else { if (\$hasil >= \$standar->eff2) { \$eff_grade = 'B'; \$eff_point = 7; } else if (\$hasil >= \$standar->eff1 && \$hasil < \$standar->eff2) { \$eff_grade = 'C'; \$eff_point = 6; } else { \$eff_grade = 'K'; \$eff_point = 5; } } if (\$persen_nonnormatif <= 0) { \$dt_grade = 'B'; \$dt_point = 7; } else if (\$persen_nonnormatif <= 15 && \$persen_nonnormatif > 0) { \$dt_grade = 'C'; \$dt_point = 6; } else { \$dt_grade = 'K'; \$dt_point = 5; } if (\$id_bagian == 6 or \$id_bagian == 11) { //gopotato \$field += ['total_ctn'=> \$total_waste_bandrol, 'total_ctn_pcg'=> \$total_waste_bandrol_pcg, 'bs_olah_grade'=> \$bs_olah_grade, 'bs_olah_point'=> \$bs_olah_point, 'moulding'=> \$this->input->post('moulding'), 'oven'=> \$this->input->post('oven'), 'recycle_gaido'=> </pre>	<pre> 'eff_point'=> \$eff_point, 'bs_grade'=> \$bs_grade, 'bs_point'=> \$bs_point, 'dt_grade'=> \$dt_grade, 'dt_point'=> \$dt_point, 'kg_drum' => \$this->input->post('kg_drum'), 'drum_jam' => \$this->input->post('drum_jam'), 'produk_std_id'=> \$this->input->post('id_standar'), 'produk'=> \$this->input->post('produk'), 'tanggal_input'=> date('Y-m-d'), 'user_input'=> \$this->session->userdata('iduser')); } \$this->db->insert('transaksi_balancing', \$field); </pre>
---	--

<pre> \$this->input->post('recycle_gaido'), 'sapuan'=> \$this->input->post('sapuan'), 'total_kg_bsolah'=> \$total_kg_bsolah, 'bs_basah_dus'=> \$bs_basah_dus, 'bs_basah_pcg'=> \$bs_basah_pcg, 'bs_recgaido_dus'=> \$bs_recgaido_dus, 'bs_recgaido_pcg'=> \$bs_recgaido_pcg, 'bs_recpack_dus'=> \$bs_recpack_dus, 'bs_recpack_pcg'=> \$bs_recpack_pcg, 'total_dus_bsolah'=> \$total_dus_bsolah, 'total_pcg_bsolah'=> \$total_pcg_bsolah, 'tampungan_wip'=> \$this->input->post('tampungan_wip'), 'tampungan_wip_pcg' => round(\$this->input->post('tampungan_wip') / \$this->input->post('std') * 100, 2), 'mc_pack'=> \$this->input->post('mc_pack') + \$this->input->post('bs_teknik'), 'mc_kospack'=> \$this->input->post('mc_kospack'), 'kg_recycle'=> \$this->input->post('kg_recycle'), 'dus_recycle'=> \$dus_recycle, 'real_recycle'=> \$dus_recycle_pcg, 'bs_teknik'=> \$this->input->post('bs_teknik'), 'bs_qc'=> \$this->input->post('bs_qc')]; } else { //krupuk \$field += ['kg_bandroll' => \$bs_bandroll, 'dus_bandroll'=> \$dus_bandroll, 'standar_bandroll' => \$standar_bandroll, 'real_bandroll'=> \$hasil_bs, 'kg_bandroll_detail' => \$kg_bandroll_detail, 'real_dus_detail' => \$real_dus_detail]; } } \$this->db->insert('transaksi_balancing', \$field); </pre>	
--	--

3.3 Data Pelaksana

Data pelaksana per-transaksi diproses di controller Balancing pada function-function berikut,

```

public function TambahShift($id)
{
...
}
public function UbahShift()
{
...
}
public function SimpanShift()
{
...
}
public function TransferMpShift()
{
...
}

```

Fungsi masing-masing function sesuai nama functionnya.

<pre> public function TambahShift(\$id) { \$data['Shift'] = \$this->b->getShift(\$id); \$data['Jabatan'] = \$this->b->getJabatan(); \$data['AllJabatan'] = \$this->db->query("SELECT * FROM jabatan order by nama_jabatan")->result(); \$data['Proses'] = \$this->b->getShiftProses(\$id); \$data['Produk'] = \$this->b->getProdukShift(\$id); \$data['AllPelaksana'] = \$this->db->query("SELECT * FROM pelaksana p </pre>	<pre> public function SimpanShift() { \$result=\$this->b->getSimpanShift(); if (\$result) { } else { } redirect(\$_SERVER['HTTP_REFERER']); } </pre>
--	--

```

JOIN jabatan j on p.id_jabatan=j.id_jabatan order by nama")->result();
$this->load->view('template/header');
$this->load->view($this->menu);
$this->load->view("Balancing/TambahShift", $data);
$this->load->view('template/footer');
}

```

Untuk view TambahShift.php berikut preview sintaknya,

```

<div class="col-md-4">
<!-- general form elements -->
<div class="box box-primary">
<div class="box-header with-border">
<h3 class="box-title">Tambah Data Shift</h3>
</div>
<!-- /.box-header -->
<!-- form start -->
<form role="form" name="form" method="post" action="{<php echo base_url('Balancing/Simpan
>
<div class="box-body"> ...
</div>
<!-- /.box-body -->
<div class="box-footer">
<button type="submit" class="btn btn-primary">Simpan</button>
<button type="reset" class="btn btn-danger">Batal</button>
<button type="button" class="btn btn-success" id="transfer">Transfer MP</button>
</div>
</form>
</div>
<!-- /.box -->

</div>
<div class="col-md-8">
<div class="box">
<div class="box-header row">
<div class="col-md-2"> ...
</div>
<div class="col-md-4"> ...
</div>
<div class="col-md-6"> ...
</div>
</div>
<!-- /.box-header -->
<div class="box-body">
<div style="overflow-x: auto;">
<table id="example1" class="table table-bordered table-striped"> ...
</table>
</div>
</div>
<!-- /.box-body -->

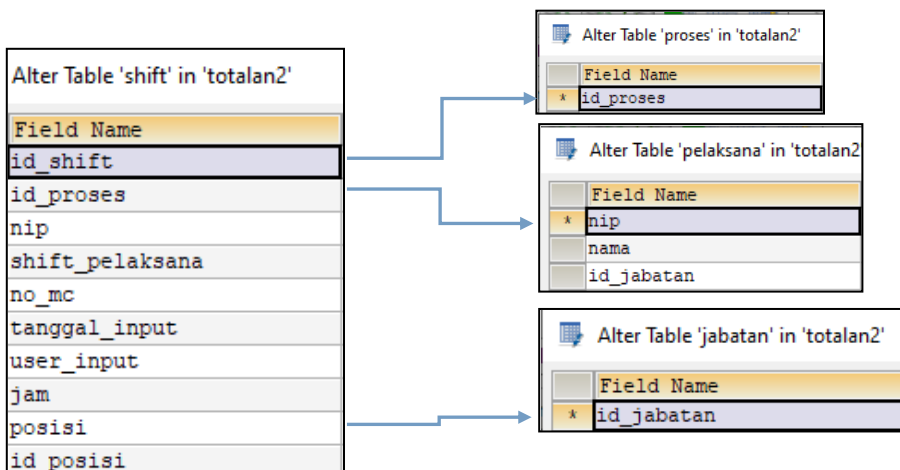
```

Form input untuk menambahkan data pelaksana, jika ingin mentransfer data pelaksana dari transaksi sebelumnya dapat menggunakan fungsi button transfer MP

Tampilan data header transaksi yang sedang ditambahkan data pelaksananya

Tampilan list data pelaksana yang sudah diinputkan, user dapat mengubah atau menghapus list data tsb

Data pelaksana per-transaksi ini diproses di table shift yang berelasi dengan table-table lain, berikut gambaran relasinya,



id_posisi berelasi dengan table jabatan. Jika nip pelaksana sedang tidak menjalankan jabatan sebenarnya (pengganti)

3.4 Data Pendukung Kesimpulan Akhir Shift

Data pendukung kesimpulan akhir shift di proses di controller Sheets pada function updateDataProses. Proses ini tidak menggunakan table baru, kolom sudah ada pada table proses dengan default null, ketika data pendukung diisi, maka akan mengupdate data pada table proses. Berikut preview source codenya,

```
public function updateDataProses()
{
    $d = $this->input->post('submit_gopotato');
    $field = array(
        'line' => $this->input->post('line'),
        'penambahan_recycle' => $this->input->post('penambahan_recycle'),
        'stock_awal' => $this->input->post('stock_awal'),
        'stock_akhir' => $this->input->post('stock_akhir'),
        'afterpacking' => $this->input->post('afterpacking'),
        'prapacking' => $this->input->post('prapacking'),
        'bs_basah_akhir' => $this->input->post('bs_basah_akhir'),
        'bs_gaido_akhir' => $this->input->post('bs_gaido_akhir'),
        'bs_recpack' => $this->input->post('bs_recpack'),
        'bs_sapuan' => $this->input->post('bs_sapuan'),
        'bs_bumbu' => $this->input->post('bs_bumbu'), //update
        'plan_ppic' => $this->input->post('plan_ppic'),
        'plan_jk' => $this->input->post('plan_jk'),
        'jml_adukan' => $this->input->post('jml_adukan'),
        'transfer' => $this->input->post('transfer'),
        'transfer_ket' => $this->input->post('transfer_ket'),
        'wip' => $this->input->post('wip'),
        'wip_lain' => $this->input->post('wip_lain'),
        'hasil_jadi' => $this->input->post('hasil_jadi'), //update
        'hasil_olah' => $this->input->post('hasil_olah'), //update
        'jml_mc_packing' => $this->input->post('jml_mc_packing'), //update
        'penambahan_pracospack' => $this->input->post('real'),
        'penambahan_12jadi' => $this->input->post('penambahan_12jadi'), //tambahan
        'sisamaterial_12jadi' => $this->input->post('sisamaterial_12jadi'), //tambahan
    );
    $this->db->where('id_proses', $this->input->post('id_proses'));
    $this->db->update('proses', $field);
    redirect($_SERVER['HTTP_REFERER']);
}
```

3.5 Data Report

Data raport di proses di controller Sheets pada function simpanRaport, berikut preview source codenya,

```
public function simpanRaport($bagian = null)
{
    $id = $this->input->post('id_proses');
    if ($this->input->post('poh') == 'B') {
        $poh_grade = 'B';
        $poh_point = 7;
    } else if ($this->input->post('poh') == 'K') {
        $poh_grade = 'K';
    }
}
```

```

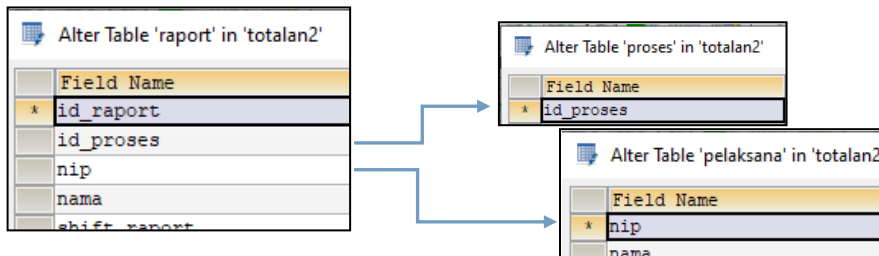
        $poh_point = 5;
    } else {
        $poh_grade = null;
        $poh_point = null;
    }
    if ($this->input->post('yield') == 'B') {
        $yield_grade = 'B';
        $yield_point = 7;
    } else if ($this->input->post('yield') == 'K') {
        $yield_grade = 'K';
        $yield_point = 5;
    } else {
        $yield_grade = null;
        $yield_point = null;
    }
}
$sql = "SELECT
b.id_bagian,p.id_packing,p.plan_ppic,t.tipe_jam,p.shift,j.jenis_packing,b.nama_bagian,p.tanggal AS
tanggal_proses,t.* FROM proses p
JOIN jenis j ON j.id_packing=p.id_packing
LEFT JOIN bagian b ON b.id_bagian=j.id_bagian
RIGHT JOIN transaksi_balancing t ON t.`id_proses`=p.`id_proses`
WHERE p.id_proses='$id'
ORDER BY p.id_proses DESC";
$query = $this->db->query($sql);
$trxs = $query->result();
$field=[];
if ($trxs) {
    //update hapus dulu raport
    $this->db->where('id_proses', $this->input->post('id_proses'));
    $this->db->delete('raport');

    foreach ($trxs as $i => $trx) {
        $idtrx = $trx->id_transaksi;
        // var_dump($trx->id_transaksi);
        $sql = "SELECT * FROM shift p JOIN pelaksana pe ON pe.nip=p.nip WHERE p.id_proses='$id'";
        $query = $this->db->query($sql);
        $staff = $query->result();

        if ($staff) {
            $jam1 = '';
            $jam2 = '';
            $st_jam = '';
            foreach ($staff as $i => $d) {
.....DST

```

Proses ini menginsert data ke table raport yang berelasi ketable proses untuk informasi detail transaksinya, dan table pelaksana untuk detail pelaksana tsb.



sedangkan untuk viewnya diproses di controller Laporan pada function raport dan view raport, berikut source codenya,

```
public function raport()
{
    $data = array( "konten" => "Raport");
    $id = $this->input->get('id_bagian');
    if ($this->input->get('bln')) {
        $bulan = date('m', strtotime($this->input->get('bln') . '-01'));
        $tahun = date('Y', strtotime($this->input->get('bln') . '-01'));
        $param = date('F Y', strtotime($this->input->get('bln') . '-01'));
    } else {
        $bulan = date('m');
        $tahun = date('Y');
        $param = date('F Y');
    }
    $data['thb'] = $tahun.'-'. $bulan;
    $data['Bagian'] = $this->db->query("SELECT * FROM bagian WHERE id_bagian='$id'")->row();
    $data['Allbagian'] = $this->db->query("SELECT * FROM bagian")->result();
    $data['Jenis'] = $this->db->query("SELECT * FROM jenis WHERE id_bagian='$id'")->result();
    $data['RecordKategori'] = $this->db->query("SELECT * FROM record_kategori WHERE id_bagian='$id' order
by kategori")->result();
    if ($data['Bagian']) {
        $bag = $data['Bagian']->nama_bagian;
        $data['param'] = $bag . ' ' . $param;
        $data['Asdiv'] = $this->getStaff($bulan, $tahun, $bag, '2,22');
        $data['Staff'] = $this->getStaff($bulan, $tahun, $bag, '10,9,4');
        $data['Staff2'] = $this->getStaff($bulan, $tahun, $bag, '3');
        $data['Karu'] = $this->getStaff($bulan, $tahun, $bag, '5,7,11,12,13,14,16,17,18');
        $data['Teknik'] = $this->getStaff($bulan, $tahun, $bag, '6,20,28,29,30,32,31,39,21,25,26,27');
        $data['raportAsdiv'] = $this->getRaport($bulan, $tahun, $bag, '2,22');
        $data['raportStaff'] = $this->getRaport($bulan, $tahun, $bag, '10,9,4');
        $data['raportStaff2'] = $this->getRaport($bulan, $tahun, $bag, '3');
        $data['raportKaru'] = $this->getRaport($bulan, $tahun, $bag, '5,7,11,12,13,14,16,17,18,39');
        $data['raportTeknik'] = $this->getRaport($bulan, $tahun, $bag, '6,20,28,29,30,32,31,39,21,25,26,27');
        $data['TOInsentifAsdiv'] = $this->getTOInsentif($bulan, $tahun, $bag, '2,22');
        $data['TOInsentifStaff'] = $this->getTOInsentif($bulan, $tahun, $bag, '10,9,4');
        $data['TOInsentifStaff2'] = $this->getTOInsentif($bulan, $tahun, $bag, '3');
        $data['TOInsentifKaru'] = $this->getTOInsentif($bulan, $tahun, $bag, '5,7,11,12,13,14,16,17,18');
        $data['TOInsentifTeknik'] = $this->getTOInsentif($bulan, $tahun, $bag,
'6,20,28,29,30,32,31,39,21,25,26,27');
    }else{
        $bag = '';
        $data['param'] = $bag . ' ' . $param;
        $data['Asdiv'] = [];
        $data['Staff'] = [];
        $data['Staff2'] = [];
        $data['Karu'] = [];
        $data['Teknik'] = [];
        $data['raportAsdiv'] = [];
        $data['raportStaff'] = [];
        $data['raportStaff2'] = [];
        $data['raportKaru'] = [];
        $data['raportTeknik'] = [];
        $data['TOInsentifAsdiv'] = [];
        $data['TOInsentifStaff'] = [];
        $data['TOInsentifStaff2'] = [];
        $data['TOInsentifKaru'] = [];
        $data['TOInsentifTeknik'] = [];
    }
    $this->load->view('template/header');
    $this->load->view($this->menu);
    $this->load->view("Laporan/raport", $data);
    $this->load->view('template/footer');
}
```

Function report ini memanggil function lain, berikut function-functionnya,

```
public function getReport($bln, $th, $bag, $jab)
{
    $sql = "SELECT t2.tipe_standar,t2.selected_std,t4.std_extruder,t4.yield as yield_extru, t4.poh as poh_extru,t4.eff as eff_e
    $query = $this->db->query($sql);
    return $query->result();
}

public function getTOInsentif($bln, $th, $bag, $jab)
{
    ...
}

public function getTOInsentifNIP($nip, $bag, $bln, $th){...
```

Source code View report cukup kompleks dan menggunakan data yang banyak, berikut sample code view report.php

```
report.php > report
<!-- data-toggle="tab" href="#menu22"Karu--></li>
<!-- data-toggle="tab" href="#menu33"Staff Teknik--></li>
<!-- data-toggle="tab" href="#menu22"Teknik--></li>
<!-- data-toggle="tab" href="#menu4"Teknik Packing--></li>
</ul>
<div class="tab-content">
<div id="home2" class="tab-pane fade in active">
<h4>RAPORT FUNGSI LEVEL AST DIV</h4>
<?php
    if ($raportAstdiv) {
        raport($raportAstdiv, $Astdiv, $Bagian, $TOInsentifAstdiv);
    }
    >
</div>
<div id="menu22" class="tab-pane fade">
<h4>RAPORT FUNGSI LEVEL STAFF PRODUKSI</h4>
<?php
    if ($raportStaff) {
        raport($raportStaff, $Staff, $Bagian, $TOInsentifStaff);
    }
    >
</div>
<div id="menu22" class="tab-pane fade">
<h4>RAPORT FUNGSI LEVEL KARU</h4>
<?php
    if ($raportKaru) {
        raport($raportKaru, $Karu, $Bagian, $TOInsentifKaru);
    }
    >
</div>
<div id="menu33" class="tab-pane fade">
<h4>RAPORT FUNGSI LEVEL STAFF TEKNIK</h4>
<?php
    if ($raportStaff2) {
        raport($raportStaff2, $Staff2, $Bagian, $TOInsentifStaff2);
    }
    >
</div>
<div id="menu32" class="tab-pane fade">
<h4>RAPORT FUNGSI LEVEL TEKNIK</h4>
<?php
    if ($raportTeknik) {
        raport($raportTeknik, $Teknik, $Bagian, $TOInsentifTeknik);
    }
    >
</div>
</div>
```

Nilai report ditampilkan Per-bagian dengan format view report yang sama, untuk memudahkan development, view per bagian ini dikembangkan dengan memanggil function.

```
function raport($raport, $level, $Bagian, $TOInsentif)
{
    if (isset($_GET['bin'])) {
        $parambin = $_GET['bin'];
    } else {
        $parambin = date('Y-m');
    }
    if ($_GET['id_bagian'] == 5 or $_GET['id_bagian'] == 6 or $_GET['id_bagian'] == 7) {
        $kel_bagian = 'BISOLIT';
    } else if ($_GET['id_bagian'] == 1 or $_GET['id_bagian'] == 2) {
        $kel_bagian = 'MIE';
    } else {
        $kel_bagian = 'KERUPUK';
    }

    $titlPlan = 0;
    $titlMelissal = 0;
    $titlKapHE = 0;
    $titlMSolah = 0;
    $titlMSAndrol = 0;
    $titlOP = 0;
    $titlOutput = 0;
    $titlInput = 0;
    $titlPH = 0;
    $titlIFF = 0;
    $titlISO = 0;
    $titlRS = 0;
    $titlYield = 0;
    $titlRS = 0;
    $titlS = 0;

    foreach ($level as $i => $staff) {
        // return $view;
    }
    >
<div class="content-wrapper" style="@media (min-width: 767px){margin-left: 180px;}"
```

BAB IV REPORT

4.1 Monitoring Progress

Data resume progress di proses di controller Laporan pada function progress. Monitoring progress ini merupakan data resume dari transaksi balancing, tidak ada table khusus untuk menyimpan datanya, data diproses atau dikelola dengan query yang menarik data dari table proses, transaksi_balancing dan table-table standar. Beberapa proses bagian menggunakan table standar yang berbeda oleh karena itu query ini juga diproses sesuai bagian yang ingin di view, berikut preview source codenya,

```
$tambahansql='';
if( $this->input->get('produk')){ ...
}

//(SUM(real_dus)+IFNULL(p.penambahan_pracospack,0))
$sql = "SELECT p.tipe_standar,t.id_proses,p.tanggal,p.shift,j.jenis_packing, p.i
$sql = "SELECT p.line,p.tipe_standar,t.id_proses,p.tanggal,p.shift,j.jenis_packi
if ($id == 8) { ...
} else if ($id == 10) { ...
} else if ($id == '9') { ...
}
}
$data['progress'] = $this->db->query($sql)->result();
$sql = "SELECT tipe_standar, jenis_packing, id_packing,
IF (IFNULL(plan_ppic,0)=0,0,ROUND(AVG(DUS/plan_ppic*100),2)) AS poh, ...
if ($id == '8') { ...
} else if ($id==10){ ...
} else if ($id == '9') { ...
}

$data['recordAVG'] = $this->db->query($sql)->result();

$this->load->view('template/header');
$this->load->view($this->menu);
$this->load->view("Laporan/progress", $data);
$this->load->view('template/footer');
```

Untuk memproses jika ingin menampilkan data per item produk

Untuk memproses data nilai progress pertransaksi

Untuk memproses data nilai progress rata-rata peritem

Sama dengan controllernya, pada viewnya juga dibedakan menjadi beberapa bagian, karena masing-masing bagian memiliki tampilan data monitoring progress yang berbeda, berikut review source code view progress.php

```
<div class="col-md-12">
<div style="overflow-x:auto;" id="html-content-holder">
<?php if ($GET['id_bagian'] == 7 OR $GET['id_bagian']==11) { ?>
<!-- =====GORIOTID ISI 4 NONINLINE===== -->
<?php
if ($progress) {
$standar_poh = $progress[0]->poh1;
$standar_eff = $progress[0]->eff2;
} ?>
<table class="table table-bordered table-striped table-hover" id="example1"
</table>
<table class="table table-bordered table-striped table-hover" id="example1"
</table>
<?php } else if ($GET['id_bagian'] == 8) { ?>
<!-- =====GORENGAN===== -->
<table class="table table-bordered table-striped table-hover" style="font-s
</table>
<?php } else if ($GET['id_bagian'] == 9) { ?>
<!-- ===== PUFF TWIST===== -->
<table class="table table-bordered table-striped table-hover" style="font-s
</table>
<?php } else if ($GET['id_bagian'] == 10) { ?>
<!-- ===== PENGOLAHAN RIE HANCUR ===== -->
<table class="table table-bordered table-striped table-hover" style="font-s
</table>
<?php } else { ?>
<table class="table table-bordered table-striped table-hover" style="font-s
</table>
<?php } ?>
</div>
</div>
```

Untuk progress non inline, progress ditampilkan ke 2 table, pengolahan dan packing.

Untuk tampilan PDF, di proses di controller Laporan function progress_pdf(), data yang ditampilkan sama dengan yang di view sehingga query yang digunakan juga sama, jika developer melakukan update pada view, maka perlu juga mengupdate pada pdfnya, karena view dan pdf ada pada function berbeda. Library yang digunakan untuk menggenerate PDF pada controller ini yaitu TCPDF, berikut preview source codenya,

```
public function progress_pdf()
{
    $data = [];
    // $this->load->view("Laporan/bsbandrol_pdf", $data);
    // $pdf = new TCPDF(PDF_PAGE_ORIENTATION, PDF_UNIT, PDF_PAGE_FORMAT, true,
    if($this->input->get('id_bagian')==1){
        $pdf = new TCPDF('L', PDF_UNIT, 'A4', true, 'UTF-8', false);
    }else{
        $pdf = new TCPDF(PDF_PAGE_ORIENTATION, PDF_UNIT, 'A4', true, 'UTF-8', fa
    }

    // set document information
    $pdf->SetCreator(PDF_CREATOR);
    $pdf->SetAuthor('Nicola Asuni');
    $pdf->SetTitle('TOTALAN');
    $pdf->SetSubject('STT BEKASI');
    $pdf->SetKeywords('TCPDF, PDF, example, test, guide');
    $pdf->SetHeaderData("", PDF_HEADER_LOGO_WIDTH, "TOTALAN", "STTB");
    // $pdf->SetHeaderData(PDF_HEADER_LOGO, PDF_HEADER_LOGO_WIDTH, PDF_HEADER_T
    $pdf->setHeaderFont(array(PDF_FONT_NAME_MAIN, '', PDF_FONT_SIZE_MAIN));
    $pdf->setFooterFont(array(PDF_FONT_NAME_DATA, '', PDF_FONT_SIZE_DATA));
    $pdf->SetDefaultMonospacedFont(PDF_FONT_MONOSPACED);
    $pdf->SetMargins(10, 20, 10);
    // $pdf->SetMargins(PDF_MARGIN_LEFT, PDF_MARGIN_TOP, PDF_MARGIN_RIGHT);
    $pdf->SetHeaderMargin(PDF_MARGIN_HEADER);
    $pdf->SetFooterMargin(PDF_MARGIN_FOOTER);
    $pdf->SetAutoPageBreak(TRUE, PDF_MARGIN_BOTTOM);
    $pdf->setImageScale(PDF_IMAGE_SCALE_RATIO);
    if (@file_exists(dirname(__FILE__) . '/lang/eng.php')) {
        require_once(dirname(__FILE__) . '/lang/eng.php');
        $pdf->setLanguageArray($l);
    }
    $pdf->SetFont('helvetica', '', 10);
    // add a page
    //=====TABLE=====
    $id = $this->input->get('id_bagian');
    if ($this->input->get('bln')) {
        $bulan = date('m', strtotime($this->input->get('bln') . '-01'));
        $tahun = date('Y', strtotime($this->input->get('bln') . '-01'));
        $param = date('F Y', strtotime($this->input->get('bln') . '-01'));
    } else {
        $bulan = date('m');
        $tahun = date('Y');
        $param = date('F Y');
    }
    $paramBulan = $tahun.'-'.$bulan.'-01';

    $Bagian = $this->db->query("SELECT * FROM bagian WHERE id_bagian='$id'")->ro
    $pdf->AddPage();
    $tambahansql='';
    if($this->input->get('produk')){
        $produk = $this->input->get('produk');
        if ($produk!='all') {
            $tambahansql="AND j.id_packing='$produk'";
        }
    }

    $sql = "SELECT p.tipe_standar,p.line,t.id_proses,p.tanggal,p.shift,j.jenis_p
    if ($id == 8) { ...
    } else if ($id == 10) { ...
    } else if ($id == 9) { ...
    }
    $progress = $this->db->query($sql)->result();
    $sql = "SELECT jenis_packing, ...
    $recordAWG = $this->db->query($sql)->result();

    $html = ' ...
    if ($id == 8) { ...
    } else if ($id == 9) { ...
    } else if ($id == 10) { ...
    } else if ($id == 7 or $id==11) { ...
    } else { ...
    }
    $pdf->writeHTML($html, true, false, true, false, '');
    $pdf->lastPage();
    $pdf->Output('Monitoring Progress ' . $param . '-' . $Bagian->nama_bagian .
```

Settingan PDF, Function-funtion
dari library TCPDF

Source code query data

Data pada menu ini hasil pemrosesan data pada table transaksi_balancing dan table proses. Relasi table sudah digambarkan pada point 3.1


```

<div class="modal fade" id="modalNilaiKasusMutu" tabindex="-1" role="dialog">
  <div class="modal-dialog modal-lg">
    <form role="form" name="form" method="post" action="{?php echo base_url('Laporan/insentif_simpanNilaiKasusMutu')}";>
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal">&times;</button>
          <h4>Input Nilai Kasus Mutu</h4>
        </div>
        <div class="modal-body">
          <div class="box-body">...
        </div>
        <div class="modal-footer">...
        </div>
      </div>
    </form>
  </div>
</div>

<div class="modal fade" id="modalNilaiFKP" tabindex="-1" role="dialog">
  <div class="modal-dialog modal-lg">
    <form role="form" name="form" method="post" action="{?php echo base_url('Laporan/insentif_simpanNilaiFKP')}";>
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal">&times;</button>
          <h4>Input point FKP</h4>
        </div>
        <div class="modal-body">
          <div class="box-body">...
        </div>
        <div class="modal-footer">...
        </div>
      </div>
    </form>
  </div>
</div>

<div class="modal fade" id="modalProsesPremi" tabindex="-1" role="dialog">
  <div class="modal-dialog modal-lg">
    <form role="form" name="form" method="post" action="{?php echo base_url('Laporan/insentif_simpan')}";>
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal">&times;</button>
          <h4>Proses Premi</h4>
        </div>
        <div class="modal-body">...
        </div>
        <div class="modal-footer">...
        </div>
      </div>
    </form>
  </div>
</div>

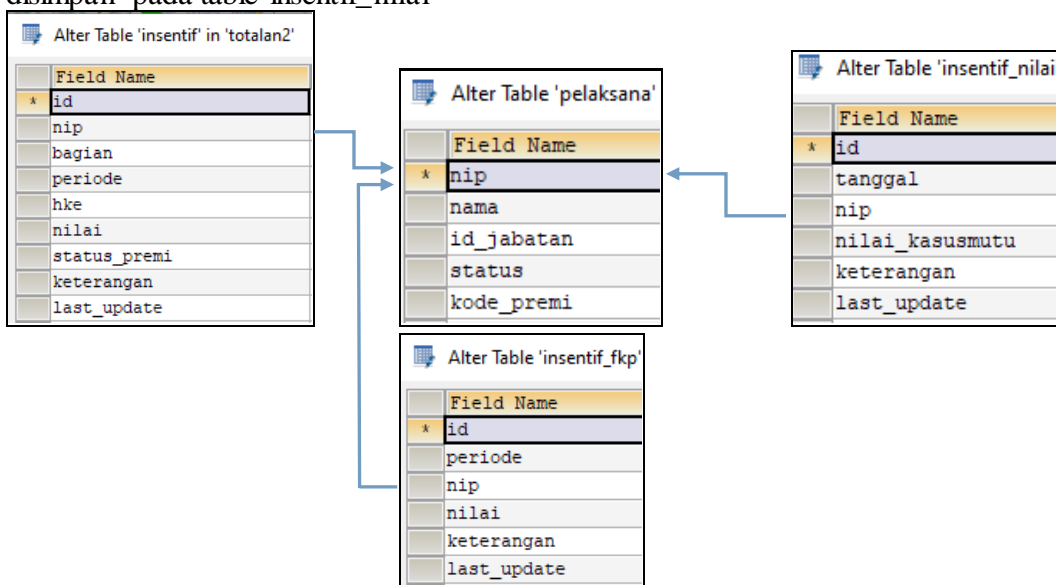
```

Form input untuk memproses data pendukung perhitungan nilai summary raport, yaitu Nilai kasus mutu. Form diproses di controller Laporan function insentif_simpanNilaiKasusMutu

Form input untuk memproses data pendukung perhitungan nilai summary raport, yaitu Point FKP. Form diproses di controller Laporan function insentif_simpanNilaiFKP

Form input untuk menyimpan data perhitungan nilai summary raport. Form diproses di controller Laporan function insentif_simpan

Data pada menu ini hasil pemrosesan data pada table raport. Data pendukung seperti jumlah HKE dan hasil pemrosesan nilai raport disimpan pada table insentif, sedangkan untuk nilai mutu disimpan pada table insentif_nilai



4.3 Monitoring BS Bandroll

Menu BS Bandroll mengolah data summary BS pertransaksinya dan dibandingkan dengan hasil input user yang melakukan validasi ke lapangan. Data ini diproses di controller Laporan pada function bsbandroll, function ini menampilkan view bsbandroll.php, berikut preview source code function bsbandroll,

```
public function bsbandroll()
{
    $data = array(
        "konten" => "bsbandroll"
    );
    $id = $this->input->get('id_bagian');
    if ($this->input->get('bin')) { ...
    } else { ...
    }
    $data['Bagian'] = $this->db->query("SELECT * FROM bagian WHERE id_bagian='$id'")->row();
    if ($SESSION['leveluser'] == 'Manager') {
        $data['Allbagian'] = $this->db->query("SELECT * FROM bagian where id_bagian not in (8,9,10)")->result();
    } else {
        $data['Allbagian'] = $this->db->query("SELECT * FROM bagian where id_bagian not in (8,9,10)")->result();
    }
    $bag = $data['Bagian']->nama_bagian;
    $data['Jenis'] = $this->db->query("SELECT * FROM jenis WHERE id_bagian='$id'")->result();
    $data['Recordkategori'] = $this->db->query("SELECT * FROM record_kategori WHERE id_bagian='$id' order by kategori");
    $data['param'] = $bag . ' ' . $param;

    if ($GET['id_bagian'] == 5 or $GET['id_bagian'] == 6 or $GET['id_bagian'] == 7 or $GET['id_bagian'] == 11) {
        if ($this->input->get('produk') and $this->input->get('produk') != 'all') { ...
        } else { ...
        }
        $data['bsbandroll'] = $this->db->query($sql)->result();
    } else { ...
    }

    $data['record_produk'] = $this->db->query("SELECT DISTINCT j.id_packing,j.jenis_packing FROM record_r
    inner join proses p ON r.id_proses=p.id_proses -- and p.is_deleted=0
    LEFT JOIN jenis j ON j.id_packing=p.id_packing
    WHERE r.id_bagian=$id AND MONTH(p.tanggal)='$bulan' AND YEAR(p.tanggal)='$tahun'
    order by j.jenis_packing")->result();

    $this->load->view('template/header');
    $this->load->view($this->xmenu);
    $this->load->view('Laporan/bsbandroll', $data);
    $this->load->view('template/footer');
}
```

Data diproses sesuai bagiannya, untuk beberapa bagian data BSnya lebih dari 1 jenis

Sedangkan untuk preview source code view bsbandroll.php sebagai berikut,

```
<?php
<form action="" method="GET">
    <input name="id_bagian" value="" type="text" value="" id="id_bagian">
    <input name="bin" value="" type="text" value="" id="bin">
    <select name="produk">
        <option value="">
        <option value="">
    </select>
    <button type="submit" class="btn btn-sm btn-success">lihat hanya produk ini</button>
    <a target="_blank" href="">
    </a>
</form>

<!-- <a class="btn btn-primary btn-sm" href="">
    </a>
<table class="table table-bordered table-striped table-hover" style="font-size: small;" id="example1">
    <thead>
        <tr>
            <th colspan="10" style="text-align: center; background-color: #87CEEB;">Monitoring BS Bandroll</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>
                <?php if ($GET['id_bagian'] == 5 or $GET['id_bagian'] == 6 or $GET['id_bagian'] == 7 or $GET['id_bagian'] == 11) {
                <?php } else { ?>...
                <?php } ?>
            </td>
        </tr>
    </tbody>
</table>

<div class="modal fade" id="modalVerifBS" tabindex="1" role="dialog">
    <div class="modal-dialog modal-lg">
        <div role="form" name="form" method="post" action="">
            <div class="modal-content">
                <div class="modal-header">
                    <button type="button" class="close" data-dismiss="modal">&times;</button>
                    <h4>Update data BS Bandroll</h4>
                </div>
                <div class="modal-body">
                    <div>
                        <div>
                            <div class="modal-footer">
                                <button type="submit" class="btn btn-success btn-block">Simpan</button>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

Form untuk memilih data bsbandroll dari bagian apa yang akan di tampilkan

Table view data bs bandroll

Form input untuk menginput data BS Bandroll hasil validasi lapangan


```

<?php
if (isset($_GET['bln'])) {
    $b = $_GET['bln'];
} else {
    $b = date('Y-m');
}
?>
<ul class="nav nav-tabs">
<li class="active"><a data-toggle="tab" href="#menu1">POH</a></li>
<li><a data-toggle="tab" href="#menu2">EFF</a></li>
<li><a data-toggle="tab" href="#menu3">YIELD</a></li>
<li><a data-toggle="tab" href="#menu4">All Data</a></li>
</ul>
<div class="tab-content">
<div id="menu1" class="tab-pane fade in active">...
</div>
<div id="menu2" class="tab-pane fade in">...
</div>
<div id="menu3" class="tab-pane fade in">...
</div>
<div id="menu4" class="tab-pane fade in">...
</div>
</div>

```

Tampilan data performance perbagian nilai

Hasil submit form input (2) diproses di controller Laporan function performance_bulan, berikut preview source codenya,

```

public function performance_bulan()
{
    $tgl1 = $this->input->post('tanggal1') . '-01';
    $tgl2 = $this->input->post('tanggal2') . '-31';
    $bagian = $this->input->post('bagian');
    $nilai = $this->input->post('nilai');
    $begin = new DateTime($tgl1);
    $end = new DateTime($tgl2);
    $interval = DateInterval::createFromDateString('31 day');
    $period = new DatePeriod($begin, $interval, $end);

    $param = date('F Y', strtotime($this->input->post('tanggal1')));

    //===== POH =====
    $sql = "SELECT p.bagian,p.item,p.std_poh,";
    foreach ($period as $i => $dt) { ...
    }
    $sql.=" FROM performance p";
    foreach ($period as $dt) { ...
    }
    $sql .= " WHERE p.tanggal between '$tgl1' and '$tgl2' $bag
    GROUP BY item,std_poh
    ORDER BY p.id_bagian,bagian,item,std_poh";
    if ($nilai=='POH' || $nilai=='ALL') {
        $POH = $this->db->query($sql)->result();
    }

    if($POH){ ...
    }

    //===== POH =====
    $html.= '<br><br>';
    //===== EFF =====
    $sql = "SELECT p.bagian,p.item,p.std_eff,";
    foreach ($period as $i => $dt) { ...
    }
    $sql.=" FROM performance p";
    foreach ($period as $dt) { ...
    }
    $sql .= " WHERE p.tanggal between '$tgl1' and '$tgl2' $bag
    GROUP BY item,std_eff
    ORDER BY p.id_bagian,bagian,item,std_eff";
    if ($nilai=='EFF' || $nilai=='ALL') {
        $EFF = $this->db->query($sql)->result();
    }

    if($EFF){ ...
    }

    //===== EFF =====
}

```

Generate period perbulan dari range bulan yang diinput

Memproses summary nilai POH per bulannya

Memproses summary nilai EFF per bulannya

```

//===== BS =====
$sql = "SELECT p.bagian,p.item,p.std_bsbandroll,";
foreach ($period as $i => $dt) { ...
}
$sql.=" FROM performance p";
foreach ($period as $dt) { ...
}
$sql .= " WHERE p.tanggal between '$tgl1' and '$tgl2' $bag
GROUP BY item,std_bsbandroll
ORDER BY p.id_bagian,bagian,item,std_bsbandroll";
$sql2 = "SELECT p.bagian,p.item,p.std_bsroast,";
foreach ($period as $i => $dt) { ...
}
$sql2.=" FROM performance p";
foreach ($period as $dt) { ...
}
$sql2 .= " WHERE p.tanggal between '$tgl1' and '$tgl2' $bag
GROUP BY item,std_bsroast
ORDER BY p.id_bagian,bagian,item,std_bsroast";
if ($nilai=='BS' || $nilai=='ALL') {
    $BS = $this->db->query($sql)->result();
    $BS2 = $this->db->query($sql2)->result();
}
if ($bagian=='BISKUIT') { ...
} if ($bagian=='PUFFTWIST'){ ...
} else { ...
}

//===== BS =====
$html.= '<br><br>';
//===== yield =====
$sql = "SELECT p.bagian,p.item,p.std_yield1,p.std_yield2,";
foreach ($period as $i => $dt) { ...
}
$sql.=" FROM performance p";
foreach ($period as $dt) { ...
}
$sql .= " WHERE p.tanggal between '$tgl1' and '$tgl2' $bag
GROUP BY item,std_yield1,std_yield2
ORDER BY p.id_bagian,bagian,item,std_yield1,std_yield2";
if ($nilai=='YIELD' || $nilai=='ALL') {
    $YIELD = $this->db->query($sql)->result();
}

if($YIELD){ ...
}

//===== yield =====

```

Memproses summary nilai BS per bulannya

Untuk nilai BS pada beberapa bagian, nilai BSnya terbagi 2, oleh karena itu ada pemisahan proses nilai BS per bagian

Memproses summary nilai BS per bulannya