

BAB IV

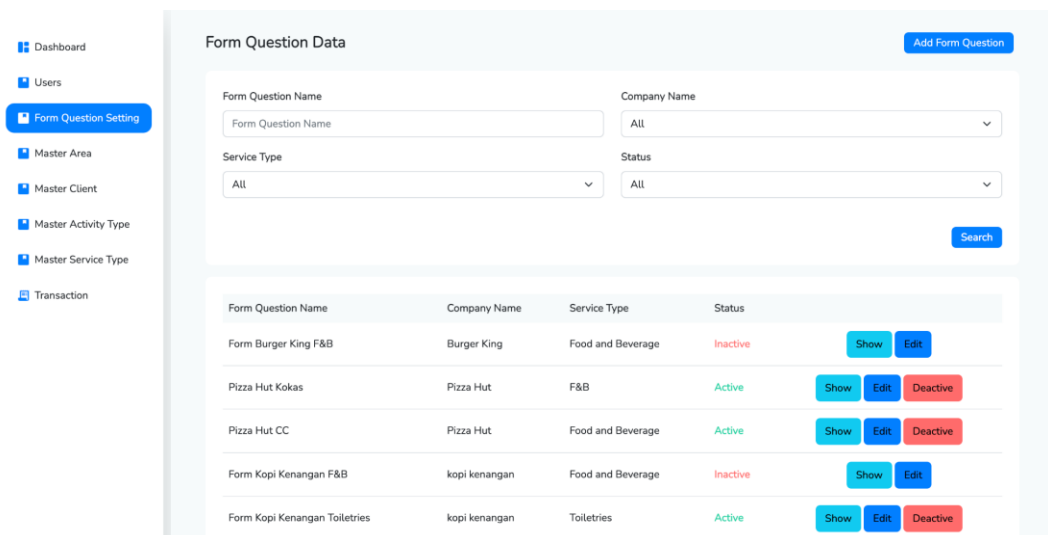
PEMBAHASAN

4.1 Hasil Pengujian

Hasil pengujian akan membahas *interface* dari aplikasi Implementasi *Dynamic Form* Pada aplikasi *Sales Activity Tracking*

4.1.1 Pengujian Menu Form Question Setting

Menu *Form Question Setting* adalah menu yang digunakan oleh admin untuk menambahkan setting pertanyaan atau detail form yang nantinya akan digunakan oleh teknisi saat melakukan aktivitas.



Form Question Name	Company Name	Service Type	Status	
Form Burger King F&B	Burger King	Food and Beverage	Inactive	Show Edit
Pizza Hut Kokas	Pizza Hut	F&B	Active	Show Edit Deactivate
Pizza Hut CC	Pizza Hut	Food and Beverage	Active	Show Edit Deactivate
Form Kopi Kenangan F&B	kopi kenangan	Food and Beverage	Inactive	Show Edit
Form Kopi Kenangan Toiletries	kopi kenangan	Toiletries	Active	Show Edit Deactivate

Gambar 4.1.1 Halaman paging menu *Form Question Setting*

```

public function index(Request $request)
{
    $formQ = RefFormQuestion::has('serviceType')->has('refCoy');
    $serviceType = RefServiceType::where('is_active', '=', '1')->get();
    $coys = RefCoy::all();

    if ($request->form_question_name) {
        $formQ->where('form_question_name', 'like', '%'. $request->form_question_name . '%');
    }
    if ($request->status) {
        $formQ->where('is_active', $request->status);
    }
    if ($request->service_type) {
        $formQ->where('ref_service_type_id', '=', $request->service_type);
    }
    if ($request->ref_coy_id) {
        $formQ->where('ref_coy_id', '=', $request->ref_coy_id);
    }

    return view('pages.form_question.index', [
        'formQ' => $formQ->orderBy('created_at', 'desc')->paginate(5),
        'serviceType' => $serviceType,
        'coy' => $coys
    ]);
}

```

Segmen Program 4.1.1 Segment Program *paging menu form setting*

Halaman awal untuk menu ini dapat dilihat pada gambar 4.1.1, pada halaman ini user bisa melakukan pencarian data *form question* berdasarkan dari nama form, nama *client*, *service type* dan status.

Ketika user menekan *button* “Add Form Question” maka user akan diarahkan ke-halaman *add form question* yang bisa dilihat pada gambar 4.1.2.

Gambar 4.1.2 Halaman add Form Question Setting

```

public function save_header(Request $request)
{
    $serviceType = RefServiceType::where('is_active', '=', '1')->get();
    $coys = RefCoy::all();
    $ref_form = RefFormQuestion::where('ref_service_type_id', $request->service_type)->where('ref_coy_id',$request->ref_coy_id)
->where('is_active', "1")->first();
    if($ref_form != null){
        return view('pages.form_question.add_header', [
            'serviceType' => $serviceType,
            'coy' => $coys,
            'ref_form_question_id' => 0,
            'errors' => "There's already data with same service type and activity type!",
            'service_type' => $request->service_type,
            'ref_coy_id' => $request->ref_coy_id,
            'is_active' => $request->is_active,
            'form_name' => $request->form_name,
        ]);
    }else{
        $ref_form_question = array();
        $ref_form_question['form_question_name'] = $request->form_name;
        $ref_form_question['is_active'] = $request->is_active;
        $ref_form_question['total_form_signer'] = '1';
        $ref_form_question['ref_service_type_id'] = $request->service_type;
        $ref_form_question['ref_coy_id'] = $request->ref_coy_id;
        $form_q = RefFormQuestion::create($ref_form_question);

        $formQ = RefFormQuestion::has('serviceType')->where('id', $form_q->id)->first();
        $coys = RefCoy::where('id', $request->ref_coy_id)->first();

        return view('pages.form_question.add_detail', [
            'serviceType' => $serviceType,
            'form_q' => $formQ,
            'coy' => $coys,
            'errors' => ""
        ]);
    }
}

```

Segmen Program 4.1.2 Segment Program save form question header

Ketika user telah mengisi inputan maka user menekan *button* “Next” dan user akan diarahkan ke-halaman *add form question detail* yang bisa dilihat pada gambar 4.1.3.

Gambar 4.1.3 Halaman add form question setting detail

```

public function save_detail(Request $request)
{
    $a = array();
    $a = $request->header_seq;
    $index_header = array_key_last($a);
    $ref_form = RefFormQuestion::where('id', $request->ref_form_question_id->first());

    for ($x = 0; $x <= $index_header; $x++) {
        if(array_key_exists($x, $request->header_seq)){
            $form_h = array();
            $form_h['ref_form_question_id'] = $request->ref_form_question_id;
            $form_h['question_header'] = $request->header_desc[$x];
            $form_h['seq_no'] = $request->header_seq[$x];
            $form_h['is_active'] = '1';
            $form_h = RefFormQuestionH::create($form_h);

            $form_header = RefFormQuestionH::where('ref_form_question_id', $form_h->ref_form_question_id->where('question_header', $form_h->question_header)
            ->where('seq_no', $form_h->seq_no->where('is_active', $form_h->is_active)->first());

            $b = array();
            $b = $request->detail_question[$x];
            $index_detail = array_key_last($b);
            $seq_no = 1;
            for ($i = 0; $i <= $index_detail; $i++) {
                if(array_key_exists($i, $request->detail_question[$x]){
                    $form_d = array();
                    $form_d['ref_form_question_h_id'] = $form_header->id;
                    $form_d['seq_no'] = $seq_no;
                    $form_d['is_active'] = '1';
                    $form_d['question'] = $request->detail_question[$x][$i];
                    $form_detail = RefFormQuestionD::create($form_d);
                    $seq_no++;
                }
            }
        }
    }

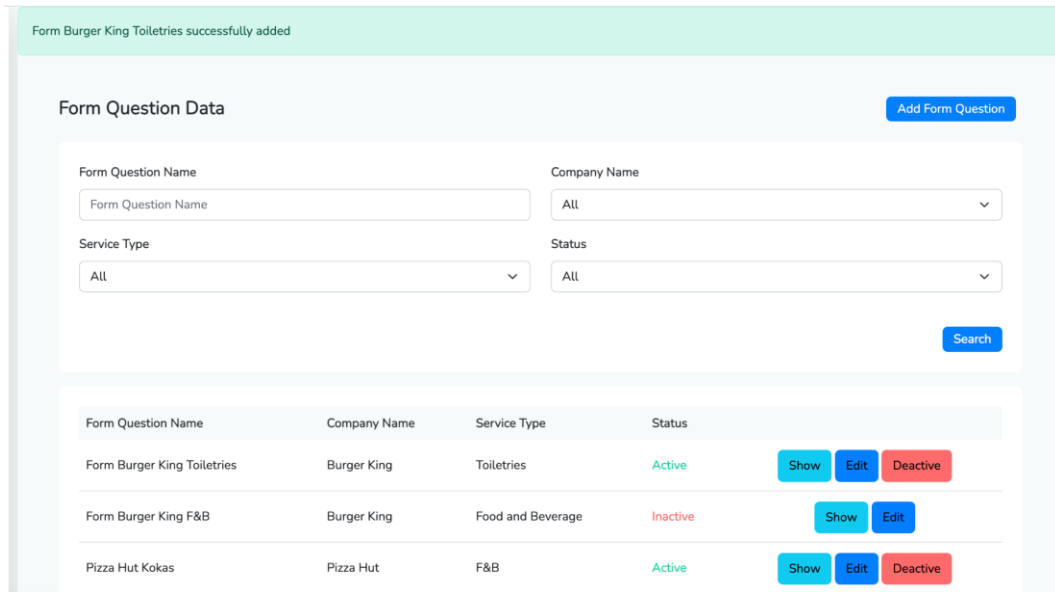
    return redirect('form_question/')->with('message', $ref_form->form_question_name.' successfully added');
}

```

Segmen Program 4.1.3 Segment Program save form question detail

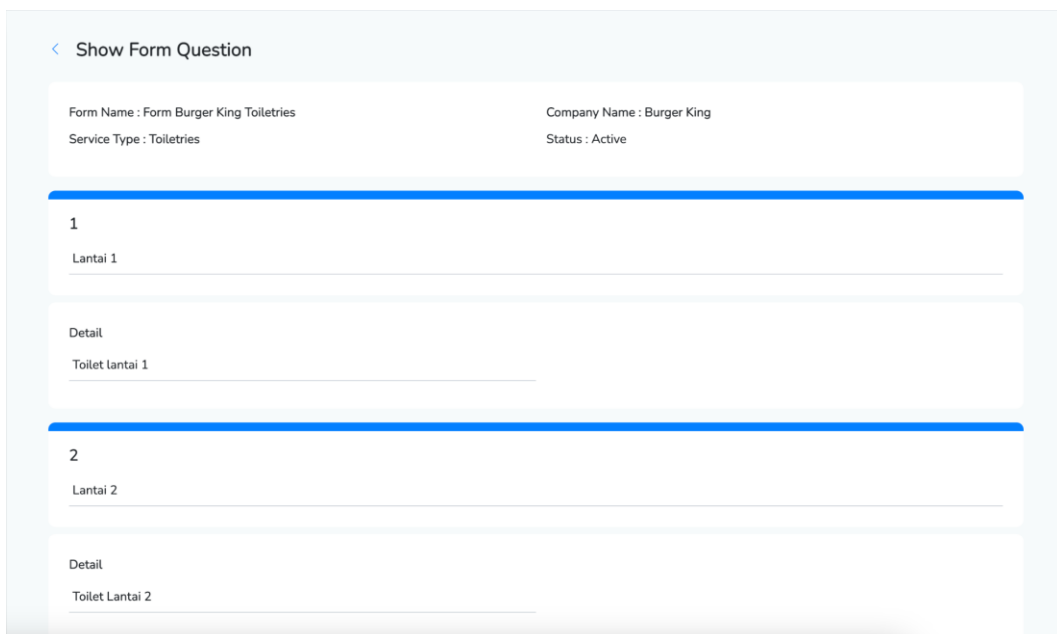
Setelah user menginputkan *detail form* dan menekan *button* “Submit” maka data akan tersimpan dan user akan dibawa menuju halaman paging kembali seperti

pada gambar 4.1.4.



Gambar 4.1.4 Halaman paging setelah menginputkan form question

Ketika *user* ingin melihat data *form question* yang sudah ditambahkan, *user* dapat menekan *button* “*Show*” dan *user* akan ditampilkan data dari *form question* yang dipilih seperti pada gambar 4.1.5.



Gambar 4.1.5 Halaman *show form question*

```

public function show(Request $request, $id)
{
    $formQ = RefFormQuestion::has('serviceType')->where('id', $id)->first();
    $coys = RefCoy::where('id', $formQ->ref_coy_id)->first();
    $formQ['ref_form_question_h'] = RefFormQuestionH::where('ref_form_question_id', $id)->where('is_active', '1')->orderBy('seq_no', 'asc')->get();

    for ($x = 0; $x < count($formQ['ref_form_question_h']); $x++) {
        $formQ['ref_form_question_h'][$x]['ref_form_question_d'] = RefFormQuestionD::where('ref_form_question_h_id', $formQ['ref_form_question_h'][$x]['id'])
            ->where('is_active', '1')->orderBy('seq_no', 'asc')->get();
    }

    return view('pages.form_question.show', [
        'form_q' => $formQ,
        'coy' => $coys
    ]);
}

```

Segmen Program 4.1.4 Segment Program *show Form Question*

Jika user ingin merubah data *form question* maka user perlu menekan *button* “Edit”, dan akan masuk pada halaman *form question* “header” seperti gambar 4.1.6.

Gambar 4.1.6 Halaman *edit form question* “header”

```

public function edit_detail(Request $request)
{
    $serviceType = RefServiceType::where('is_active', '=', '1')->get();
    $coys = RefCoy::all();
    $ref_form = RefFormQuestion::where('ref_service_type_id', $request->service_type)->where('ref_coy_id', $request->ref_coy_id)
        ->where('is_active', '1')->where('id', '=', $request->ref_form_question_id)->first();
    if($ref_form != null && $request->is_active == '1'){
        return view('pages.form_question.edit_header', [
            'serviceType' => $serviceType,
            'coy' => $coys,
            'form_q' => $ref_form,
            'errors' => "There's already data with same service type and Company!"
        ]);
    }
}
else{
    $ref_form = array();
    $ref_form['form_question_name'] = $request->form_name;
    $ref_form['is_active'] = $request->is_active;
    $ref_form['ref_service_type_id'] = $request->service_type;
    $ref_form['ref_coy_id'] = $request->ref_coy_id;
    RefFormQuestion::where('id', $request->ref_form_question_id)->update($ref_form);

    $formQ = RefFormQuestion::has('serviceType')->has('refCoy')->where('id', $request->ref_form_question_id)->first();
    $formQ['ref_form_question_h'] = RefFormQuestionH::where('ref_form_question_id', $request->ref_form_question_id)->where('is_active', '1')->orderBy('seq_no', 'asc')->get();

    for ($x = 0; $x < count($formQ['ref_form_question_h']); $x++) {
        $formQ['ref_form_question_h'][$x]['ref_form_question_d'] = RefFormQuestionD::where('ref_form_question_h_id',
            $formQ['ref_form_question_h'][$x]['id'])->where('is_active', '1')->orderBy('seq_no', 'asc')->get();
    }

    return view('pages.form_question.edit_detail', [
        'serviceType' => $serviceType,
        'coy' => $coys,
        'form_q' => $formQ,
        'errors' => ""
    ]);
}
}

```

Segmen Program 4.1.4 Segment Program *edit Form Question header*

Setelah user merubah bagian header dari *form question*, user menekan *button “Next”* dan akan diarahkan ke halaman *edit form question “detail”* seperti gambar 4.1.7.

The screenshot shows a web interface titled "Edit Detail Form". At the top right, there is a blue "Submit" button. Below the title, there is a summary section with the following information: Form Name: Form Burger King Toiletries 2, Service Type: Toiletries, Company Name: Burger King, Number of signatures: 1, and Status: Active. The main content area is divided into two sections, each representing a floor. The first section is labeled "1" and "Lantai 1", with a blue edit icon. Below it is a "Detail" section for "Toilet lantai 1" with a plus sign. The second section is labeled "2" and "Lantai 2", also with a blue edit icon. Below it is a "Detail" section for "Toilet Lantai 2" with a plus sign. At the bottom of the form, there is a button labeled "+ Tambah".

Gambar 4.1.7 Halaman *edit form question “detail”*

```

public function update_detail(Request $request)
{
    $a = array();
    $a = $request->header_seq;
    $index_header = array_key_last($a);
    $ref_form = RefFormQuestion::where('id', $request->ref_form_question_id)->first();
    for ($x = 0; $x <= $index_header; $x++) {
        if(array_key_exists($x, $request->header_seq)){
            $form_header = array();

            if(isset($request->form_h_id[$x]) && $request->form_h_id[$x] != '0'){
                $form_h = array();
                $form_h['question_header'] = $request->header_desc[$x];
                $form_h['seq_no'] = $request->header_seq[$x];
                $form_h['is_active'] = '1';

                RefFormQuestionH::where('id', $request->form_h_id[$x])->update($form_h);
                $form_header = RefFormQuestionH::where('id', $request->form_h_id[$x])->first();
            }else{
                $form_h = array();
                $form_h['ref_form_question_id'] = $request->ref_form_question_id;
                $form_h['question_header'] = $request->header_desc[$x];
                $form_h['seq_no'] = $request->header_seq[$x];
                $form_h['is_active'] = '1';

                $form_h = RefFormQuestionH::create($form_h);
                $form_header = RefFormQuestionH::where('ref_form_question_id', $form_h->ref_form_question_id)->where('question_header', $form_h->question_header)
                ->where('seq_no', $form_h->seq_no)->where('is_active', $form_h->is_active)->first();
            }

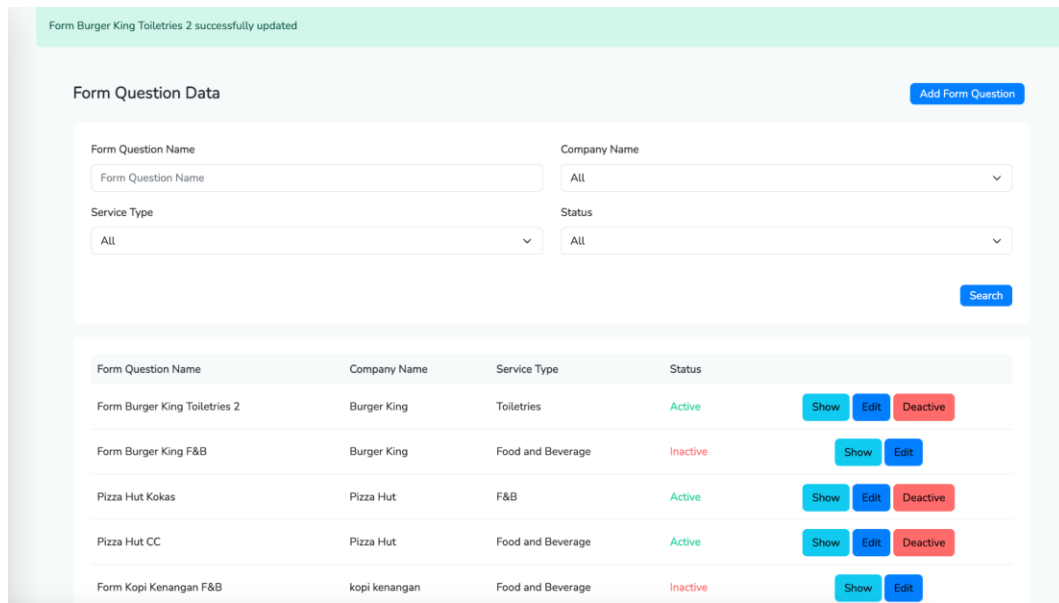
            $form_d_deactive = RefFormQuestionD::where('ref_form_question_h_id', $form_header->id)->whereNotIn('id', $request->form_d_id[$x])->get();
            for ($sa = 0; $sa < count($form_d_deactive); $sa++) {
                $form_d = array();
                $form_d['is_active'] = '0';
                RefFormQuestionD::where('id', $form_d_deactive[$sa]->id)->update($form_d);
            }

            $b = array();
            $b = $request->detail_question[$x];
            $index_detail = array_key_last($b);
            $seq_no = 1;
            for ($i = 0; $i <= $index_detail; $i++) {
                if(array_key_exists($i, $request->detail_question[$x])){
                    if(isset($request->form_d_id[$x][$i]) && $request->form_d_id[$x][$i] != '0'){
                        $form_d = array();
                        $form_d['seq_no'] = $seq_no;
                        $form_d['question'] = $request->detail_question[$x][$i];
                        RefFormQuestionD::where('id', $request->form_d_id[$x][$i])->update($form_d);
                    }else{
                        $form_d = array();
                        $form_d['ref_form_question_h_id'] = $form_header->id;
                        $form_d['seq_no'] = $seq_no;
                        $form_d['is_active'] = '1';
                        $form_d['question'] = $request->detail_question[$x][$i];
                        $form_detail = RefFormQuestionD::create($form_d);
                    }
                    $seq_no++;
                }
            }
        }
    }
}

```

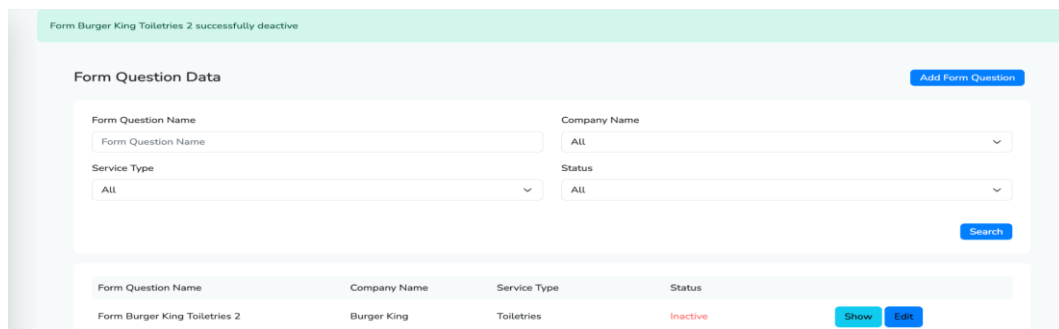
Segmen Program 4.1.5 Segment Program *edit Form Question detail*

Pada halaman tersebut *user* dapat menambahkan detail pertanyaan atau detail *form*. Setelah itu *user* menekan *button* “*Submit*” untuk menyimpan perubahan data tersebut dan akan diarahkan ke halaman paging dengan pesan bahwa data berhasil tersimpan seperti pada gambar 4.1.8.



Gambar 4.1.8 Halaman *paging* dengan pesan sukses merubah data

Jika user ingin melakukan deaktivasi dari *form question* yang sudah ditambahkan maka *user* bisa menekan *button* “*Deactive*”, setelah user menekan *button* tersebut maka *status form question* akan menjadi “*Inactive*” dan akan diarahkan ke halaman *paging* dengan pesan bahwa data sukses untuk di-deaktivasi seperti gambar 4.1.9.



Gambar 4.1.9 Halaman *paging* dengan pesan sukses deaktivasi data

```
public function deactivate(Request $request, $id)
{
    $ref_form = RefFormQuestion::where('id', $id)->first();
    $ref_form['is_active'] = '0';

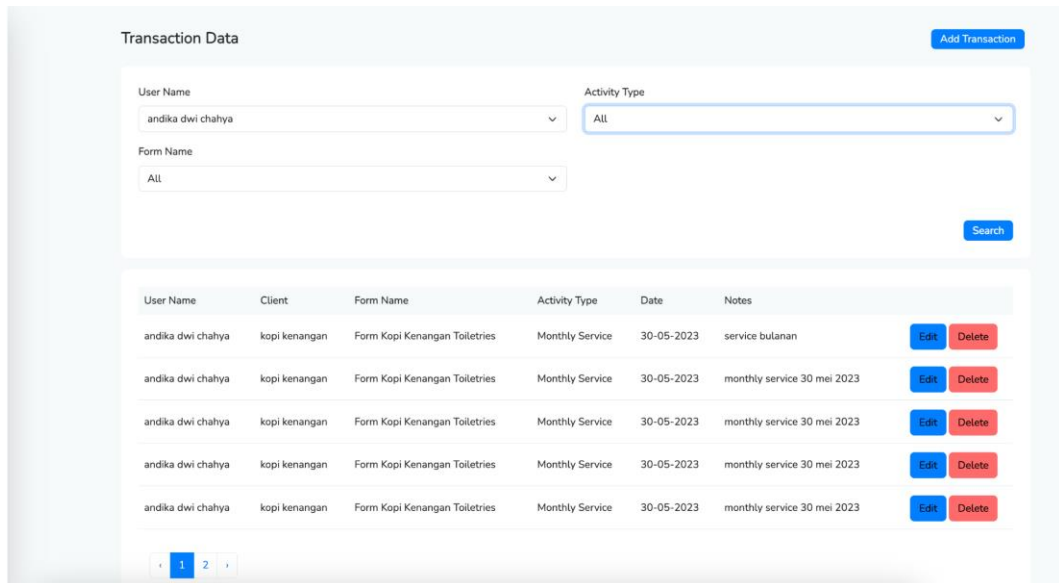
    $ref_form->save();

    return redirect('form_question/')->with('message', $ref_form->form_question_name.' successfully deactivate');
}
```

Segmen Program 4.1.6 Segment Program *deactivate Form Question*

4.1.2 Pengujian Menu Transaction

Menu Transaction adalah menu yang digunakan oleh teknisi untuk mencatat kondisi dari aset atau barang yang berada pada client saat melakukan kunjungan. Halaman awal untuk menu ini dapat dilihat pada gambar 4.2.1, pada halaman ini user bisa melakukan pencarian data transaksi berdasarkan dari nama form, nama user , activity type dan status.



Gambar 4.2.1 Halaman *paging menu transaction*

```

public function index(Request $request)
{
    $trx = TrxService::has('activityType')->has('user')->has('refFormQuestion');
    $serviceType = RefServiceType::where('is_active', '=', '1')->get();
    $activityType = RefActivityType::where('is_active', '=', '1')->get();
    $form_q = RefFormQuestion::has('refCoy')->where('is_active', '=', '1')->get();
    $user_login = Auth::user();
    if($user_login->level == '1'){
        $users = User::all();
        $coys = RefCoy::all();
    }else{
        $users = User::where('id', '=', $user_login->id)->get();
        $coys = RefCoy::where('area_id', '=', $user_login->area_id)->get();
        $coy_id= array();
        $form_id= array();
        for ($x = 0; $x < count($coys); $x++) {
            $coy_id[$x] = $coys[$x]->id;
        }
        $form = RefFormQuestion::has('refCoy')->where('is_active', '=', '1')->whereIn('ref_coy_id', $coy_id)->get();
        for ($x = 0; $x < count($form); $x++) {
            $form_id[$x] = $form[$x]->id;
        }
        $trx->where('ref_user_id', '=', $user_login->id);
    }

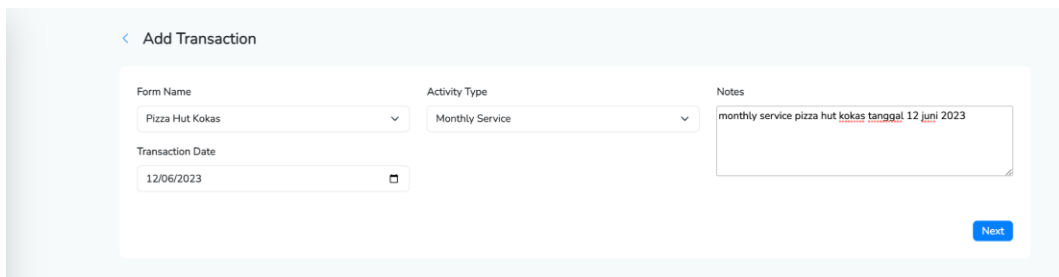
    if ($request->user_id) {
        $trx->where('ref_user_id', $request->user_id);
    }
    if ($request->form_id) {
        $trx->where('ref_form_question_id', $request->form_id);
    }
    if ($request->activity_type) {
        $trx->where('ref_activity_type_id', $request->activity_type);
    }

    return view('pages.trx.index', [
        'trx' => $trx->orderBy('created_at', 'desc')->paginate(5),
        'activityType' => $activityType,
        'user' => $users,
        'form_q' => $form_q
    ]);
}

```

Segmen Program 4.2.1 Segment Program *paging Transaction*

Ketika teknisi ingin menginputkan kunjungannya pada client maka teknisi menekan *button “Add Transaction”* dan teknisi akan diarahkan kepada halaman *add transaction “header”* seperti pada gambar 4.2.2.



Gambar 4.2.2 Halaman *add transaction “header”*

```

public function add_detail(Request $request)
{
    $activityType = RefActivityType::where('is_active', '=', '1')->get();
    $form_q = RefFormQuestion::has('refCoy')->where('is_active', '=', '1')->get();

    $trx = array();
    $trx['ref_user_id'] = Auth::user()->id;
    $trx['supervisor_id_in_charge'] = Auth::user()->supervisor_id;
    $trx['ref_form_question_id'] = $request->ref_form_question_id;
    $trx['ref_activity_type_id'] = $request->activity_type;
    $trx['service_dt'] = $request->service_dt;
    $trx['notes'] = $request->notes;
    $trxService = TrxService::create($trx);

    $trxS = TrxService::where('id', $trxService->id)->first();

    $form = RefFormQuestion::has('serviceType')->where('id', $trxS->ref_form_question_id)->first();
    $formH = RefFormQuestionH::where('ref_form_question_id', $trxS->ref_form_question_id)->where('is_active', '1')->orderBy('seq_no', 'asc')->get();
    $formD = array();
    for ($x = 0; $x < count($formH); $x++) {
        $formD[$x] = RefFormQuestionD::where('ref_form_question_h_id', $formH[$x]['id'])->where('is_active', '1')->orderBy('seq_no', 'asc')->get();
    }

    return view('pages.trx.add_detail', [
        'activityType' => $activityType,
        'trxS' => $trxS,
        'form' => $form,
        'formH' => $formH,
        'formD' => $formD,
        'errors' => "",
        'trx_id' => $trxS->id,
        'form_q' => $form_q
    ]);
}

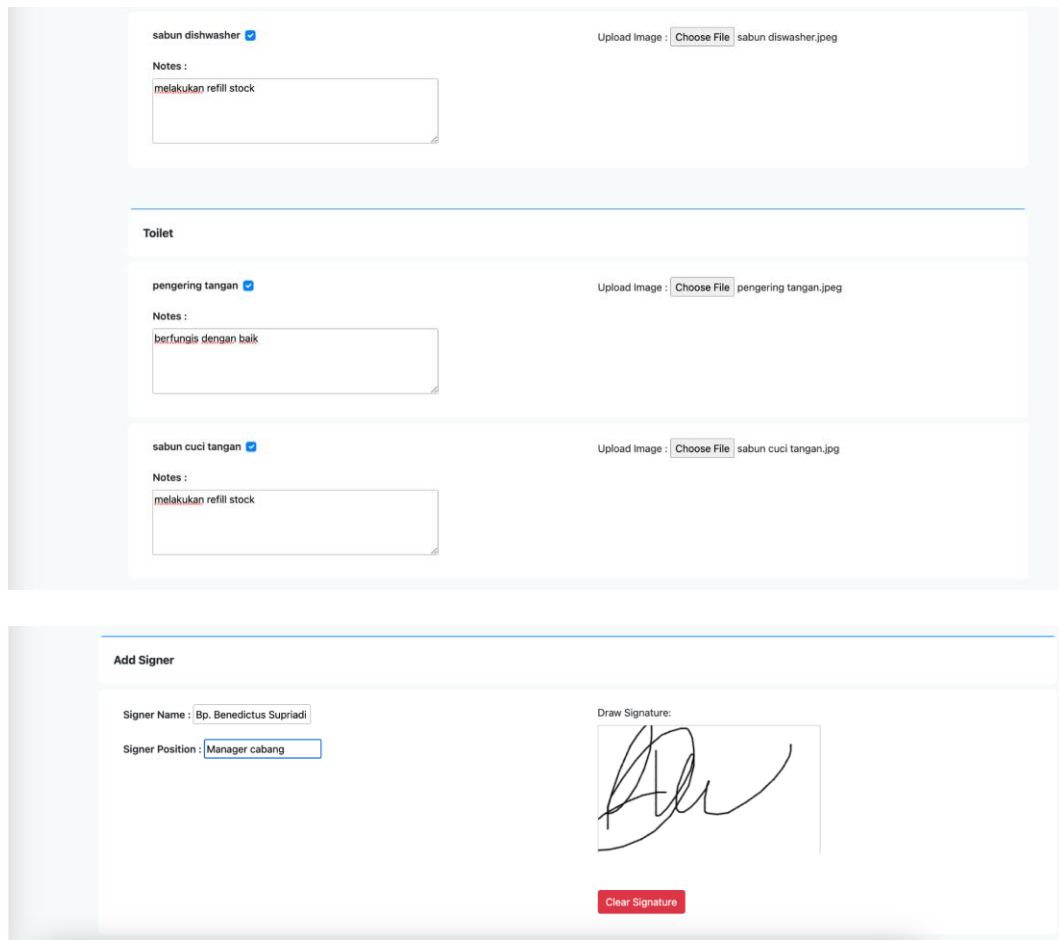
```

Segmen Program 4.2.2 Segment Program *save transaction header*

Pada halaman ini teknisi mengisi *form name*, jenis aktivitas, tanggal dan catatan terkait dengan kunjungan yang dilakukan ke-*client*. Setelah menginputkan data tersebut teknisi menekan *button “Next”* untuk diarahkan ke-halaman *add transaction detail* seperti pada gambar 4.2.3.

The screenshot shows a web form titled "Add Transaction Detail". At the top right is a blue "Submit" button. The form is divided into several sections:

- Form Name:** A dropdown menu with "Pizza Hut Kokas" selected.
- Activity Type:** A dropdown menu with "Monthly Service" selected.
- Transaction Date:** A text input field containing "12/06/2023".
- Notes:** A large text area containing "monthly service pizza hut kokas tanggal 12 juni 2023".
- Dapur (Kitchen):** A section with a "dishwasher" checkbox that is checked. To the right is an "Upload Image" button with "Choose File" and the filename "dishwasher.avif". Below this is a "Notes:" text area containing "berfungsi dengan baik".



Gambar 4.2.2 Halaman *add transaction "detail"*

```

public function save_detail(Request $request)
{
    for ($x = 0; $x < count($request->form_h_id); $x++) {
        $trx_h = array();
        $trx_h['trx_service_id'] = $request->trx_id;
        $trx_h['ref_form_question_h_id'] = $request->form_h_id[$x];
        $trx_header = TrxServiceH::create($trx_h);

        for ($i = 0; $i < count($request->form_d_id[$x]); $i++) {
            $trx_d = array();
            $trx_d['trx_service_h_id'] = $trx_header->id;
            $trx_d['ref_form_question_d_id'] = $request->form_d_id[$x][$i];
            $trx_d['answer'] = isset($request->answer[$x][$i]) && $request->answer[$x][$i] != "" ? $request->answer[$x][$i] : 'Tidak';
            $trx_d['notes'] = $request->notes[$x][$i];

            $image_path = $request->file('image')[$x][$i]->store('image', 'public');

            $name = $request->file('image')[$x][$i]->getClientOriginalName();
            $path = $request->image[$x][$i];
            $trx_d['image_path'] = $image_path;
            $trx_detail = TrxServiceD::create($trx_d);
        }
    }

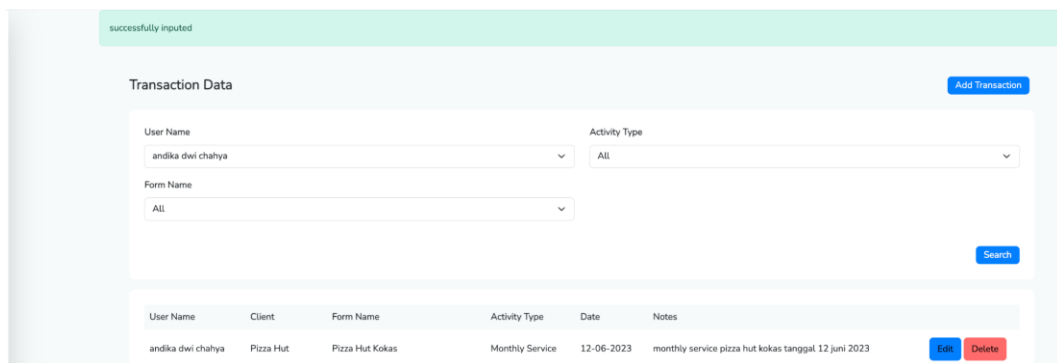
    $trx_s = array();
    $trx_s['trx_service_id'] = $request->trx_id;
    $trx_s['signer_name'] = $request->signer_name;
    $trx_s['signer_position'] = $request->signer_position;
    $trx_s['seq_no'] = '1';
    $trx_s['sign_path'] = $request->signed;
    $trx_signer = TrxServiceSigner::create($trx_s);

    return redirect('trx/')->with('message', 'successfully inputed');
}

```

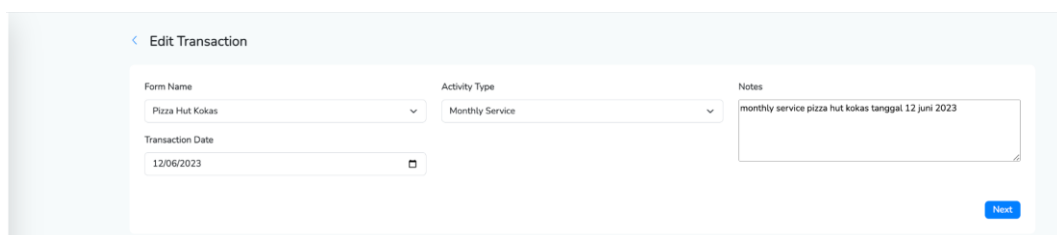
Segmen Program 4.2.3 Segment Program *save transaction detail*

Pada halaman ini teknisi melaporkan pekerjaan yang dilakukan pada *client*, data yang diinputkan berupa gambar dari *asset*, catatan terkait aset tersebut dan meminta tanda tangan dari penanggung jawab sebagai bukti bahwa teknisi telah melakukan tugas dengan sesuai. Setelah melakukan pengisian tersebut teknisi menekan *button* “*Submit*”, data akan tersimpan dan teknisi akan diarahkan ke halaman *paging* dengan pesan sukses seperti pada gambar 4.2.3.



Gambar 4.2.3 Halaman *paging* dengan pesan sukses

Jika teknisi ingin merubah data transaksi yang sudah ditambahkan, maka teknisi perlu menekan *button* “*Edit*” pada data transaksi yang akan dirubah, setelah user menekan *button* tersebut maka user akan diarahkan ke-halaman *Edit Transaction* “*header*” seperti pada gambar 4.2.4.



Gambar 4.2.4 Halaman *edit transaction* “*header*”

```

public function update_header(Request $request)
{
    $activityType = RefActivityType::where('is_active', '=', '1')->get();
    $coys = RefCoy::all();
    $form_q = RefFormQuestion::has('refCoy')->where('is_active', '=', '1')->get();

    $trx = array();
    $trx['ref_user_id'] = Auth::user()->id;
    $trx['supervisor_id_in_charge'] = Auth::user()->supervisor_id;
    $trx['ref_form_question_id'] = $request->ref_form_question_id;
    $trx['ref_activity_type_id'] = $request->activity_type;
    $trx['service_dt'] = $request->service_dt;
    $trx['notes'] = $request->notes;
    TrxService::where('id', $request->trx_id)->update($trx);

    $trxS = TrxService::where('id', $request->trx_id)->first();

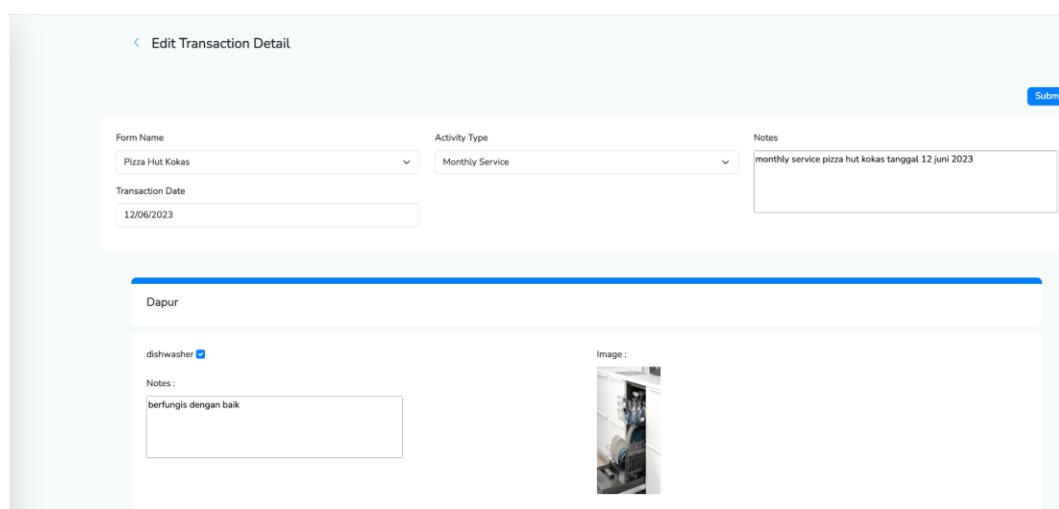
    $trxH = TrxServiceH::where('trx_service_id', $trxS->id)->has('refFormQuestionH')->get();
    for($a = 0; $a < count($trxH); $a++){
        $trxD[$a] = TrxServiceD::where('trx_service_h_id', $trxH[$a]->id)->has('refFormQuestionD')->get();
    }
    $trxSigner = TrxServiceSigner::where('trx_service_id', $trxS->id)->first();

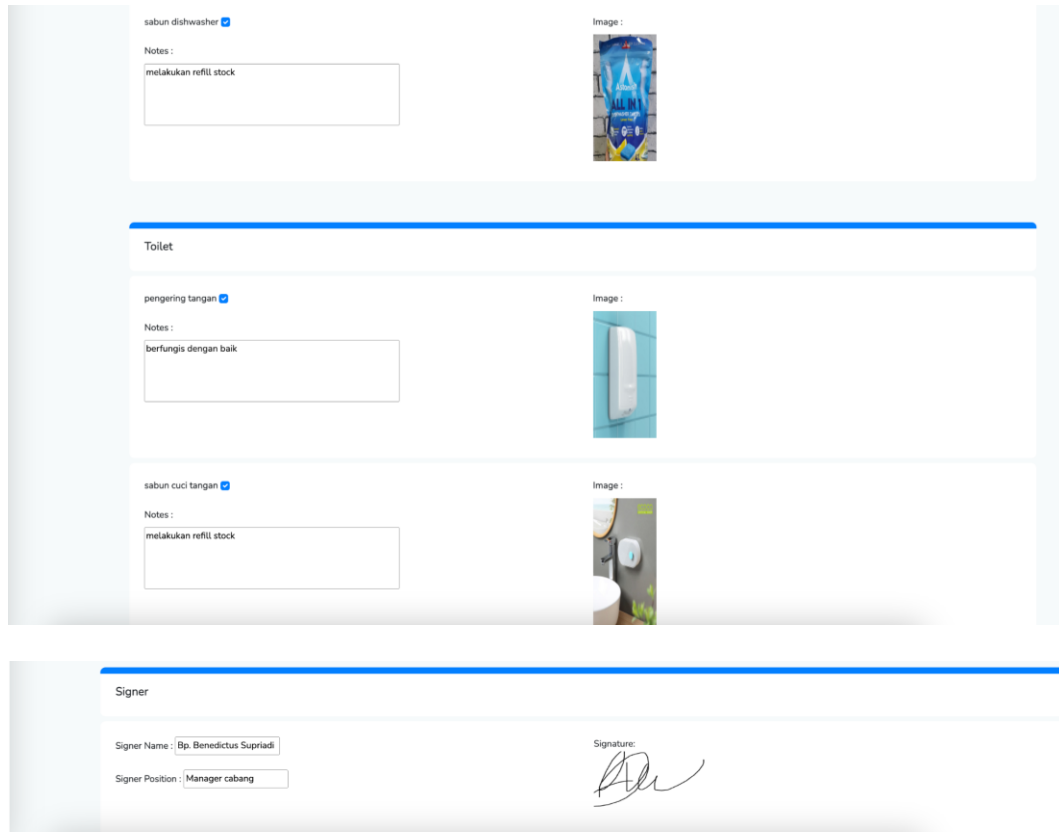
    return view('pages.trx.edit_detail', [
        'trxH' => $trxH,
        'trxD' => $trxD,
        'form_q' => $form_q,
        'activityType' => $activityType,
        'trxS' => $trxS,
        'trxSigner' => $trxSigner,
        'errors' => ""
    ]);
}

```

Segmen Program 4.2.3 Segment Program *edit transaction header*

Setelah user merubah data “*header*” tersebut, user menekan *button* “*Next*” untuk menuju halaman *edit transaction detail* seperti pada gambar 4.2.5.





Gambar 4.2.5 Halaman *edit transaction “detail”*

```

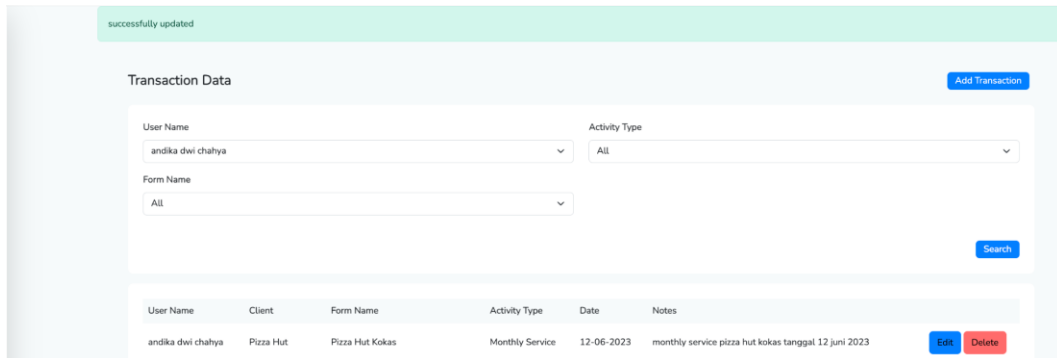
public function update_detail(Request $request)
{
    for($a = 0; $a < count($request->trx_h_id); $a++){
        $trxH = TrxServiceH::where('id', $request->trx_h_id[$a])->first();
        for($i = 0; $i < count($request->trx_d_id[$a]); $i++){
            $trx_d = array();
            $trx_d['answer'] = isset($request->answer[$a][$i]) && $request->answer[$a][$i] != "" ? $request->answer[$a][$i] : 'Tidak';
            $trx_d['notes'] = $request->notes[$a][$i];
            TrxServiceD::where('id', $request->trx_d_id[$a][$i])->update($trx_d);
        }
    }

    return redirect('trx/')->with('message', 'successfully updated');
}

```

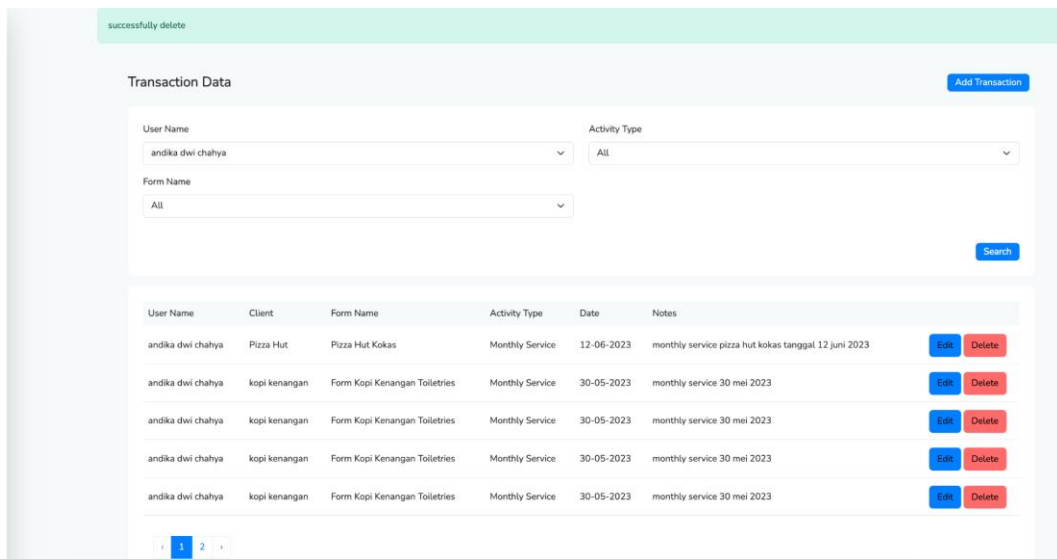
Segmen Program 4.2.4 Segment Program *edit transaction detail*

Setelah merubah data transaksi, user menekan *button “Submit”* dan user akan diarahkan ke halaman paging dengan pesan sukses seperti gambar 4.2.6.



Gambar 4.2.6 Halaman *paging* dengan pesan sukses

Jika teknisi ingin menghapus data transaksinya, maka teknisi bisa menekan *button* “Delete”, setelah menekan *button* tersebut maka data transaksi akan terhapus dan akan memunculkan pesan sukses menghapus data seperti pada gambar 4.2.7.



Gambar 4.2.7 Halaman *paging* dengan pesan sukses

```

public function delete(Request $request, $id)
{
    $trx = TrxService::where('id', $id)->first();
    $trxS = TrxServiceSigner::where('trx_service_id',$id)->get();

    $trxH = TrxServiceH::where('trx_service_id',$trx->id)->get();
    for($a = 0; $a < count($trxH); $a++){
        $trxHeader = TrxServiceH::where('id',$trxH[$a]->id)->first();
        $trxD = TrxServiceD::where('trx_service_h_id',$trxH[$a]->id)->get();
        for($i = 0; $i < count($trxD); $i++){
            $trxDetail = TrxServiceD::where('id',$trxD[$i]->id)->first();
            $trxDetail->delete();
        }
        $trxHeader->delete();
    }

    for ($c = 0; $c < count($trxS); $c++) {
        $trxSigner = TrxServiceSigner::where('id',$trxS[$c]->id)->first();
        $trxSigner->delete();
    }

    $trx->delete();

    return redirect('trx/')->with('message',' successfully delete');
}

```

Segmen Program 4.2.5 Segment Program *delete transaction*

4.1.3 Blackbox Testing

Pengujian *black box* berfokus kepada pengujian dengan melihat fungsi-fungsi yang ada dalam sistem tanpa harus mengetahui bagaimana fungsi tersebut dibuat sistemnya. Pada sistem ini, pengujian merujuk pada fungsi-fungsi yang dimiliki. Kemudian membandingkan hasil keluaran sistem dengan hasil yang diharapkan. Bila hasil yang diharapkan sesuai dengan hasil pengujian, hal ini berarti aplikasi sesuai dengan desain yang telah ditentukan sebelumnya. Bila belum sesuai maka perlu dilakukan pengecekan lebih lanjut dan perbaikan. Berikut ini pengujian *black box* terhadap sistem ini :

Tabel 4.1 *Blackbox Testing*

<i>Test ID</i>	<i>Use Case</i>	Kondisi Pengujian	Hasil yang Diharapkan	Hasil	Status	<i>Outstanding Issue</i>
1	<i>Login</i>	<i>Login Teknisi</i>	Menampilkan <i>dashboard</i> teknisi	<i>Dashboar</i> d teknisi tampil setelah	<i>Meet expectation</i>	N/A

			setelah <i>input</i> <i>Username</i> dan <i>password</i>	memasukkan <i>Username</i> dan <i>password</i>		
		<i>Login Sales</i>	Menampilkan <i>dashboard Sales</i> setelah <i>input Username</i> dan <i>password</i>	<i>Dashboard</i> teknisi tampil setelah memasukkan <i>Username</i> dan <i>password</i>	<i>Meet expectation</i>	<i>N/A</i>
		<i>Login Admin</i>	Menampilkan <i>dashboard admin</i> setelah <i>input Username</i> dan <i>password</i>	<i>Dashboard</i> teknisi tampil setelah memasukkan <i>Username</i> dan <i>password</i>	<i>Meet expectation</i>	<i>N/A</i>
2	Mengelola data <i>Master Client</i>	Klik <i>Menu Master Area</i>	Menampilkan <i>Menu Master Area</i>	Setelah klik <i>Menu</i> , laman langsung berganti tampilan ke <i>Menu Master Area</i>	<i>Meet expectation</i>	<i>N/A</i>
		Menambahkan <i>Master Area</i>	<i>Master Area</i> ditambahkan dan tampil di dalam <i>Menu Master area</i>	Setelah menambahkan <i>Master area</i> , laman menampilkan data yang terupdate	<i>Meet expectation</i>	<i>N/A</i>
		Klik <i>Menu Master Client</i>	Menampilkan <i>Menu Master Client</i>	Setelah klik <i>Menu</i> , laman langsung berganti tampilan ke <i>Menu Master Client</i>	<i>Meet expectation</i>	<i>N/A</i>
		Menambahkan	<i>Master Client</i> ditambahkan	Setelah menambahkan	<i>Meet expectation</i>	<i>N/A</i>

		<i>Master Client</i>	dan tampil di dalam <i>Menu Master Client</i>	<i>n Master Client</i> , laman menampilkan data yang <i>terupdate</i>	<i>tation</i>	
3	Mengelola data <i>Master Activity Type</i>	Klik <i>Menu Master Activity Type</i>	Menampilkan <i>Menu Master Activity Type</i>	Setelah klik <i>Menu</i> , laman langsung berganti tampilan ke <i>Menu Master Activity Type</i>	<i>Meet expectation</i>	<i>N/A</i>
		Menambahk an <i>Activity Type</i>	<i>Activity Type</i> ditambahkan dan tampil di dalam <i>Menu Activity Type</i>	Setelah menambahkan <i>Activity Type</i> , laman menampilkan data yang <i>terupdate</i>	<i>Meet expectation</i>	<i>N/A</i>

4	Mengelola data <i>Master service Type</i>	Klik <i>Menu Master Service Type</i>	Menampilkan <i>Menu Master Service Type</i>	Setelah klik <i>Menu</i> , laman langsung berganti tampilan ke <i>Menu Master Service Type</i>	<i>Meet expectation</i>	<i>N/A</i>
		Menambahk an <i>Master Service Type</i>	<i>Master Service Type</i> ditambahkan dan tampil di dalam <i>Menu Master Service Type</i>	Setelah menambahkan <i>Master Service Type</i> , laman menampilkan data yang <i>terupdate</i>	<i>Meet expectation</i>	<i>N/A</i>
5	Mengelola data <i>Master teknisi</i>	Klik <i>Menu User</i>	Menampilkan <i>Menu User</i>	Setelah klik <i>Menu</i> , laman langsung berganti	<i>Meet expectation</i>	<i>N/A</i>

				tampilan ke <i>Menu User</i>		
		Menambahkan data <i>Master</i> teknisi pada <i>Menu User</i>	<i>Master</i> Teknisi ditambahkan dan tampil di dalam <i>Menu User</i>	Setelah menambahkan <i>Master</i> Teknisi, laman menampilkan data yang terupdate	<i>Meet expectation</i>	<i>N/A</i>
6	Mengelola data <i>Master Sales</i>	Klik <i>Menu User</i>	Menampilkan <i>Menu User</i>	Setelah klik <i>Menu</i> , laman langsung berganti tampilan ke <i>Menu User</i>	<i>Meet expectation</i>	<i>N/A</i>
		Menambahkan data <i>Master Sales</i> pada <i>Menu User</i>	<i>Master Sales</i> ditambahkan dan tampil di dalam <i>Menu User</i>	Setelah menambahkan <i>Master Sales</i> , laman menampilkan data yang terupdate	<i>Meet expectation</i>	<i>N/A</i>
7	Mengelola <i>Transaction</i>	Klik <i>Menu Transaction</i>	Menampilkan data transaksi	Setelah klik <i>Menu</i> , laman langsung berganti tampilan ke <i>Menu Transaction</i>	<i>Meet expectation</i>	<i>N/A</i>
		Menambahkan transaksi	Data transaksi ditambahkan dan tampil di dalam <i>Menu Transaction</i>	Setelah menambahkan data transaksi, laman menampilkan data yang terupdate	<i>Meet expectation</i>	<i>N/A</i>
8	Mengelola <i>Form Question Setting</i>	Klik <i>Menu Form Question Setting</i>	Menampilkan <i>Form Question Setting</i>	Setelah klik <i>Menu</i> , laman langsung berganti tampilan ke <i>Menu Transaction</i>	<i>Meet expectation</i>	<i>N/A</i>
		Menambahkan	<i>Form</i>	Setelah	<i>Meet</i>	<i>N/A</i>

		<i>n Form Question Setting</i>	<i>Question Setting</i> ditambahkan dan tampil di dalam <i>Menu Form Question Setting</i>	menambahka n <i>Form Question Setting</i> , laman menampilkan data yang <i>terupdate</i>	<i>expectation</i>	
--	--	--	--	---	--------------------	--