

## BAB V

### Implementasi dan Pengujian Sistem

Pada bab ini akan membahas penggunaan sistem informasi ACCITeam yang meliputi implementasi dan pengujian perangkat lunak yang dibuat. Implementasi ini digunakan untuk menjelaskan dan mendeskripsikan bagian-bagian yang ada pada sistem. Pengujian sistem ini digunakan untuk menganalisis apakah sistem yang dibuat sudah memenuhi target yang ingin dicapai pada penelitian ini.

#### 5.1. Definisi Perangkat Lunak

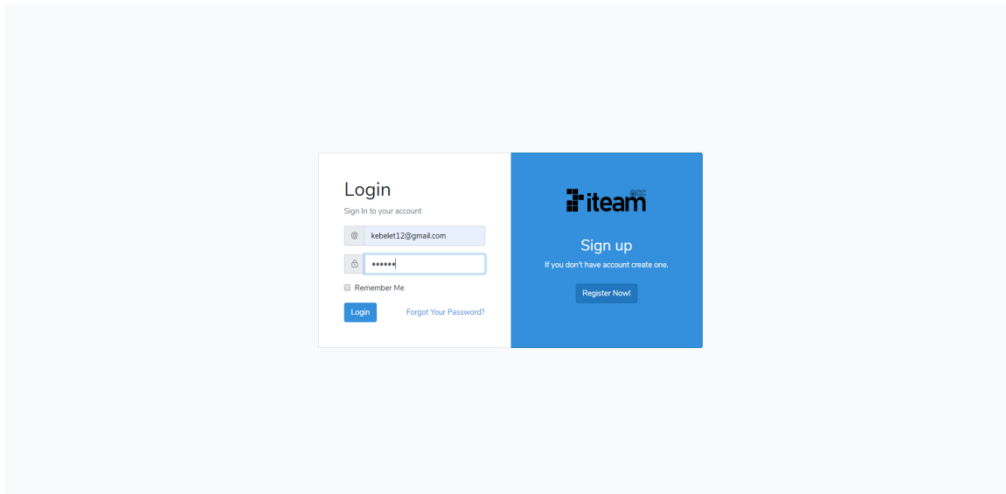
Sistem informasi *resource management* ACCITeam merupakan perangkat lunak yang dikembangkan untuk membantu para *stakeholder* dari departemen IT sebuah perusahaan secara *online*. Sistem ini dapat digunakan oleh para *stakeholder* pada departemen IT untuk melakukan pengelolaan ketersediaan karyawan dari departemen IT ataupun memantau kinerja dari karyawan departemen IT pada pengerjaan *project* yang ditanganinya. Sistem ini akan dibangun ke dalam bentuk *website*.

Sistem informasi *resource management* ini digunakan untuk mengelola segala data yang berkaitan dengan *project* yang masuk di dalam departemen IT, seperti anggota *team* dan *task*. Sistem informasi ini berjalan pada *web browser* *Chrome* ataupun *Mozilla Firefox* dan dibuat menggunakan bahasa pemrograman PHP dengan memanfaatkan *framework* *Laravel*. Sistem informasi ini juga dapat diakses melalui *browser mobile*. Sistem ini menggunakan basis data *MySQL*. Untuk mempermudah kerja dari pengguna, berjalannya sistem informasi didukung dengan penggunaan *library* dari *Javascript* yaitu *JQuery* dan *Ajax*. Penggunaan *JQuery* dan *Ajax* ini nantinya mengurangi adanya *reload* halaman web saat terjadi pemanggilan fungsi. Sistem informasi ini juga dibuat agar tampilannya dinamis. Dinamis yang dimaksud disini adalah mengikuti *platform* yang digunakan untuk mengaksesnya, sehingga tampilan pada *browser desktop* ataupun *mobile* akan terlihat berbeda mengikuti alat yang digunakan untuk mengakses

## 5.2. Implementasi Sistem

### Aplikasi Web

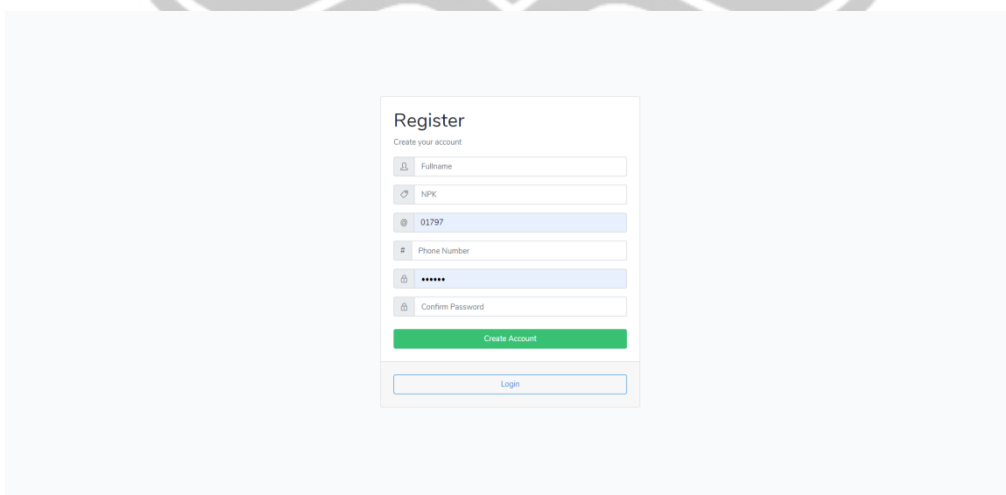
#### 5.2.1. *Login*



**Gambar 5. 1 Tampilan *Login***

Pada gambar 5.1 merupakan halaman untuk melakukan otentikasi *login*. Halaman ini merupakan halaman pertama yang ditampilkan pada saat akan mengakses sistem informasi ini. Terdapat masukan untuk email dan *password*, serta tombol *login* untuk melakukan proses otentikasi. Pengguna yang dapat melakukan proses *login* adalah pengguna yang telah terdaftar pada basis data sistem informasi ACCITeam dan masih aktif. Terdapat sebuah tombol *Register* yang akan merujuk ke halaman *register*, yang memiliki fungsi untuk mendaftarkan diri sendiri untuk dapat mengakses sistem ini.

#### 5.2.2. *Register*

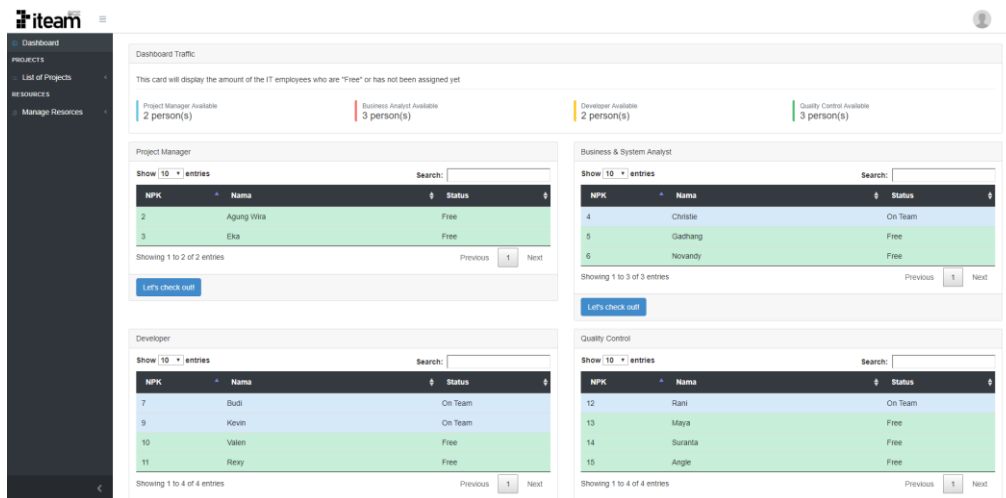


**Gambar 5. 2 Tampilan *Register***

Pada Gambar 5.2 merupakan halaman untuk melakukan pendaftaran diri dari pengguna yang ingin mengakses ke sistem. Terdapat masukan untuk *fullname*, *npk*, *email*, *phone number*, *password* dan *confirm password*, serta tombol *register* untuk melakukan

fungsi pendaftaran diri. Terdapat juga sebuah tombol *Login* yang akan merujuk ke halaman *login*, yang memiliki fungsi untuk melakukan otentikasi *login*.

### 5.2.3. Tampilan Dashboard



**Gambar 5. 3 Tampilan Dashboard**

Setelah melakukan proses *login*, akan muncul halaman untuk menampilkan *Dashboard* seperti pada gambar yang ditunjukkan pada Gambar 5.3 diatas. Halaman ini akan menunjukkan jumlah *user* atau pegawai yang memiliki status *free* yang ada di departemen IT yang dibagi berdasarkan perannya.

Halaman ini juga menampilkan tabel yang berisi data *user* dari setiap peran. Data yang ditampilkan adalah *npk*, *fullname* dan status. Terdapat juga tombol pada setiap tabel yang akan merujuk ke halaman pengelolaan data *user* sesuai dengan peran yang ditujukan di judul tabel. Pada tabel tersebut akan diberikan petunjuk sebagai pembeda status dari setiap *user*. Adapun warna biru menandakan *user* berstatus *onteam* atau sedang dalam sebuah *project*, warna hijau menandakan *user* berstatus *free* atau tidak sedang mengerjakan *project* maupun diminta menjadi anggota *team*, dan yang terakhir berwarna kuning yang menandakan *user* berstatus *booked* atau sudah dalam daftar permintaan menjadi anggota *team* sebuah *project*. Berikut adalah potongan kode yang digunakan untuk mengambil data setiap *user* untuk ditampilkan pada tabel :

```

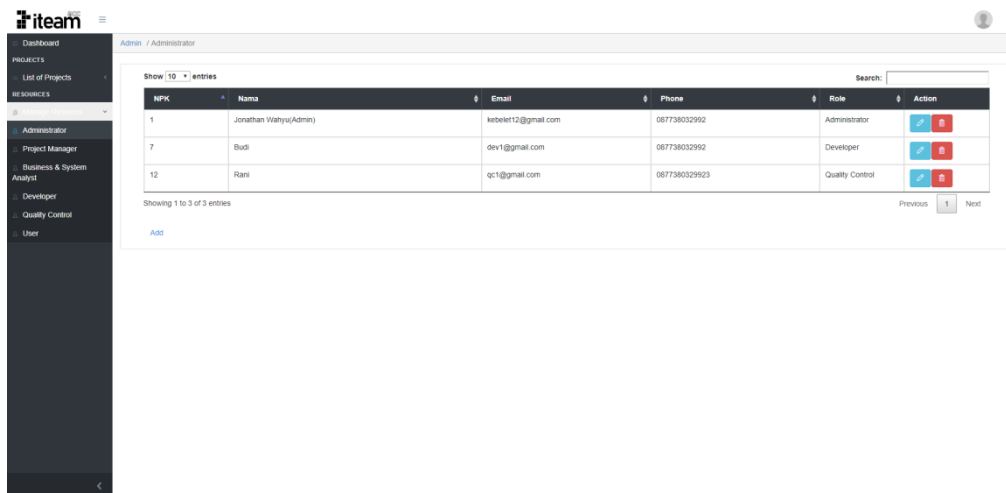
$model = DB::table('mst_users')
->select(
  'npk',
  'fullname',
  'status'
)
->whereNull('deleted_at');







if($type == "pm")
{ $model->where('role', 'pm'); }
elseif($type == "bsa")
{ $model->where('role', 'bsa'); }
elseif($type == "dev")
{ $model->where('role', 'dev'); }
elseif($type == "qc")
{ $model->where('role', 'qc'); }

return DataTables::of($model)
->setRowClass(function($model) {
    if($model->status == 'onteam')
    { return 'table-info'; }
    elseif($model->status == 'booked')
    { return 'table-warning'; }
    elseif($model->status == 'free')
    { return 'table-success'; }
})
->addColumn('status', function($model) {
    if($model->status == 'onteam')
    { $status = 'On Team'; }
    elseif($model->status == 'booked')
    { $status = 'Booked'; }
    elseif($model->status == 'free')
    { $status = 'Free'; }
    return $status;
})
->rawColumns([])
->toJson();
}

```

## 5.2.4. Tampilan *Index User*



NPK	Nama	Email	Phone	Role	Action
1	Jonathan Wahyu(Admin)	kebelet12@gmail.com	087738032992	Administrator	 
7	Budi	dev1@gmail.com	087738032992	Developer	 
12	Rani	qc-1@gmail.com	0877380329923	Quality Control	 

**Gambar 5. 4 Tampilan *Index User***

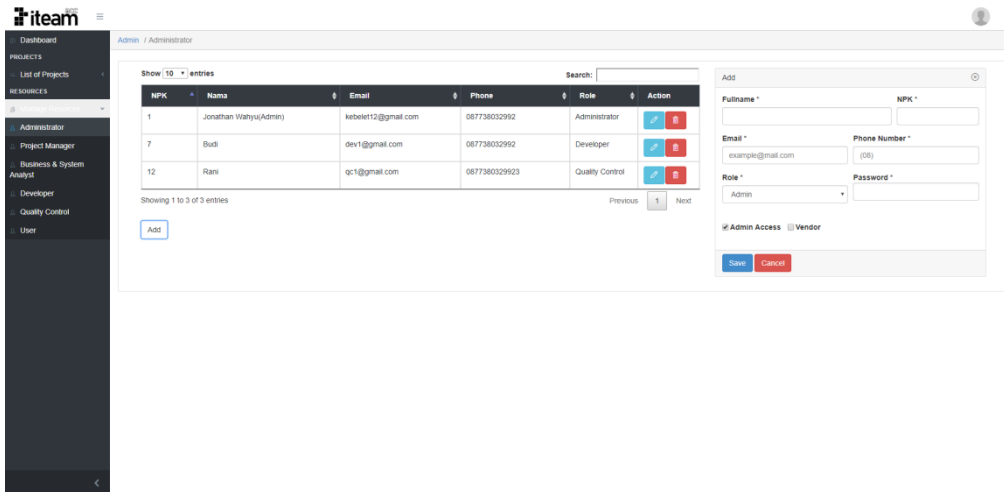
Pada gambar 5.4 merupakan halaman yang menampilkan semua data dari *user* sesuai dengan perannya atau status *is\_admin* yang dimiliki *user*, dimana hal tersebut dapat dilihat atau dipilih melalui *menu* yang ada. Maksud dari status *is\_admin* digunakan untuk memberikan informasi bahwa *user* tersebut adalah ketua dari *user* dengan peran *dev* ataupun *qc*. Tidak semua pengguna dapat mengakses halaman ini, hanya pengguna dengan peran *admin* yang dapat mengaksesnya. Halaman ini akan menampilkan data *user* di dalam sebuah tabel. Terdapat juga beberapa tombol yang berfungsi untuk menambah data *user*, mengubah data *user*, dan menghapus data *user*. Berikut adalah potongan kode yang digunakan untuk mengambil data setiap *user* untuk ditampilkan pada tabel pada halaman *Index User* :

```

$model = DB::table('mst_users')
->select(
  'mst_users.id',
  'mst_users.npk',
  'mst_users.fullname',
  'mst_users.email',
  'mst_users.phone_number',
  'mst_users.is_admin',
  'mst_users.is_vendor'
)
->whereNull('mst_users.deleted_at');
if($type == "pm")
{ $model->where('mst_users.role', 'pm'); }
elseif($type == "bsa")
{ $model->where('mst_users.role', 'bsa'); }
elseif($type == "dev")
{ $model->where('mst_users.role', 'dev'); }
elseif($type == "qc")
{ $model->where('mst_users.role', 'qc'); }
elseif($type == "user")
{ $model->where('mst_users.role', 'user'); }
return DataTables::of($model)
->setRowClass(function($model){
  $cekadmin = $model->is_admin;
  $cekvendor = $model->is_vendor;
  if($cekadmin == '1')
  { return 'table-info'; }
  if($cekvendor == '1')
  { return 'table-danger'; }
})
->addColumn('action', function ($model){
return '<a data-id="'. $model->id.'" class="btn-edit btn btn-info
btn-md" style="margin-right:1px"><i class="icons cui-pencil"
style="color:white"></i> </a><a
href="'.url('/administrator/user').'/'.$model->id.'/delete"
data-id="'. $model->id.'" class="btn btn-danger btn-md"><i
class="icons cui-trash"></i></a>';
})
->rawColumns(['action'])
->toJson();

```

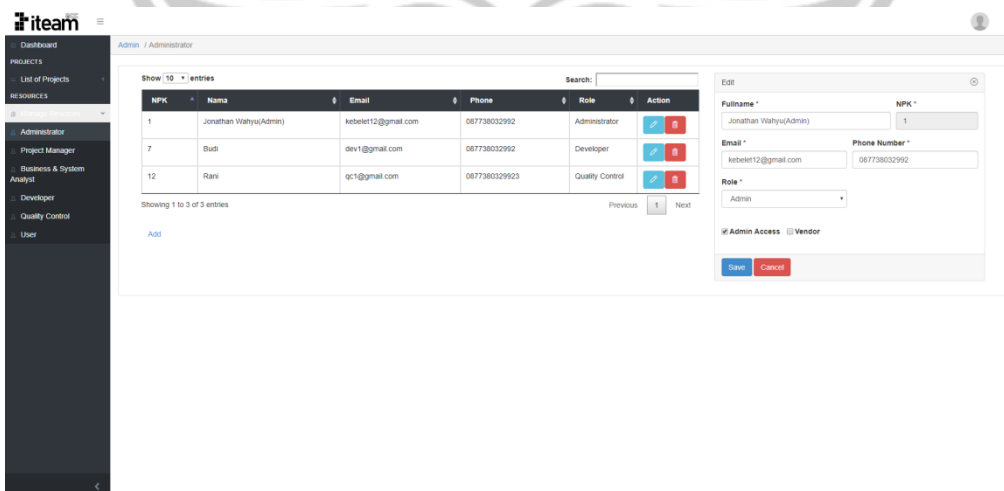
## 5.2.5. Tambah User



Gambar 5. 5 Tampilan Tambah User

Pada gambar 5.5 merupakan halaman yang sama dengan halaman tampilan *index user* dengan tambahan sebuah *card* baru untuk menambah data *user*. Halaman ini dapat diakses ketika pengguna menekan tombol *Add* dan secara dinamis tampilan akan berubah seperti yang ditampilkan diatas. Adapun masukan yang diperlukan untuk menambah data *user* adalah *fullname*, *npk*, *email*, *phone number*, *role*, dan *password* pada *textbox* yang sesuai. Pada saat pertama kali ditampilkan, *textbox* yang ada tidak berisi data apapun. Terdapat tombol *save* memiliki fungsi untuk menyimpan data *user* yang diisi dalam *textbox* setelah melewati validasi yang diberikan oleh sistem, dan juga terdapat tombol *cancel* untuk membatalkan dan menghapus semua data yang sudah diinputkan oleh pengguna.

## 5.2.6. Ubah User



Gambar 5. 6 Tampilan Ubah User

Pada gambar 5.6 merupakan halaman yang sama dengan halaman tampilan *index user* dengan tambahan sebuah *card* baru untuk mengubah data *user*. Halaman ini dapat diakses ketika pengguna menekan tombol *Edit* atau yang memiliki *icon* pensil dan secara

dinamis tampilan akan berubah seperti yang ditampilkan diatas. Adapun masukan yang diperlukan untuk mengubah data *user* adalah *fullname*, *npk*, *email*, *phone number*, dan *role* pada *textbox* yang sesuai. Pada saat pertama kali ditampilkan, *textbox* yang ada berisi data dari *user* yang sesuai dengan judul *textbox*. Terdapat tombol *save* memiliki fungsi untuk menyimpan data *user* yang diisi dalam *textbox* setelah melewati validasi yang diberikan oleh sistem, dan juga terdapat tombol *cancel* untuk membatalkan dan menghapus semua data yang sudah diinputkan oleh pengguna. Berikut adalah potongan kode yang digunakan untuk melakukan validasi baik pada saat melakukan pengubahan data ataupun penambahan data *user* :

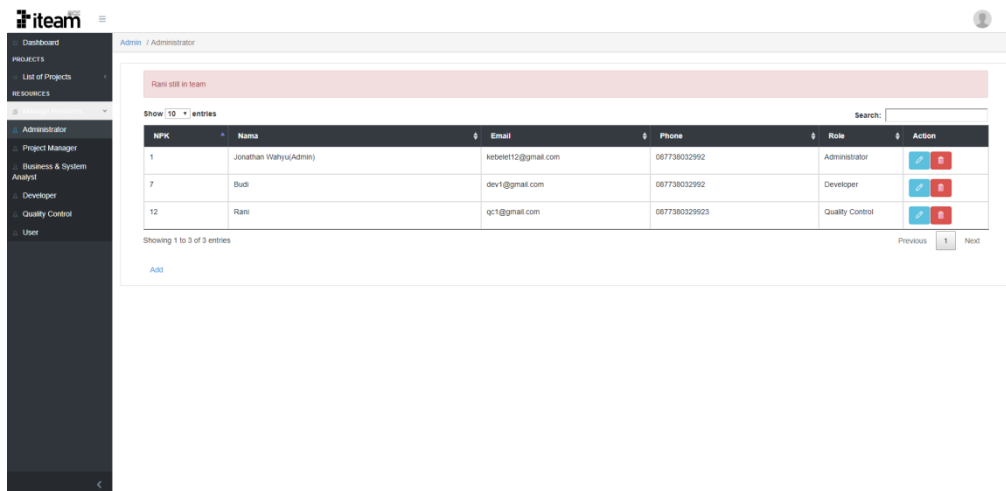
```

if($type == 'edit'){
    $this->validate($request, [
        'fullname' =>
        'required|string|min:6|max:255|regex:/^[pL\s]+$/u',
        'email' =>
        'required|string|email|max:255|unique:mst_users,email, '.$request->id.',id,deleted_at,NULL',
        'phone_number' => 'required|regex:/ (08) [0-9]{10}/',
        'password' => 'required|string|min:5'
    ]);
}
elseif($type == 'create'){
    $this->validate($request, [
        'fullname' =>
        'required|string|min:6|max:255|regex:/^[pL\s]+$/u',
        'npk' =>
        'required|string|regex:/^[0-9]+$/|unique:mst_users,npk,NULL,id,deleted_at,NULL',
        'email' =>
        'required|string|email|max:255|unique:mst_users,email,NULL,id,deleted_at,NULL',
        'phone_number' => 'required|regex:/ (08) [0-9]{10}/',
        'password' => 'required|string|min:5'
    ]);
}

```



## 5.2.7. Hapus User



**Gambar 5.7 Tampilan Hapus User**

Pada gambar 5.7 merupakan halaman yang sama dengan halaman tampilan *index user* setelah pengguna mencoba untuk menghapus data dari seorang *user*. Setiap kali pengguna melakukan penghapusan data dari *user*, sistem akan menampilkan sebuah *alert* yang memberikan informasi apakah data *user* berhasil dihapus atau tidak. Adapun data *user* yang tidak bisa dihapus adalah data dari *user* yang sedang mengerjakan sebuah *project*, sehingga akan muncul *alert* seperti gambar diatas. Berikut adalah potongan kode yang digunakan untuk melakukan pengecekan terhadap *user* yang akan dihapus :

```
$user = User::find($id);
$ccek = Team::where('id_user',
$id)->whereNull('deleted_at')->get();
if($ccek->isEmpty()){
    $user->delete();
    return redirect()->back()->with('workz', $user->fullname.'
    succesfully deleted');
}
return redirect()->back()->with('fail', $user->fullname.' still
in team');
```

## 5.2.8. Tampilan *Index Project*

Show 10 entries	Search:					
ID Project	User Client	Title	Class	Status	Created At	Action
1	Bagus(User)	Speakup	Gold	Approve	2019-05-17 08:49:45	[Action Icons]
2	Bagus(User)	PAKRW	Platinum	Pass	2019-05-17 08:49:48	[Action Icons]
3	Bagus(User)	CCRP	Silver	Pass	2019-05-17 08:49:50	[Action Icons]
4	Bagus(User)	BidMart	Uncategorized	Queue	2019-05-17 08:50:58	[Action Icons]
5	Bagus(User)	Tencent	Uncategorized	Queue	2019-05-17 08:51:00	[Action Icons]

Gambar 5. 8 Tampilan *Index Project*

Pada gambar 5.8 merupakan halaman yang menampilkan semua data dari *project*. Halaman ini akan menampilkan data *project* di dalam sebuah tabel dengan informasi tambahan warna yang bergantung pada status dari setiap *project*. Untuk *project* berstatus *queue* maka akan berwarna putih polos, untuk *project* berstatus *pass* akan berwarna hijau, untuk *project* berstatus *approve* dan sudah memiliki BSA akan berwarna biru tua, namun jika tidak memiliki BSA akan berwarna kuning, sedangkan untuk *project* berstatus *done* akan berwarna biru muda. Terdapat juga beberapa tombol yang berfungsi untuk mengubah data *project* ataupun mengelola data *project* sesuai dengan proses bisnis yang ada dengan cara mengecek siapa pengguna yang sedang mengakses, status dari sebuah *project*, dan akses mana yang diperbolehkan diberikan pada *project*.

## 5.2.9. *Pass, Ubah, atau Reject Project*

Show 10 entries	Search:					
ID Project	User Client	Title	Class	Status	Created At	Action
1	Bagus(User)	Speakup	Gold	Approve	2019-05-17 08:49:45	[Action Icons]
2	Bagus(User)	PAKRW	Platinum	Pass	2019-05-17 08:49:48	[Action Icons]
3	Bagus(User)	CCRP	Silver	Pass	2019-05-17 08:49:50	[Action Icons]
4	Bagus(User)	BidMart	Uncategorized	Queue	2019-05-17 08:50:58	[Action Icons]
5	Bagus(User)	Tencent	Uncategorized	Queue	2019-05-17 08:51:00	[Action Icons]

Title: BidMart

Description: Proyek untuk melajang barang - barang yang gagal di selesaikan kreditya

Class: Choose... Status: Queue

Request Date: 2019-05-17 08:50:58

User Client: Name: Bagus(User), Email: user1@gmail.com, Phone Number: 887730329923

[Save] [Cancel]

Gambar 5. 9 Tampilan *Pass, Ubah, atau Reject Project*

Pada gambar 5.9 merupakan halaman yang menampilkan tampilan *Index Project* dengan tambahan sebuah *card* untuk melakukan perubahan data *project*. Halaman ini dapat

diakses ketika pengguna ya berperan *admin* menekan tombol *Edit* atau yang memiliki *icon* pensil dan secara dinamis tampilan akan berubah seperti yang ditampilkan diatas. Selain itu, pada saat menekan tombol *Pass* atau *Reject* tampilan akan berubah seperti diatas jika nilai dari *class* belum ditentukan. Adapun masukan yang diperlukan untuk mengubah data *project* ialah *title*, *description*, *class* dan status. Terdapat tombol *save* yang memiliki fungsi untuk menyimpan data *project* yang diisi dalam *textbox* setelah melewati validasi yang diberikan oleh sistem, dan juga terdapat tombol *cancel* untuk membatalkan dan menghapus semua data yang sudah diinputkan oleh pengguna. Jika pengguna berniat hanya mengubah status *project* menjadi *pass* atau *reject*, dapat dilakukan dengan mudah yaitu dengan cara menekan tombol *Pass* atau *Reject*. Berikut adalah potongan kode yang digunakan untuk melakukan validasi terhadap perubahan data *project* :

```
$this->validate($request, [  
    'title' => 'required|string|max:255',  
    'description' => 'required|string|max:255',  
    'status' => 'required'
```

Berikut adalah potongan kode yang digunakan pada saat pengguna melakukan perubahan data *project* :

```
$project = Project::where('id', $request->id)->first();  
$id_project = $project->id;  
$project->title = $request->title;  
$project->description = $request->description;  
$project->id_class = $request->projectclass;  
$project->save();
```

Berikut adalah potongan kode yang digunakan pada saat pengguna mengubah status *project* dengan menekan tombol *Pass* atau *Reject*:

```

$project = Project::find($id);
$id_project = $project->id;
if(!empty($project->id_class)){
    if($type == 'pass'){
        $project->status = 'pass';
        $project->save();
        return back()->with('work', $project->title.'
        succesfully approved by Admin(pass)');
    }
    elseif($type == 'reject'){
        $project->status = 'reject';
        $project->save();
        return back()->with('work', $project->title.' has
        been rejected by Admin');
    }
}
}

```

### 5.2.10. Approve Project

ID Project	User Class	Title	Class	Status	Created At	Action
1	Bagas/User	Speakup	Gold	Approve	2019-05-17 08:49:45	Approve Pass
2	Bagas/User	PMAWV	Platinum	Pass	2019-05-17 08:49:48	Approve Pass
3	Bagas/User	CCRP	Silver	Pass	2019-05-17 08:49:50	Approve Pass

**Gambar 5. 10 Tampilan Approve Project**

Pada gambar 5.10 merupakan halaman yang menampilkan tampilan *Index Project* yang diakses oleh PM. Data yang ditampilkan ialah data dari semua *project* yang sudah memiliki status *pass* untuk dilanjutkan oleh PM untuk menangani sebuah *project* atau tidak. Terdapat tombol *Approve* yang digunakan oleh pengguna untuk menyetujui menangani *project* tersebut dan juga tombol tampil data *project* yang lebih detail dari yang ditampilkan di tabel dengan menggunakan sebuah *card* baru. Pada fungsi *approve* ini sistem tidak hanya mengubah status dari *project* menjadi *approve* namun juga memasukkan data jadwal pengerjaan *project* dan *phase* dari *project*. Selain itu sistem juga menyimpan data baru pada basis data *task* yang ditujukan pada PM, dan juga menyimpan data baru pada basis data *team* yang menunjukkan bahwa PM sudah menjadi anggota *team* untuk pengerjaan *project* tersebut. Berikut adalah potongan kode yang digunakan untuk melakukan *Approve project* yang dilakukan oleh PM :

```

elseif($type == 'approve'){
    $class =
ProjectClass::where('id',$project->id_class)->first();
    $project->date_start = Carbon::today();
    $project->date_end =
Carbon::today()->addDays($class->schedule);
    $project->actual_date_start = Carbon::today();
    $project->phase = 'assesment';
    $project->status = 'approve';
    $project->save();

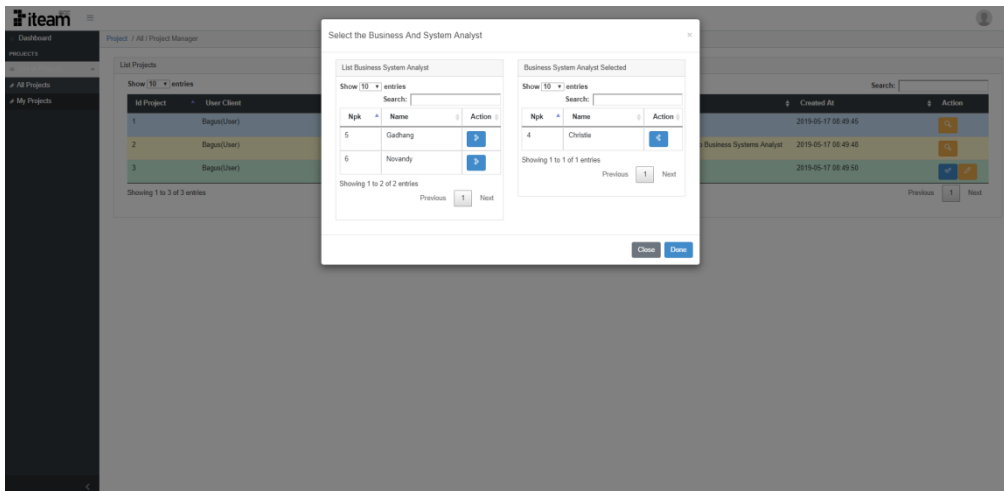
    Task::create([
        'id_project' => $project->id,
        'id_user' => Auth::user()->id,
        'task_name' => 'Assesment Project',
        'phase' => 'assesment',
        'date_start' => Carbon::today(),
        'date_end' => Carbon::today()->addDays(1),
        'actual_date_start' => Carbon::today(),
        'progress' => 0,
        'status' => 'process'
    ]);

    Team::create([
        'id_user' => Auth::user()->id,
        'id_project' => $project->id
    ]);

    session(['id_project'=>$id_project]);
    return
redirect('/administrator/project/index/all?type=approve');
}

```

### 5.2.11. Tampilan *Select BSA*



Gambar 5. 11 Tampilan *Select BSA*

Pada gambar 5.11 merupakan halaman yang menampilkan tampilan *My Project* yang diakses oleh PM setelah memilih sebuah *project* untuk ditangani. Pada gambar ditampilkan terdapat sebuah *modal* baru yang berisi 2 buah tabel dengan judul dan data yang berbeda. Tabel pertama berisi data *user* dengan peran BSA yang memiliki status *free*, sedang tabel kedua berisi data *user* dengan peran BSA yang telah dipilih sebagai *user* yang diminta untuk menangani *project*. Untuk memilih *user* yang ingin diminta menangani *project* pengguna hanya perlu menekan tombol dengan icon '>', jika pengguna ingin membatalkan salah satu pilihannya maka hanya perlu menekan tombol dengan icon '<'. Terdapat tombol *Done* yang memiliki fungsi untuk menyelesaikan proses pemilihan BSA dan menyimpan permintaan terhadap BSA langsung menjadi anggota *team* dan *Cancel* yang berfungsi untuk mengagalkan seluruh permintaan terhadap BSA. Berikut adalah potongan kode yang digunakan untuk melakukan fungsi *Done* yang dilakukan oleh PM :

```
$wishlist = DB::table('wishlist_teams')
->join('mst_users', 'mst_users.id', 'wishlist_teams.id_user')
->select(
    'mst_users.id',
    'mst_users.fullname',
    'mst_users.role'
)
->where('wishlist_teams.id_project',
$id_project)->where('mst_users.role', 'bsa')
->get();
```

```

foreach($wishlist as $teams)
{
    Task::create([
        'id_project' => $project->id,
        'id_user' => $teams->id,
        'task_name' => 'Assesment User by BSA '.$teams->fullname,
        'phase' => 'assesment',
        'date_start' => Carbon::today(),
        'date_end' => Carbon::today()->addDays(3),
        'actual_date_start' => Carbon::today(),
        'progress' => 0,
        'status' => 'process'
    ]);

    Team::create([
        'id_user' => $teams->id,
        'id_project' => $project->id
    ]);

    $delete =
    WishlistTeam::where('id_user',$teams->id)->where('id_project',
    $id_project)->first();
    $delete->delete();

    $user = User::find($teams->id);
    $user->status = 'onteam';
    $user->save();
}
}

```

Berikut adalah potongan kode yang digunakan untuk melakukan fungsi *Cancel* yang dilakukan oleh PM :

```

$wishlist = DB::table('wishlist_teams')
->join('mst_users', 'mst_users.id', 'wishlist_teams.id_user')
->select(
    'mst_users.id',
    'mst_users.fullname',
    'mst_users.role'
)
->where('wishlist_teams.id_project',
$id_project)->where('mst_users.role','bsa')->get();

```

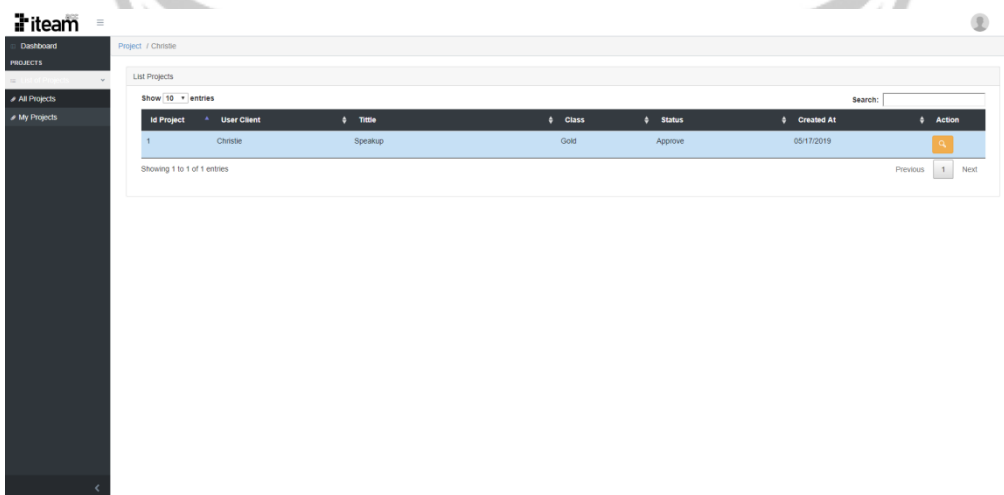
```

foreach($wishlist as $teams)
{
    $temp = $teams->id;
    $delete =
WishlistTeam::where('id_user',$teams->id)->where('id_project',
$id_project)->first();
    $delete->delete();

    $cek = WishlistTeam::where('id_user', $temp)->get();
    $cekteam = Team::where('id_user', $temp)->get();
    if($cekteam->isEmpty()){
        if($cek->isEmpty()){
            $user = User::find($temp);
            $user->status = 'free';
            $user->save();
        }
        else
        {
            $user = User::find($temp);
            $user->status = 'booked';
            $user->save();
        }
    }
    else
    {
        $user = User::find($temp);
        $user->status = 'onteam';
        $user->save();
    }
}
}

```

### 5.2.12. Tampilan *My Project*

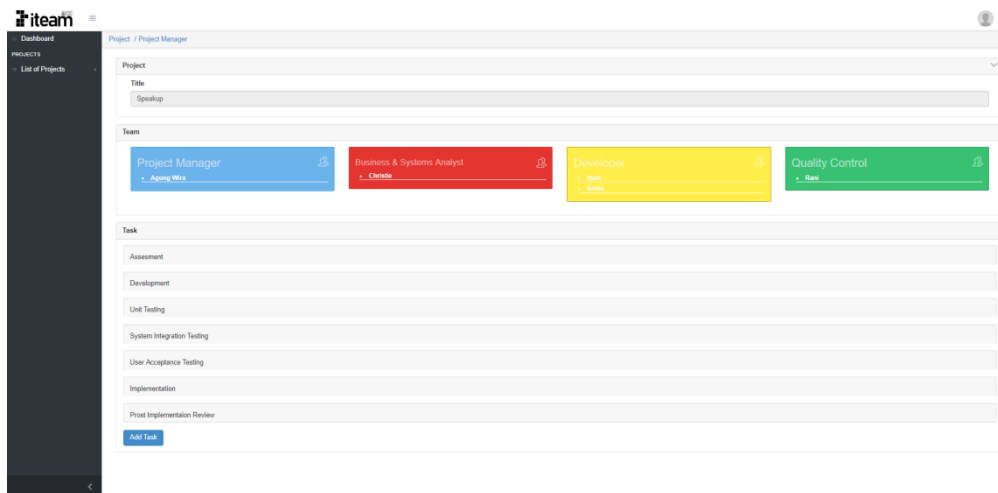


**Gambar 5. 12 Tampilan *My Project***

Pada gambar 5.12 merupakan halaman yang menampilkan semua data dari *project* yang dikerjakan oleh pengguna yang sedang mengakses sistem. Halaman ini akan menampilkan data *project* di dalam sebuah tabel. Terdapat juga tombol yang berfungsi untuk melihat *detail* dari *project* tersebut.



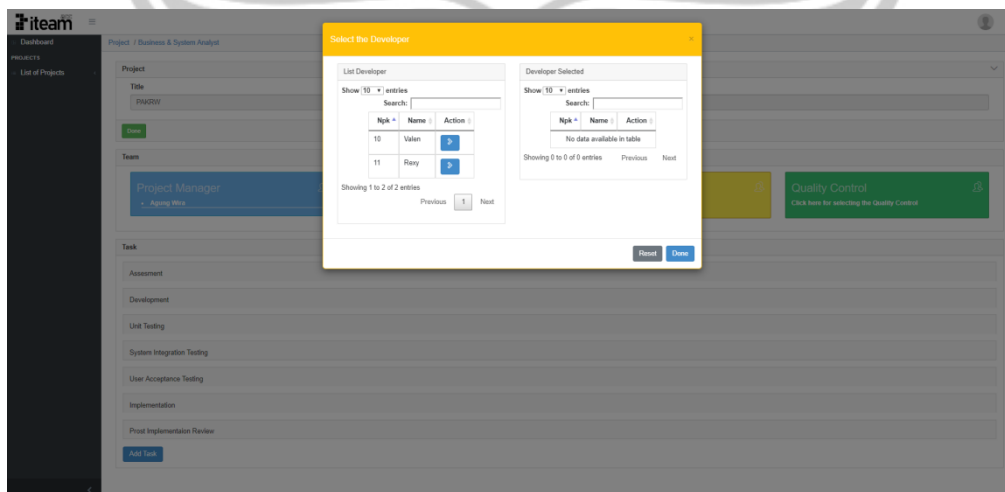
### 5.2.13. Tampilan *Detail Project*



Gambar 5. 13 Tampilan *Detail Project*

Pada gambar 5.13 merupakan halaman yang menampilkan tampilan *Detail Project* yang dapat diakses oleh semua pengguna yang terdaftar sebagai anggota *team* dari *project* tersebut. Terdapat banyak data dari berbagai basis data yang ditampilkan di halaman ini. Selain data *project*, terdapat juga data *team*, dan data *team*. Melalui halaman ini pengguna dapat mengelola semua ada yang penting dalam pengerjaan *project* sesuai dengan perannya masing - masing. Untuk pengaksesan anggota *team* dengan peran BSA hanya dapat diakses oleh PM, anggota *team* dengan peran Developer dan QC hanya bisa diakses oleh BSA. Untuk pengaksesan tambah, ubah dan hapus *task* hanya bisa dilakukan oleh BSA, namun untuk fungsi *start* dan *done* hanya bisa diakses oleh pengguna yang memiliki *task* tersebut.

### 5.2.14. Tampilan *Select Member*



Gambar 5. 14 Tampilan *Select Member*

Pada gambar 5.14 merupakan halaman yang menampilkan tampilan *Detail Project* pada saat anggota *team* dengan peran Dev dipilih, sehingga akan menampilkan sebuah modal baru. Dalam modal ini ditampilkan 2 buah tabel dengan data yang berbeda. Tabel pertama

berisi data *user* dengan peran Dev yang memiliki status *free*, sedangkan pada tabel kedua berisi data *user* dengan peran Dev yang telah tersimpan sebagai *user* yang diminta untuk menjadi anggota *team*. Untuk memilih *user* yang ingin diminta menangani *project* pengguna hanya perlu menekan tombol dengan icon '>', jika pengguna ingin mebatalkan salah satu pilihannya maka hanya perlu menekan tombol dengan icon '<'. Terdapat tombol *Done* yang memiliki fungsi untuk menyelesaikan proses pemilihan Dev dan menyimpan permintaan terhadap Dev sebagai *wishlist* dan *Cancel* yang berfungsi untuk mengagalkan seluruh permintaan terhadap Dev. Penjelasan di atas berlaku juga terhadap semua anggota *team* dengan peran apapun pada halaman *detail project*. Berikut adalah potongan kode yang digunakan untuk mengambil data pada tabel pertama :

```
$idUser = DB::table('mst_users')
->leftjoin('wishlist_teams', 'wishlist_teams.id_user',
'mst_users.id')
->leftjoin('mst_teams', 'mst_teams.id_user', 'mst_users.id')
->select(
'mst_users.id'
)
->where('mst_teams.id_project', $id_project)
->orWhere('wishlist_teams.id_project', $id_project)
->distinct();
$model = DB::table('mst_users')
->select(
'mst_users.id',
'mst_users.npk',
'mst_users.fullname'
);
```

```

if($type == 'bsa'){
    $model->whereNotIn('mst_users.id', $iduser)
    ->where('mst_users.role', 'bsa')
    ->whereNull('mst_users.deleted_at');
}
elseif($type == 'dev'){
    $model->whereNotIn('mst_users.id', $iduser)
    ->where('mst_users.role', 'dev')
    ->where('mst_users.status', '!=', 'onteam')
    ->whereNull('mst_users.deleted_at');
}
elseif($type == 'qc'){
    $model->whereNotIn('mst_users.id', $iduser)
    ->where('mst_users.role', 'qc')
    ->where('mst_users.status', '!=', 'onteam')
    ->whereNull('mst_users.deleted_at');
}

return DataTables::of($model)
->addColumn('action', function($model) use ($id_project){
return
href="'.url('/administrator/project').'/'.$id_project.'/select/
user/'.$model->id.'" class="btn-add btn btn-primary btn-md"
style="margin-right:1px"><i class="icons cui-chevron-right"
style="color:white"></i> </a>';
}))
->rawColumns(['action'])
->toJson();

```

Berikut adalah potongan kode yang digunakan untuk mengambil data pada tabel

kedua:

```

$model = DB::table('mst_users')
->leftjoin('wishlist_teams', 'wishlist_teams.id_user',
'mst_users.id')
->select(
'mst_users.id',
'mst_users.npk',
'mst_users.fullname'
);

```

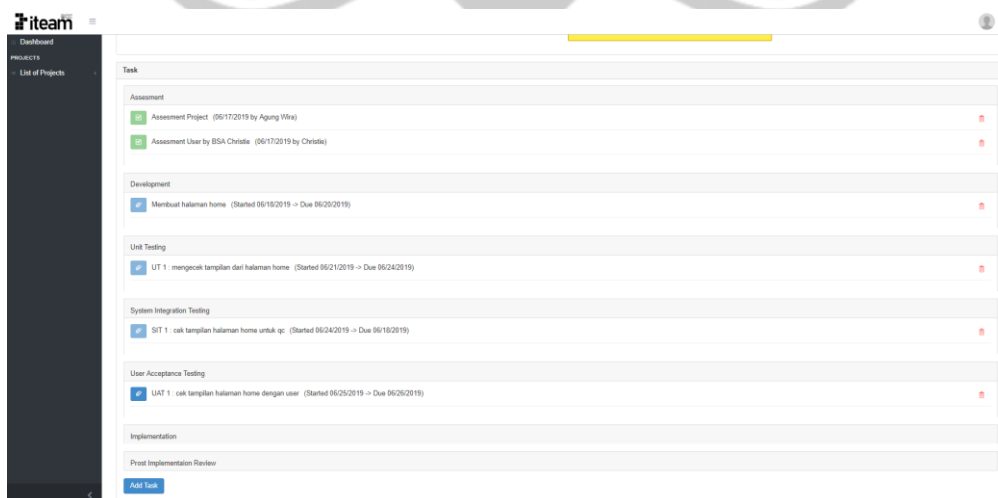
```

if($type == 'bsa')
{
    $model->where('wishlist_teams.id_project', $id_project)
    ->where('mst_users.role','bsa')
    ->whereNull('mst_users.deleted_at');
}
elseif($type == 'dev')
{
    $model->where('wishlist_teams.id_project', $id_project)
    ->where('mst_users.role','dev')
    ->whereNull('mst_users.deleted_at');
}
elseif($type == 'qc')
{
    $model->where('wishlist_teams.id_project', $id_project)
    ->where('mst_users.role','qc')
    ->whereNull('mst_users.deleted_at');
}

return DataTables::of($model)
->addColumn('action', function($model) use ($id_project){
return
href="'.url('/administrator/project')..'/'.$id_project.'/cancel/
user/'.$model->id.'" class="btn-add btn btn-primary btn-md"
style="margin-right:1px"><i class="icons cui-chevron-left"
style="color:white"></i> </a>';
})
->rawColumns(['action'])
->toJson();

```

### 5.2.15. Tampilan *Index Task*

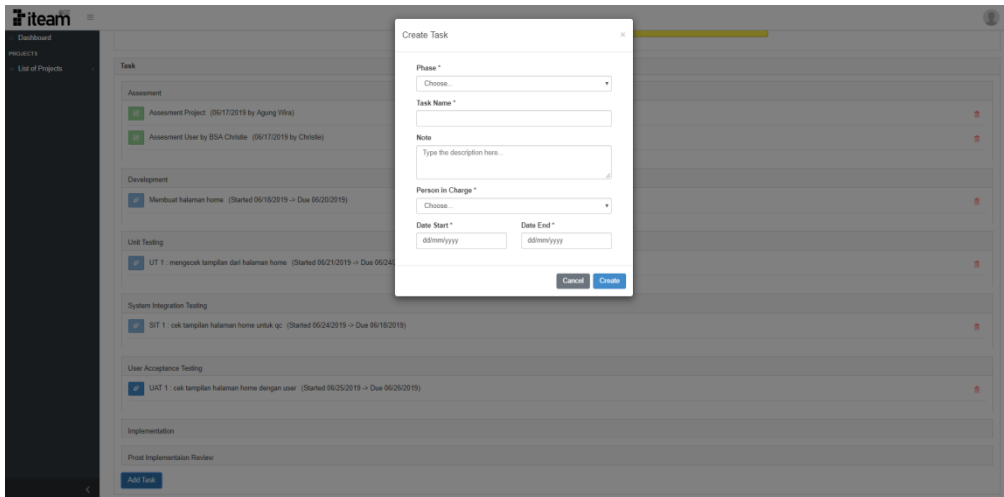


**Gambar 5.15 Tampilan *Index Task***

Pada gambar 5.15 merupakan halaman yang menampilkan tampilan *Detail Project* dengan berfokus pada bagian *task*. Setiap data *task* yang berada didalam *project* ini

ditampilkan dalam bentuk *list*. Pengguna perlu menekan judul dari *list task* yang ada untuk melihat keseluruhan dari *task*. Data *task* dipisahkan berdasarkan *phase* dari *task* tersebut. Terdapat beberapa tombol seperti tambah *task*, *start task*, *done task*, hapus *task* dan sebuah *link* yang berada pada judul *task*. Disebelah kanan judul *task* terdapat informasi mengenai jadwal pengerjaan dari *task* tersebut dan siapa *user* yang bertanggung jawab pada *task* tersebut. Penulisan informasi tersebut akan berubah warna menjadi merah jika hari saat mengakses sistem sudah lebih dari tanggal terakhir pengerjaan *task*.

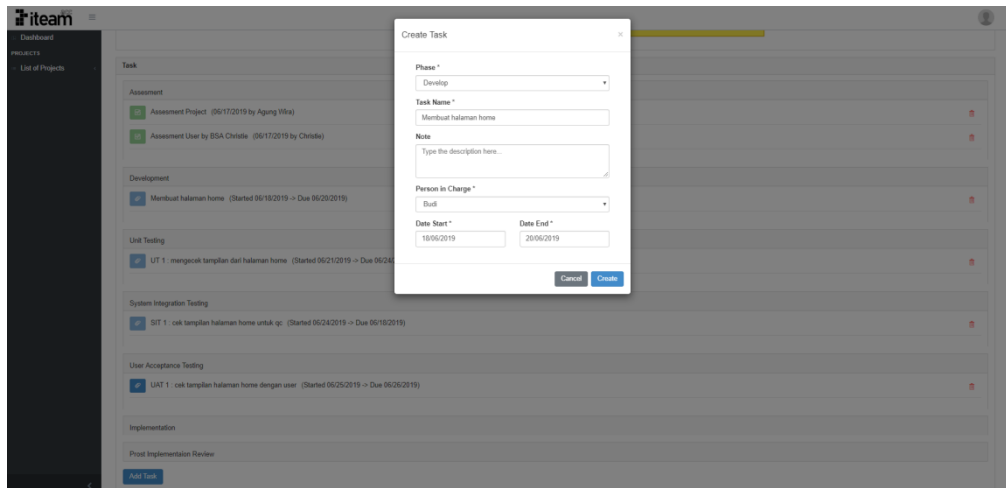
### 5.2.16. Tambah *Task*



**Gambar 5. 16 Tampilan Tambah *Task***

Pada gambar 5.16 merupakan halaman yang sama dengan halaman tampilan *detail project* dengan tambahan sebuah *modal* baru untuk menambah data *task*. Halaman ini dapat diakses ketika pengguna menekan tombol *Add* dibagian bawah *task* dan *modal* seperti yang ditampilkan diatas. Adapun masukan yang diperlukan untuk menambah data *task* adalah *phase*, *task name*, *note*, *person in charge*, *date start*, dan *date end* pada *textbox* yang sesuai. Pada saat pertama kali ditampilkan, *textbox* yang ada tidak berisi data apapun. Terdapat tombol *save* memiliki fungsi untuk menyimpan data *user* yang diisi dalam *textbox* setelah melewati validasi yang diberikan oleh sistem, dan juga terdapat tombol *cancel* untuk membatalkan dan menghapus semua data yang sudah diinputkan oleh pengguna.

## 5.2.17. Ubah Task

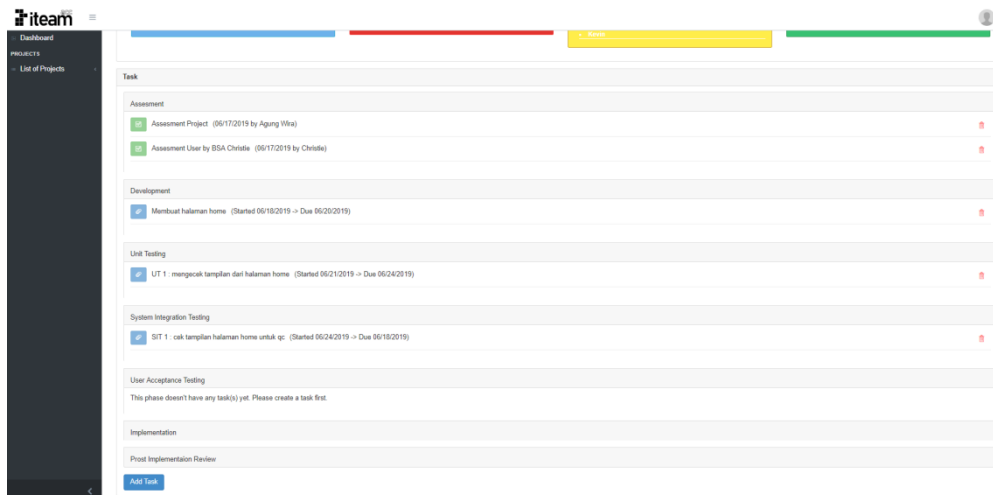


**Gambar 5. 17 Tampilan Ubah Task**

Pada gambar 5.17 merupakan halaman yang sama dengan halaman tampilan *detail project* dengan tambahan sebuah *modal* baru untuk menambah data *task*. Halaman ini dapat diakses ketika pengguna menekan *link* judul *task* dan *modal* seperti yang ditampilkan diatas. Adapun masukan yang diperlukan untuk menambah data *task* adalah *phase*, *task name*, *note*, *person in charge*, *date start*, dan *date end* pada *textbox* yang sesuai. Pada saat pertama kali ditampilkan, *textbox* yang ada berisi data *project*. Terdapat tombol *save* memiliki fungsi untuk menyimpan data *user* yang diisi dalam *textbox* setelah melewati validasi yang diberikan oleh sistem, dan juga terdapat tombol *cancel* untuk membatalkan dan menghapus semua data yang sudah diinputkan oleh pengguna. Berikut adalah potongan kode yang digunakan untuk melakukan validasi baik pada saat melakukan pengubahan data ataupun penambahan data *task* :

```
$this->validate($request, [
    'user_pic' => 'required|integer',
    'task_name' => 'required|string|max:255',
    'phase' => 'required',
    'date_start' => 'required',
    'date_end' => 'required'
]);
```

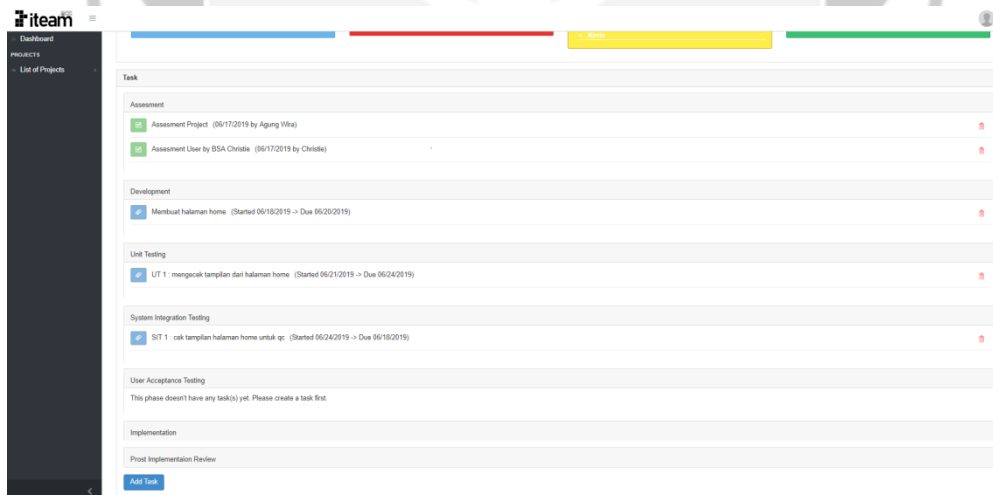
## 5.2.18. Hapus Task



Gambar 5. 18 Tampilan Hapus Task

Pada gambar 5.18 merupakan halaman yang sama dengan halaman tampilan *detail project*. Penghapusan pada *task* tidak memiliki pengecekan melalui sistem dan yang memiliki akses untuk melakukan penghapusan data *task* hanya *user* yang memiliki peran BSA. Setiap pengguna menghapus *task*, sistem akan kembali ke halaman yang sama dengan data terbaru.

## 5.2.19. Start atau Done Task



Gambar 5. 19 Tampilan Start atau Done Task

Pada gambar 5.19 merupakan halaman yang sama dengan halaman tampilan *detail project* yang akan berfokus pada tombol *start* ataupun *done* dari sebuah *task*. Pengguna yang bisa mengakses fungsi tersebut hanyalah pengguna yang tersimpan bahwa menjadi *person in charge* dari *task* tersebut. Tombol *start* memiliki warna biru dan memiliki *icon paperclip* sedangkan untuk tombol *done* memiliki warna hijau dan memiliki *icon check*. Fungsi *start* berguna pada saat *user* yang diberi *task* ingin memulai pengerjaannya, dengan menekan tombol *start* sistem akan menyimpan tanggal *user* menekan tombol *start* sebagai data *actual*

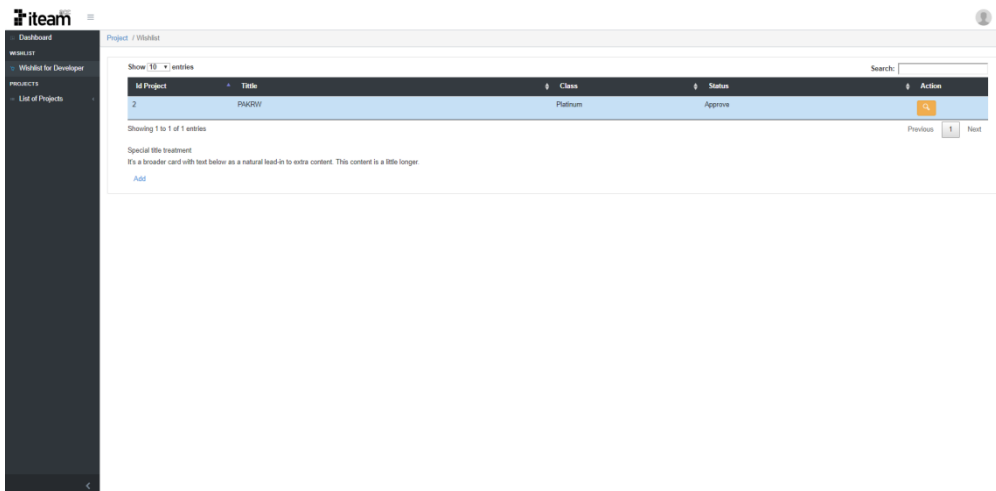
*date start* atau tanggal mulai sesungguhnya. Sedangkan fungsi *done* berguna apabila saat *user* telah selesai menyelesaikan *task* yang diberikan padanya, dengan menekan tombol *done* sistem akan menyimpan tanggal *user* menekan tombol *done* sebagai data *actual date end* atau tanggal selesai sesungguhnya. Berikut adalah potongan kode yang digunakan untuk pengelolaan *task* ini :

```
$task = Task::where('id', $id_task)->first();
if($type == 'start')
{
    $task->actual_date_start = Carbon::today();
    $task->status = 'process';
    $task->save();

    $project = Project::find($task->id_project);
    $project->phase = $task->phase;
    $project->save();
    return back();
}
elseif($type == 'done')
{
    $task->actual_date_end = Carbon::today();
    $task->status = 'done';
    $task->update_by = Auth::user()->fullname;
    $task->save();
    return back();
}
elseif($type == 'detail')
{ return json_encode($task); }
elseif($type == 'delete')
{
    $task->delete();
    return back();
}
```



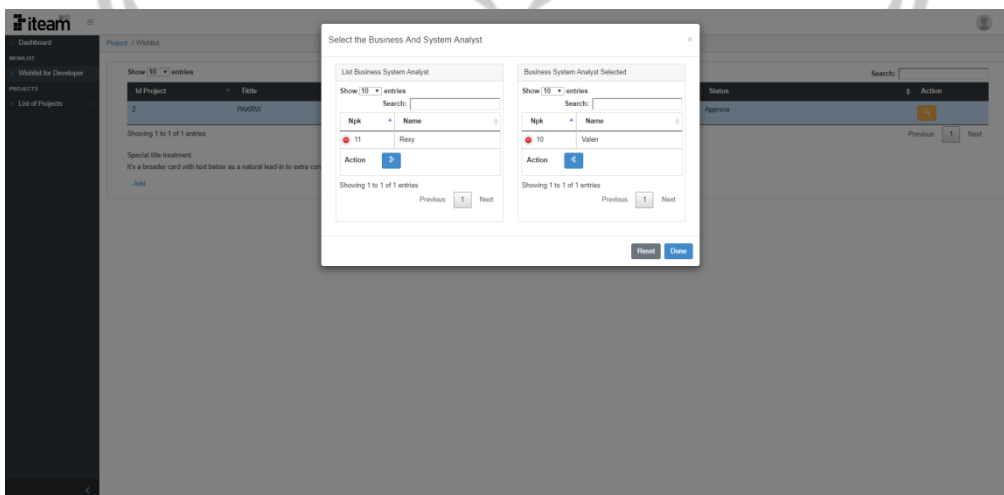
## 5.2.20. Tampilan *Index Wishlist*



Gambar 5. 20 Tampilan *Index Wishlist*

Pada gambar 5.20 merupakan halaman yang menampilkan semua data *project* yang memiliki permintaan *user* sebagai anggota *team* - nya. Pengguna yang dapat mengakses halaman ini hanya yang memiliki peran Dev dan QC serta memiliki data status *is\_admin* = 1. Terdapat tombol *view* untuk melihat permintaan dari *project* tersebut. Halaman ini digunakan untuk memenuhi proses bisnis dimana kepala dari *user* dengan peran Dev dan QC menentukan disetujui tidaknya pegawai dengan peran yang sama untuk menjadi anggota *team* dari sebuah *project*.

## 5.2.21. *Approve atau Reject Wishlist*



Gambar 5. 21 Tampilan *Approve atau Reject Wishlist*

Pada gambar 5.21 merupakan halaman yang menampilkan tampilan *Index Wishlist* pada saat tombol *view* ditekan sehingga akan menampilkan sebuah modal baru. Dalam modal ini ditampilkan 2 buah tabel dengan data yang berbeda. Tabel pertama berisi data *user* dengan peran yang sama dengan pengguna yang memiliki status *free*, sedangkan pada tabel kedua berisi data *user* dengan peran yang sama yang telah tersimpan sebagai *user* yang diminta

untuk menjadi anggota *team*. Untuk mengelola *wishlist*, pengguna dapat melakukannya dengan menekan tombol dengan *icon* '>' jika ingin memasukkan seseorang sebagai *team*, jika pengguna ingin membatalkan salah satu pilihannya maka hanya perlu menekan tombol dengan *icon* '< '.

Terdapat tombol *Done* yang memiliki fungsi *approve* untuk menyelesaikan proses permintaan anggota *team* dan menyimpan permintaan *user* didalam sebuah *team* dalam pengerjaan *project* tersebut dan *Reset* yang berfungsi *reject* untuk menolak seluruh permintaan. Setiap melakukan *approve* sistem akan melakukan pengecekan terhadap basis data *wishlist*, jika masih ditemukan data *user* yang sama untuk *project* lain maka sistem akan menghapusnya dan mengubah status dari *user*. Berikut adalah potongan kode yang digunakan untuk melakukan pengelolaan *wishlist* tersebut :



```

$project = Project::where('id', $id_project)->first();
elseif($type == 'insertdev')
{
    $wishlist = DB::table('wishlist_teams')
    ->join('mst_users', 'mst_users.id',
'wishlist_teams.id_user')
    ->select(
        'mst_users.id',
        'mst_users.fullname',
        'mst_users.role'
    )
    ->where('wishlist_teams.id_project',
$id_project)->where('mst_users.role','dev')
    ->get();
    foreach($wishlist as $teams)
    {
        Team::create([
            'id_user' => $teams->id,
            'id_project' => $project->id
        ]);

        $delete =
WishlistTeam::where('id_user',$teams->id)->where('id_proj
ect', $id_project)->first();
        $delete->delete();
        $wishlist = DB::table('wishlist_teams')
        ->join('mst_users', 'mst_users.id',
'wishlist_teams.id_user')
        ->select(
            'mst_users.id',
            'mst_users.role'
        )
        ->where('wishlist_teams.id_user',
$teams->id)->delete();

        $user = User::find($teams->id);
        $user->status = 'onteam';
        $user->save();
    }
}

```

```

elseif($type == 'canceldev')
{
    $wishlist = DB::table('wishlist_teams')
    ->join('mst_users', 'mst_users.id',
    'wishlist_teams.id_user')
    ->select(
        'mst_users.id',
        'mst_users.fullname',
        'mst_users.role'
    )
    ->where('wishlist_teams.id_project', $id_project)
    ->where('mst_users.role', 'dev')
    ->get();

    foreach($wishlist as $teams)
    {
        $delete =
        WishlistTeam::where('id_user', $teams->id)->where('id_project',
        $id_project)->first();

        $cek = WishlistTeam::where('id_user',
        $teams->id)->where('id_project', '!=', $id_project)->get();

        if($cek->isEmpty())
        {
            $user = User::find($teams->id);
            $user->status = 'free';
            $user->save();
        }
        else
        {
            $user = User::find($teams->id);
            $user->status = 'booked';
            $user->save();
        }
        $delete->delete();
    }
}

```

### 5.3. Hasil Pengujian

Identifikasi	Deskripsi	Prosedur Pengujian	Masukan	Keluaran Yang Diharapkan	Kriteria evaluasi hasil	Hasil yang didapat	Kesimpulan
<b>Aplikasi Web</b>							
<i>Register</i>	Pengujian terhadap fungsi <i>register</i>	<ul style="list-style-type: none"> <li>- Masukkan <i>fullname</i>, <i>npk</i>, <i>email</i>, <i>phone number</i>, <i>password</i> dan <i>confirm password</i></li> <li>- Tekan <i>enter</i> atau tombol <i>Create Account</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Fullname</i> : “Bagus (User)”</li> <li>- <i>Email</i> : “user1@gmail.com”</li> <li>- <i>Npk</i> : 16</li> <li>- <i>Phone Number</i> : “0877380329923”</li> <li>- <i>Password</i> : “admin1”</li> <li>- <i>Confirm Password</i> : “admin1”</li> </ul>	<ul style="list-style-type: none"> <li>- Data berhasil disimpan</li> <li>- Masuk ke dalam sistem dan menampilkan halaman <i>dashboard</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data berhasil disimpan</li> <li>- Masuk ke dalam sistem dan menampilkan halaman <i>dashboard</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data berhasil disimpan</li> <li>- Masuk ke dalam sistem dan menampilkan halaman <i>dashboard</i></li> </ul>	Handal
<i>Login</i>	Pengujian terhadap fungsi <i>login</i>	<ul style="list-style-type: none"> <li>- Masukkan <i>email</i> dan <i>password</i></li> <li>- Tekan <i>enter</i> atau tekan tombol <i>login</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Email</i> “kebelet12@gmail.com”</li> <li>- <i>Password</i> “admin1”</li> </ul>	<ul style="list-style-type: none"> <li>- Masuk ke dalam sistem dan menampilkan halaman <i>dashboard</i></li> </ul>	<ul style="list-style-type: none"> <li>- Masuk ke dalam sistem dan menampilkan halaman <i>dashboard</i></li> </ul>	<ul style="list-style-type: none"> <li>- Masuk ke dalam sistem dan menampilkan halaman <i>dashboard</i></li> </ul>	Handal
<i>Tampil Dashboard</i>	Pengujian terhadap fungsi	<ul style="list-style-type: none"> <li>- Setelah melakukan fungsi <i>Login</i> atau</li> </ul>	<ul style="list-style-type: none"> <li>- Menekan <i>menu Dashboard</i></li> </ul>	<ul style="list-style-type: none"> <li>- Menampilkan data jumlah <i>user</i> sesuai</li> </ul>	<ul style="list-style-type: none"> <li>- Menampilkan data jumlah <i>user</i> sesuai</li> </ul>	<ul style="list-style-type: none"> <li>- Menampilkan data jumlah <i>user</i> sesuai</li> </ul>	Handal

	Tampil Dashboard	memilih menu Dashboard		perannya yang sedang <i>free</i> - Menampilkan seluruh data <i>user</i>	perannya yang sedang <i>free</i> - Menampilkan seluruh data <i>user</i>	perannya yang sedang <i>free</i> - Menampilkan seluruh data <i>user</i>	
Tampil Data User	Pengujian terhadap fungsi Tampil Data User	- Memilih menu <i>Resource</i> dan pilih <i>sub-menu</i> sesuai dengan peran dari <i>user</i>	- Menekan <i>sub-menu Developer</i>	- Menampilkan data <i>user</i> dengan peran <i>developer</i> beserta dengan statusnya	- Menampilkan data <i>user</i> dengan peran <i>developer</i> beserta dengan statusnya	- Menampilkan data <i>user</i> dengan peran <i>developer</i> beserta dengan statusnya	Handal
Tambah Data User	Pengujian terhadap fungsi Tambah Data User	- Memilih menu <i>Resource</i> dan pilih <i>sub-menu</i> sesuai dengan peran dari <i>user</i> - Tekan tombol tambah data <i>user</i> - Masukkan <i>fullname</i> , <i>npk</i> , <i>email</i> , <i>phone number</i> , <i>role</i> , dan <i>password</i> - Tekan tombol <i>Save</i>	- <i>Fullname</i> : "Christie" - <i>Email</i> : "bsa1@yahoo.com" - <i>Npk</i> : 4 - <i>Phone Number</i> : "0877380329923" - <i>Role</i> : "bsa" - <i>Password</i> : "admin1"	- Data <i>user</i> berhasil disimpan - Data <i>user</i> yang baru berhasil ditampilkan di dalam tabel	- Data <i>user</i> berhasil disimpan - Data <i>user</i> yang baru berhasil ditampilkan di dalam tab	- Data <i>user</i> berhasil disimpan - Data <i>user</i> yang baru berhasil ditampilkan di dalam tab	Handal
Ubah Data User	Pengujian terhadap	- Memilih menu <i>Resource</i> dan	- <i>Fullname</i> : "Christie"	- Data <i>user</i> berhasil disimpan	- Data <i>user</i> berhasil disimpan	- Data <i>user</i> berhasil disimpan	Handal

	fungsi Ubah Data User	<p>pilih <i>sub-menu</i> sesuai dengan peran dari <i>user</i></p> <ul style="list-style-type: none"> <li>- Tekan tombol ubah data <i>user</i></li> <li>- Sistem menampilkan data <i>user</i> yang dipilih</li> <li>- Masukkan <i>fullname</i>, <i>npk</i>, <i>email</i>, <i>phone number</i>, <i>role</i>, dan <i>password</i></li> <li>- Tekan tombol <i>Save</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Email</i> : "bsa1@gmail.com"</li> <li>- <i>Npk</i> : 4</li> <li>- <i>Phone Number</i> : "0877380329923"</li> <li>- <i>Role</i> : "bsa"</li> <li>- <i>Password</i> : "admin1"</li> </ul>	- Data <i>user</i> yang baru berhasil ditampilkan di dalam tabel	- Data <i>user</i> yang baru berhasil ditampilkan di dalam tabel	- Data <i>user</i> yang baru berhasil ditampilkan di dalam tabel	
Hapus Data User	Pengujian terhadap fungsi Hapus Data Use	<ul style="list-style-type: none"> <li>- Memilih <i>menu Resource</i> dan pilih <i>sub-menu</i> sesuai dengan peran dari <i>user</i></li> <li>- Tekan tombol ubah data <i>user</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Id user</i> : "11"</li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> berhasil dihapus</li> <li>- Data <i>user</i> yang baru berhasil ditampilkan di dalam tabel</li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> berhasil dihapus</li> <li>- Data <i>user</i> yang baru berhasil ditampilkan di dalam tabel</li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> berhasil dihapus</li> <li>- Data <i>user</i> yang baru berhasil ditampilkan di dalam tabel</li> </ul>	Handal

		<ul style="list-style-type: none"> <li>- Sistem menghapus data <i>user</i> yang dipilih</li> </ul>					
Tampil Data Project	Pengujian terhadap fungsi Tampil Data Project	<ul style="list-style-type: none"> <li>- Memilih menu <i>Resource</i> dan pilih <i>sub-menu All Project</i></li> </ul>	<ul style="list-style-type: none"> <li>- Menekan <i>sub-menu All Project</i></li> </ul>	<ul style="list-style-type: none"> <li>- Menampilkan seluruh data <i>project</i> dengan beserta dengan statusnya dan tombol yang berisi fungsi untuk dilakukan kepada <i>project</i> tersebut</li> </ul>	<ul style="list-style-type: none"> <li>- Menampilkan seluruh data <i>project</i> dengan beserta dengan statusnya dan tombol yang berisi fungsi untuk dilakukan kepada <i>project</i> tersebut</li> </ul>	<ul style="list-style-type: none"> <li>- Menampilkan seluruh data <i>project</i> dengan beserta dengan statusnya dan tombol yang berisi fungsi untuk dilakukan kepada <i>project</i> tersebut</li> </ul>	Handal
Pass Project	Pengujian terhadap fungsi Pass Project	<ul style="list-style-type: none"> <li>- Menekan tombol dengan fungsi <i>Pass</i> pada salah satu <i>project</i></li> </ul>	<ul style="list-style-type: none"> <li>- Menekan tombol fungsi <i>Pass</i> pada <i>project</i> dengan <i>id project : 1</i></li> </ul>	<ul style="list-style-type: none"> <li>- Sistem mengubah status dari <i>project</i> menjadi <i>pass</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	<ul style="list-style-type: none"> <li>- Sistem mengubah status dari <i>project</i> menjadi <i>pass</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	<ul style="list-style-type: none"> <li>- Sistem mengubah status dari <i>project</i> menjadi <i>pass</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	Handal
Approve Project	Pengujian terhadap fungsi Approve Project	<ul style="list-style-type: none"> <li>- Menekan tombol dengan fungsi <i>Approve</i> pada salah satu <i>project</i></li> </ul>	<ul style="list-style-type: none"> <li>- Menekan tombol fungsi <i>Approve</i> pada <i>project</i> dengan <i>id project : 1</i></li> </ul>	<ul style="list-style-type: none"> <li>- Sistem mengubah status dari <i>project</i> menjadi <i>approve</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	<ul style="list-style-type: none"> <li>- Sistem mengubah status dari <i>project</i> menjadi <i>approve</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	<ul style="list-style-type: none"> <li>- Sistem mengubah status dari <i>project</i> menjadi <i>approve</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	Handal



<i>Reject Project</i>	Pengujian terhadap fungsi <i>Reject Project</i>	<ul style="list-style-type: none"> <li>- Menekan tombol dengan fungsi <i>Reject</i> pada salah satu <i>project</i></li> </ul>	<ul style="list-style-type: none"> <li>- Menekan tombol fungsi <i>Reject</i> pada <i>project</i> dengan <i>id project : 5</i></li> </ul>	<ul style="list-style-type: none"> <li>- Sistem mengubah status dari <i>project</i> menjadi <i>reject</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	<ul style="list-style-type: none"> <li>- Sistem mengubah status dari <i>project</i> menjadi <i>reject</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	<ul style="list-style-type: none"> <li>- Sistem mengubah status dari <i>project</i> menjadi <i>reject</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	Handal
<i>Ubah Project</i>	Pengujian terhadap fungsi <i>Ubah Project</i>	<ul style="list-style-type: none"> <li>- Menekan tombol dengan fungsi <i>Ubah</i> pada salah satu <i>project</i></li> <li>- Sistem akan menampilkan beberapa data dari <i>project</i></li> <li>- Masukkan <i>title, description, id class</i>, dan status</li> <li>- Tekan tombol <i>Save</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Id Project : "5"</i></li> <li>- <i>Title : "Tencent"</i></li> <li>- <i>Description : "Proyek untuk memainkan gitar hero di mobile"</i></li> <li>- <i>Id Class : "1"</i></li> <li>- Status : "<i>queue</i>"</li> </ul>	<ul style="list-style-type: none"> <li>- Sistem menyimpan perubahan dari data <i>project</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	<ul style="list-style-type: none"> <li>- Sistem menyimpan perubahan dari data <i>project</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	<ul style="list-style-type: none"> <li>- Sistem menyimpan perubahan dari data <i>project</i></li> <li>- Menampilkan seluruh data <i>project</i> terbaru</li> </ul>	Handal
<i>Detail Project</i>	Pengujian terhadap fungsi <i>Detail Project</i>	<ul style="list-style-type: none"> <li>- Menekan tombol dengan fungsi <i>Detail</i> pada salah satu <i>project</i></li> <li>- Sistem akan menampilkan</li> </ul>	<ul style="list-style-type: none"> <li>- <i>Id project : "5"</i></li> </ul>	<ul style="list-style-type: none"> <li>- Tampilkan data <i>title, description, id class</i>, dan status dari <i>project</i> dengan <i>id = 5</i></li> </ul>	<ul style="list-style-type: none"> <li>- Tampilkan data <i>title, description, id class</i>, dan status dari <i>project</i> dengan <i>id = 5</i></li> </ul>	<ul style="list-style-type: none"> <li>- Tampilkan data <i>title, description, id class</i>, dan status dari <i>project</i> dengan <i>id = 5</i></li> </ul>	Handal

		beberapa data dari <i>project</i> seperti <i>title</i> , <i>description</i> , <i>id class</i> , dan status					
Tampil Detail Project	Pengujian terhadap fungsi Tampil Detail Project	<ul style="list-style-type: none"> <li>- Menekan tombol dengan fungsi Tampil Detail Project</li> <li>- Sistem akan menampilkan seluruh data yang berhubungan dengan <i>project</i>, seperti data <i>Team</i> dan <i>Task</i></li> </ul>	- <i>Id project</i> : "5"	<ul style="list-style-type: none"> <li>- Tampilkan data <i>title</i>, <i>description</i>, <i>id class</i>, dan status dari <i>project</i> dengan <i>id</i> = 5</li> <li>- Tampilkan data <i>Team</i> dengan <i>id project</i> = 5</li> <li>- Tampilkan data <i>Task</i> dengan <i>id project</i> = 5</li> </ul>	<ul style="list-style-type: none"> <li>- Tampilkan data <i>title</i>, <i>description</i>, <i>id class</i>, dan status dari <i>project</i> dengan <i>id</i> = 5</li> <li>- Tampilkan data <i>Team</i> dengan <i>id project</i> = 5</li> <li>- Tampilkan data <i>Task</i> dengan <i>id project</i> = 5</li> </ul>	<ul style="list-style-type: none"> <li>- Tampilkan data <i>title</i>, <i>description</i>, <i>id class</i>, dan status dari <i>project</i> dengan <i>id</i> = 5</li> <li>- Tampilkan data <i>Team</i> dengan <i>id project</i> = 5</li> <li>- Tampilkan data <i>Task</i> dengan <i>id project</i> = 5</li> </ul>	Handal
Tampil Data User	Pengujian terhadap fungsi Tampil Data User	<ul style="list-style-type: none"> <li>- Memilih salah satu peran dari <i>user</i> yang ditampilkan di <i>detail project</i></li> <li>- Sistem menampilkan seluruh <i>user</i> dengan peran</li> </ul>	- <i>Role</i> : "dev"	<ul style="list-style-type: none"> <li>- Tampilkan semua data <i>user</i> dengan peran "dev" yang memiliki status <i>free</i></li> </ul>	<ul style="list-style-type: none"> <li>- Tampilkan semua data <i>user</i> dengan peran "dev" yang memiliki status <i>free</i></li> </ul>	<ul style="list-style-type: none"> <li>- Tampilkan semua data <i>user</i> dengan peran "dev" yang memiliki status <i>free</i></li> </ul>	Handal

		sesuai dengan pilihan					
Tampil Data Wishlist	Pengujian terhadap fungsi Tampil Data Wishlist	<ul style="list-style-type: none"> <li>- Memilih salah satu peran dari <i>user</i> yang ditampilkan di <i>detail project</i></li> <li>- Sistem menampilkan seluruh <i>user</i> dengan peran sesuai dengan pilihan yang sudah tersimpan di tabel <i>wishlist</i></li> </ul>	- Role : "dev"	<ul style="list-style-type: none"> <li>- Tampilkan semua data <i>user</i> dengan peran "dev" yang sudah berada dalam <i>wishlist</i></li> </ul>	<ul style="list-style-type: none"> <li>- Tampilkan semua data <i>user</i> dengan peran "dev" yang sudah berada dalam <i>wishlist</i></li> </ul>	<ul style="list-style-type: none"> <li>- Tampilkan semua data <i>user</i> dengan peran "dev" yang sudah berada dalam <i>wishlist</i></li> </ul>	Handal
Select User	Pengujian terhadap fungsi Select User	<ul style="list-style-type: none"> <li>- Menekan tombol <i>add</i> pada 1 data <i>user</i></li> <li>- Sistem menyimpan data <i>user</i> kedalam tabel <i>wishlist</i></li> <li>- Sistem mengubah status dari <i>user</i> menjadi "booked"</li> </ul>	- Id user : "12"	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>list user</i> yang dapat dipilih</li> <li>- Data <i>user</i> dengan <i>id</i> = 12 masuk ke dalam <i>list wishlist</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>list user</i> yang dapat dipilih</li> <li>- Data <i>user</i> dengan <i>id</i> = 12 masuk ke dalam <i>list wishlist</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>list user</i> yang dapat dipilih</li> <li>- Data <i>user</i> dengan <i>id</i> = 12 masuk ke dalam <i>list wishlist</i></li> </ul>	Handal

		<ul style="list-style-type: none"> <li>- Menghapus data <i>user</i> dari <i>list</i> untuk dipilih</li> </ul>					
<i>Cancel User</i>	Pengujian terhadap fungsi <i>Cancel User</i>	<ul style="list-style-type: none"> <li>- Menekan tombol <i>cancel</i> pada 1 data <i>user</i></li> <li>- Sistem menghapus data <i>user</i> dari tabel <i>wishlist</i></li> <li>- Sistem mengubah status dari <i>user</i> menjadi "<i>free</i>"</li> <li>- Menampilkan kembali data <i>user</i> dari <i>list</i> untuk dipilih</li> </ul>	<ul style="list-style-type: none"> <li>- <i>Id user</i> : "12"</li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>wishlist</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>wishlist</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>wishlist</i></li> </ul>	Handal
<i>Done Wishlist</i>	Pengujian terhadap fungsi <i>Done Wishlist</i>	<ul style="list-style-type: none"> <li>- Menekan tombol <i>done</i> dari sebuah card yang menampilkan data <i>wishlist</i> untuk <i>project</i> tertentu</li> <li>- Sistem menyimpan</li> </ul>	<ul style="list-style-type: none"> <li>- <i>Id project</i> : "1"</li> <li>- <i>Id user</i> : "12"</li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>wishlist</i></li> <li>- Data <i>user</i> dengan <i>id</i> = 12 tersimpan di dalam <i>team</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>wishlist</i></li> <li>- Data <i>user</i> dengan <i>id</i> = 12 tersimpan di dalam <i>team</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>wishlist</i></li> <li>- Data <i>user</i> dengan <i>id</i> = 12 tersimpan di dalam <i>team</i></li> </ul>	Handal

		<p>semua data dari <i>wishlist</i> dengan <i>id project</i> yang sesuai ke dalam <i>team</i> dengan <i>id project</i> yang sesuai juga</p> <ul style="list-style-type: none"> <li>- Sistem menghapus data <i>user</i> dari <i>wishlist</i> yang baru saja di masukkan ke dalam <i>team</i></li> <li>- Sistem mengubah status <i>user</i> menjadi <i>onteam</i></li> </ul>		<p>untuk <i>id project</i> = 1</p> <ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 berubah status menjadi <i>onteam</i></li> </ul>	<p>untuk <i>id project</i> = 1</p> <ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 berubah status menjadi <i>onteam</i></li> </ul>	<p>untuk <i>id project</i> = 1</p> <ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 berubah status menjadi <i>onteam</i></li> </ul>	
<i>Cancel Wishlist</i>	<p>Pengujian terhadap fungsi <i>Cancel Wishlist</i></p>	<ul style="list-style-type: none"> <li>- Menekan tombol <i>reset</i> dari sebuah card yang menampilkan data <i>wishlist</i> untuk <i>project</i> tertentu</li> <li>- Sistem menghapus semua data dari <i>wishlist</i> dengan <i>id</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Id project</i> : "1"</li> <li>- <i>Id user</i> : "12"</li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>wishlist</i></li> <li>- Data <i>user</i> dengan <i>id</i> = 12 berubah status menjadi <i>free</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>wishlist</i></li> <li>- Data <i>user</i> dengan <i>id</i> = 12 berubah status menjadi <i>free</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>user</i> dengan <i>id</i> = 12 sudah tidak ada di dalam <i>wishlist</i></li> <li>- Data <i>user</i> dengan <i>id</i> = 12 berubah status menjadi <i>free</i></li> </ul>	Handal

		<p><i>project</i> yang sesuai</p> <ul style="list-style-type: none"> <li>- Sistem mengubah status <i>user</i> menjadi <i>free</i></li> </ul>					
Tampil Task	Pengujian terhadap fungsi Tampil Task	<ul style="list-style-type: none"> <li>- Memilih salah satu nama <i>phase</i> dari <i>task</i> sebuah <i>project</i></li> <li>- Data <i>task</i> akan ditampilkan dalam <i>list</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Phase</i> : “<i>assessment</i>”</li> </ul>	<ul style="list-style-type: none"> <li>- <i>Card</i> akan melakukan <i>expand</i> dan menampilkan seluruh <i>task</i> dengan <i>phase assesment</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Card</i> akan melakukan <i>expand</i> dan menampilkan seluruh <i>task</i> dengan <i>phase assesment</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Card</i> akan melakukan <i>expand</i> dan menampilkan seluruh <i>task</i> dengan <i>phase assesment</i></li> </ul>	Handal
Tambah Task	Pengujian terhadap fungsi Tambah Task	<ul style="list-style-type: none"> <li>- Menekan tombol <i>Add</i> pada bagian <i>task</i></li> <li>- Mengisi data <i>task</i> seperti <i>phase</i>, <i>task name</i>, <i>note</i>, <i>person in charge</i>, <i>date start</i> dan <i>date end</i></li> <li>- Tekan tombol <i>create</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Phase</i> : “<i>development</i>”</li> <li>- <i>Task Name</i> : “Membuat halaman <i>Home</i>”</li> <li>- <i>Note</i> : “Halaman pertama saat mengakses web”</li> <li>- <i>Person in Charge</i> : “Budi”</li> <li>- <i>Date Start</i> : “18/06/2019”</li> <li>- <i>Date End</i> : “20/06/2019”</li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>task</i> berhasil disimpan</li> <li>- Data <i>task</i> dimunculkan di dalam <i>list</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>task</i> berhasil disimpan</li> <li>- Data <i>task</i> dimunculkan di dalam <i>list</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>task</i> berhasil disimpan</li> <li>- Data <i>task</i> dimunculkan di dalam <i>list</i></li> </ul>	Handal

<i>Start Task</i>	Pengujian terhadap fungsi <i>Start Task</i>	<ul style="list-style-type: none"> <li>- Menekan tombol <i>Start</i> pada bagian <i>task</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Id task</i> = "2"</li> </ul>	<ul style="list-style-type: none"> <li>- Menambahkan data untuk <i>actual date start</i> pada <i>task</i> yang dimaksud</li> </ul>	<ul style="list-style-type: none"> <li>- Menambahkan data untuk <i>actual date start</i> pada <i>task</i> yang dimaksud</li> </ul>	<ul style="list-style-type: none"> <li>- Menambahkan data untuk <i>actual date start</i> pada <i>task</i> yang dimaksud</li> </ul>	Handal
<i>Done Task</i>	Pengujian terhadap fungsi <i>Done Task</i>	<ul style="list-style-type: none"> <li>- Menekan tombol <i>Done</i> pada bagian <i>task</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Id task</i> = "2"</li> </ul>	<ul style="list-style-type: none"> <li>- Menambahkan data untuk <i>actual date end</i> pada <i>task</i> yang dimaksud</li> </ul>	<ul style="list-style-type: none"> <li>- Menambahkan data untuk <i>actual date end</i> pada <i>task</i> yang dimaksud</li> </ul>	<ul style="list-style-type: none"> <li>- Menambahkan data untuk <i>actual date end</i> pada <i>task</i> yang dimaksud</li> </ul>	Handal
<i>Ubah Task</i>	Pengujian terhadap fungsi <i>Ubah Task</i>	<ul style="list-style-type: none"> <li>- Menekan nama <i>task</i> yang memiliki fungsi <i>ubah task</i></li> <li>- Mengisi data <i>task</i> seperti <i>phase, task name, note, person in charge, date start</i> dan <i>date end</i></li> <li>- Tekan tombol <i>create</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Id task</i> = "5"</li> <li>- <i>Phase</i> : "development"</li> <li>- <i>Task Name</i> : "Membuat halaman home"</li> <li>- <i>Note</i> : (NULL)</li> <li>- <i>Person in Charge</i> : "Budi"</li> <li>- <i>Date Start</i> : "18/06/2019"</li> <li>- <i>Date End</i> : "20/06/2019"</li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>task</i> berhasil disimpan</li> <li>- Data <i>task</i> dimunculkan di dalam <i>list</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>task</i> berhasil disimpan</li> <li>- Data <i>task</i> dimunculkan di dalam <i>list</i></li> </ul>	<ul style="list-style-type: none"> <li>- Data <i>task</i> berhasil disimpan</li> <li>- Data <i>task</i> dimunculkan di dalam <i>list</i></li> </ul>	Handal
<i>Hapus Task</i>	Pengujian terhadap fungsi	<ul style="list-style-type: none"> <li>- Menekan tombol <i>Delete</i> pada bagian <i>task</i></li> </ul>	<ul style="list-style-type: none"> <li>- <i>Id task</i> = "4"</li> </ul>	<ul style="list-style-type: none"> <li>- Menambahkan data untuk <i>actual date start</i> pada</li> </ul>	<ul style="list-style-type: none"> <li>- Menambahkan data untuk <i>actual date start</i> pada</li> </ul>	<ul style="list-style-type: none"> <li>- Menambahkan data untuk <i>actual date start</i> pada</li> </ul>	Handal

	Hapus <i>Task</i>			<i>task</i> yang dimaksud - <i>Task</i> tidak ditampilkan dalam <i>list</i>	<i>task</i> yang dimaksud - <i>Task</i> tidak ditampilkan dalam <i>list</i>	<i>task</i> yang dimaksud - <i>Task</i> tidak ditampilkan dalam <i>list</i>	
--	----------------------	--	--	--	--	--	--





#### 5.4. Analisis Kelebihan dan Kekurangan Sistem

Kelebihan yang dimiliki oleh sistem informasi *resource management* ACCITeam yang telah dirancang ini antara lain:

1. Sistem *resource management* ACCITeam ini akan membantu para *stakeholder* untuk melihat dan mengelola para karyawannya di departemen IT. Para pengguna tidak perlu saling bertatap muka untuk mengatur jalannya *project* maupun kinerja *team* dan dapat mengurangi waktu yang dibutuhkan untuk permintaan anggota *team*.
2. Sistem *resource management* ACCITeam ini dapat diakses dengan mudah melalui *website*. Pengguna dapat mengaksesnya melalui *browser* baik dalam perangkat *desktop* dengan berbagai macam lebar layar maupun perangkat *mobile* selama perangkat tersebut terhubung dengan internet
3. Sistem *resource management* ACCITeam ini memiliki tampilan yang dinamis, sehingga dari segi *User Interface* (UI) sistem ini dapat mudah dilihat dan digunakan oleh para penggunanya.
4. Sistem ini dirancang agar tidak ada penggunaan *form* yang terlalu banyak dalam pelaksanaan *resource management*. Pengguna akan dimanjakan dengan fungsi - fungsi yang berjalan sesuai proses bisnis hanya melalui tombol - tombol.

Sedangkan untuk kekurangan yang dimiliki oleh sistem *resoruce management* ACCITeam ini antara lain:

1. Sistem belum dapat melakukan *auto refresh* pada data yang ditampilkan.
2. Sistem belum mampu membuat atau *generate* laporan dari data kerja *project*.
3. Sistem belum memiliki jalur komunikasi sendiri, sehingga masih membutuhkan komunikasi melalui perantara lain
4. Belum semua halaman memiliki fungsi tanpa *reload page*.

Pada bab implementasi dan pengujian perangkat lunak ini telah dijelaskan mengenai definisi sistem, implementasi sistem, dan hasil pengujian sistem. Pada bab selanjutnya yaitu bab penutup, akan diberikan kesimpulan dan saran yang didapatkan selama pembuatan tugas akhir ini