



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB III**

### **PELAKSANAAN KERJA MAGANG**

#### **3.1 Kedudukan dan Koordinasi**

Selama lebih kurang tiga bulan periode kerja magang dilakukan di PT Dynaplast pada Divisi *Application Development* dengan tugas untuk membuat aplikasi dalam bentuk *web* sesuai dengan *user requirement* yang diperoleh meliputi perancangan *user interface*, perancangan *database*, menyiapkan *server* serta tugas lain yang diperlukan dalam mengembangkan aplikasi Dynaplast e-Room ini, di bawah bimbingan Bapak Sucipto Darmaputra selaku manajer *Application Development*. *User requirement* yang diperlukan dalam pengembangan dihasilkan dari diskusi yang dilakukan setelahnya.

#### **3.2 Tugas yang Dilakukan**

Dalam periode kerja magang, peserta magang bertugas untuk membuat aplikasi berbasis *web* dimana harus didasarkan pada spesifikasi yang telah didapatkan berdasarkan diskusi yang dilakukan dengan Bapak Sucipto Darmaputra. Tugas yang diberikan kepada peserta magang dalam pengembangan Dynaplast e-Room ini adalah sebagai berikut.

1. Pengumpulan *requirement document*, *scope*, dan *goal* dari proyek  
Pengumpulan dokumen-dokumen ini dilakukan untuk mendapatkan informasi mengenai kebutuhan yang benar, sehingga pembangunan aplikasi dapat menjadi solusi yang tepat untuk memecahkan masalah yang ada.
2. Perancangan *database*. *Database* adalah struktur dari koleksi data yang menjadi pusat informasi dari aplikasi ini. *Database* dirancang sesuai dengan *requirement* yang ada, dalam perancangan *database* sangat diusahakan untuk dirancang secara optimal melalui normalisasi.
3. Pembangunan *user interface* yang digunakan sebagai media interaksi sehingga menjadi lebih mudah dan dapat dimengerti baik oleh pengguna

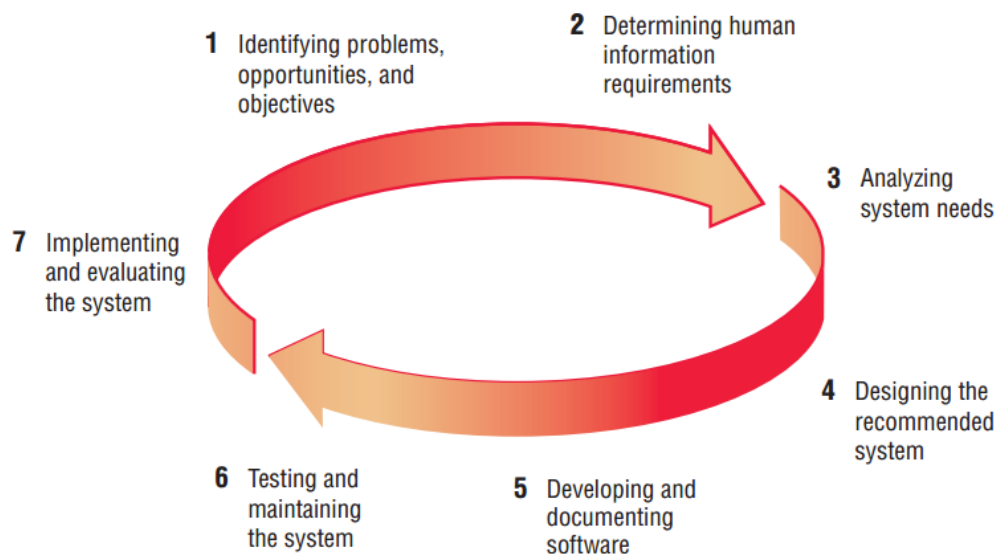
lama, maupun pengguna yang baru. Dalam aplikasi, *user interface* dapat langsung dilihat oleh pengguna ketika mengakses aplikasi yang berbasis *web* ini.

4. Pengembangan *content management system* yang dapat digunakan sebagai perantara dengan sumber daya secara langsung, seperti *database*. *Content management system* ditampilkan sesuai dengan batasan yang ada dengan berdasarkan spesifikasi yang ditentukan. *Content management system* berguna dalam aplikasi dimana pengguna tidak perlu menuliskan perintah-perintah pemrograman, misalkan perintah SQL, sehingga dengan mudah dapat melakukan perubahan terhadap sumber daya yang digunakan.
5. Pengaturan *server* meliputi pengaturan keamanan terhadap *database* dan akses terhadap *file* yang di-*host* pada *server*. Selain itu, penyesuaian terhadap sistem operasi *server* juga perlu dilakukan dan yang terakhir adalah melakukan pengaturan *scheduler* yang digunakan untuk melakukan pemanggilan perintah secara otomatis.

### 3.3 Tahapan Pelaksanaan Kerja Magang

Tahapan pengembangan aplikasi Dynaplast e-Room ini dilaksanakan dalam tujuh tahap berdasarkan pada metodologi pengembangan, yaitu *Systems Development Life Cycle* seperti yang digambarkan pada gambar berikut.

U  
M  
N



Gambar 3.1 *Systems Development Life Cycle* (Kendall & Kendall, 2006)

Berikut ini adalah penjelasan dari masing–masing tahapan dari metodologi yang digunakan sesuai dengan pengembangan Dynaplast e-Room.

### 3.3.1 Tahap Identifikasi Masalah, Peluang, dan Tujuan

Tahap pertama dari *systems development life cycle*, dimana diperlukan pengamatan mengenai bisnis yang berjalan dan setelah itu bersama dengan anggota perusahaan lainnya menentukan dimana letak permasalahan yang terjadi. Selain itu, penentuan peluang solusi dari masalah itu dapat meningkatkan tingkat kompetitif maupun menetapkan standar. Penentuan serta pengetahuan tujuan juga merupakan komponen penting sehingga dapat diketahui apa yang diperlukan oleh perusahaan (Kendall & Kendall, 2006). Dalam tahap ini, bersama dengan Bapak Sucipto, dapat didefinisikan informasi–informasi yang berguna dalam pengembangan aplikasi berikut ini.

1. PT Dynaplast merupakan perusahaan yang bergerak dalam bidang kemasan, maupun komponen lainnya yang terbuat dari plastik serta memiliki cabang, maupun anak perusahaan yang terletak baik di luar, maupun dalam negeri.

2. Dalam melakukan proyek, terutama yang melibatkan anak, maupun cabang perusahaan sering terjadi kendala dimana koordinasi data proyek sulit dilakukan, yang mana biasanya dilakukan melalui surat elektronik, sehingga tidak terorganisir secara teratur.
3. Dalam diskusi bersama dapat diperoleh peluang pemecahan masalah dengan membangun sebuah *document management* dengan menggunakan *server* yang tersedia milik PT Dynaplast.
4. Penetapan *scope* berdasarkan tujuan yang diperoleh.

### 3.3.2 Tahap Menentukan Informasi Kebutuhan Manusia

Tahap selanjutnya adalah tahap penentuan dari *user requirement* yang terlibat dengan menggunakan berbagai alat untuk mengetahui bagaimana pengguna berinteraksi dalam konteks kerja dengan sistem informasi (Kendall & Kendall, 2006). Dalam tahap ini, penentuan kebutuhan dilakukan dengan cara *interview*, mendengarkan pengguna bercerita mengenai apa dan bagaimana pengguna ingin menggunakan aplikasi ini. Dari tahap ini berhasil diperoleh informasi berikut.

1. Fitur-fitur yang diharapkan, maupun dibutuhkan oleh pengguna mengenai aplikasi berbasis *web* yang akan dibangun.
2. Pembagian tingkatan dokumen dalam *room*, *folder*, dan *file*.
3. Pola interaksi yang diharapkan pengguna sehingga dapat memudahkan penggunaan.
4. Penyesuaian fitur, maupun pola interaksi dengan tujuan serta *scope* yang telah diperoleh dari tahap sebelumnya.

### 3.3.3 Tahap Analisis Kebutuhan Sistem

Tahap selanjutnya adalah penetapan kebutuhan sistem dengan menggunakan alat maupun teknik. Alat maupun teknik yang digunakan dapat berupa *data flow diagram* untuk menggambarkan *input*, proses, dan *output* dari fungsi bisnis yang ada atau *activity diagram* atau *sequence diagram* untuk melihat urutan kejadian

(Kendall & Kendall, 2006). Dari tahap ini dihasilkan *data flow diagram* sesuai dengan informasi yang telah didapatkan sebelumnya.

### **3.3.4 Tahap Merancang Sistem Rekomendasi**

Tahap dimana seluruh informasi yang telah didapatkan digunakan untuk menghasilkan desain logis dari sebuah sistem informasi. Bagian dari desain logis akan digunakan untuk merancang *human computer interaction* (Kendall & Kendall, 2006). Dari tahap ini, dihasilkan *entity relationship diagram* yang menjelaskan mengenai *database*, serta informasi mengenai *field* yang ada dan hubungan antara masing-masing *database* yang diperlukan.

### **3.3.5 Tahap Pengembangan dan Dokumentasi Perangkat Lunak**

Merupakan tahap kelima dari keseluruhan *systems development life cycle*. Dalam tahap ini, pengembangan aplikasi dengan menggunakan informasi yang didapatkan sebelumnya mulai dilakukan. Selain itu, penulisan dokumentasi dalam bentuk *help* atau *frequently asked question* dapat dimulai (Kendall & Kendall, 2006). Dalam tahap ini, perlu diperhatikan hal-hal berikut.

1. Pengembangan adalah berbasis *web* dengan bahasa pemrograman yang sesuai dan juga IDE yang dapat mendukungnya.
2. *Framework* yang diperlukan disertai dengan *plugin* lainnya yang digunakan.
3. Perangkat lunak lainnya yang membantu pengembangan (penjelajah *web*, aplikasi *server virtual*, dan lainnya).
4. Penulisan *help* yang dapat diakses pada aplikasi serta *frequently asked question* yang dapat ditambahkan secara dinamis juga dapat diakses melalui aplikasi.

### **3.3.6 Tahap Pengetesan dan Pemeliharaan Sistem**

Sebelum digunakan, sistem informasi perlu dilakukan pengetesan sehingga seluruh masalah dapat diketahui sebelum digunakan, dimana hal ini lebih baik jika dibandingkan apabila masalah terjadi saat digunakan. Selain pengetesan, hal lain

yang perlu dilakukan adalah pemeliharaan, baik sistem maupun dokumentasi, yang dilakukan secara rutin (Kendall & Kendall, 2006). Dalam tahap ini, perlu diperhatikan hal-hal sebagai berikut.

1. Pengetesan, baik tampilan maupun fungsionalitas, tahap pertama dilakukan oleh peserta magang sebagai pihak yang membangun aplikasi dengan menggunakan data contoh.
2. Pengetesan, baik tampilan maupun fungsionalitas, tahap kedua dilakukan oleh peserta magang dan pembimbing lapangan dengan menggunakan data contoh.
3. Pengetesan, baik tampilan maupun fungsionalitas, dilakukan oleh pembimbing magang dan pengguna dengan menggunakan data aktual.
4. Pemeliharaan dokumentasi terus dilakukan dalam bentuk *frequently asked question* yang dinamis.

### 3.3.7 Tahap Implementasi dan Evaluasi Sistem

Tahap terakhir dalam *systems development life cycle*, dimana proses mengubah *file* dari format yang lama ke format yang baru, atau membangun *database*, memasang peralatan, dan menerapkan sistem yang baru ke dalam produksi.

Evaluasi juga termasuk dalam tahap terakhir ini, biasanya digunakan sebagai bahan diskusi. Sebenarnya evaluasi dilakukan pada setiap tahap, ketika suatu tahap selesai dan bergerak ke tahap selanjutnya, masalah dapat ditemukan, sehingga harus melakukan pengulangan tahap yang telah selesai.

Dalam tahap terakhir ini, perlu diperhatikan mengenai hal-hal berikut.

1. Implementasi dilakukan dengan melakukan pengaturan aplikasi pada komputer *server* yang sudah disiapkan oleh perusahaan.
2. Evaluasi baik dalam hal tampilan, pola interaksi, fungsionalitas, kesesuaian dengan kebutuhan yang sudah dideskripsikan sebelumnya.



### 3.4 Uraian Pelaksanaan Kerja Magang

Berikut adalah uraian mengenai pelaksanaan kerja magang dalam pengembangan aplikasi berbasis *web*, Dynaplast e-Room.

#### 3.4.1 *User Requirement*

Berikut adalah *user requirement* yang didapatkan melalui proses diskusi bersama dengan pembimbing magang.

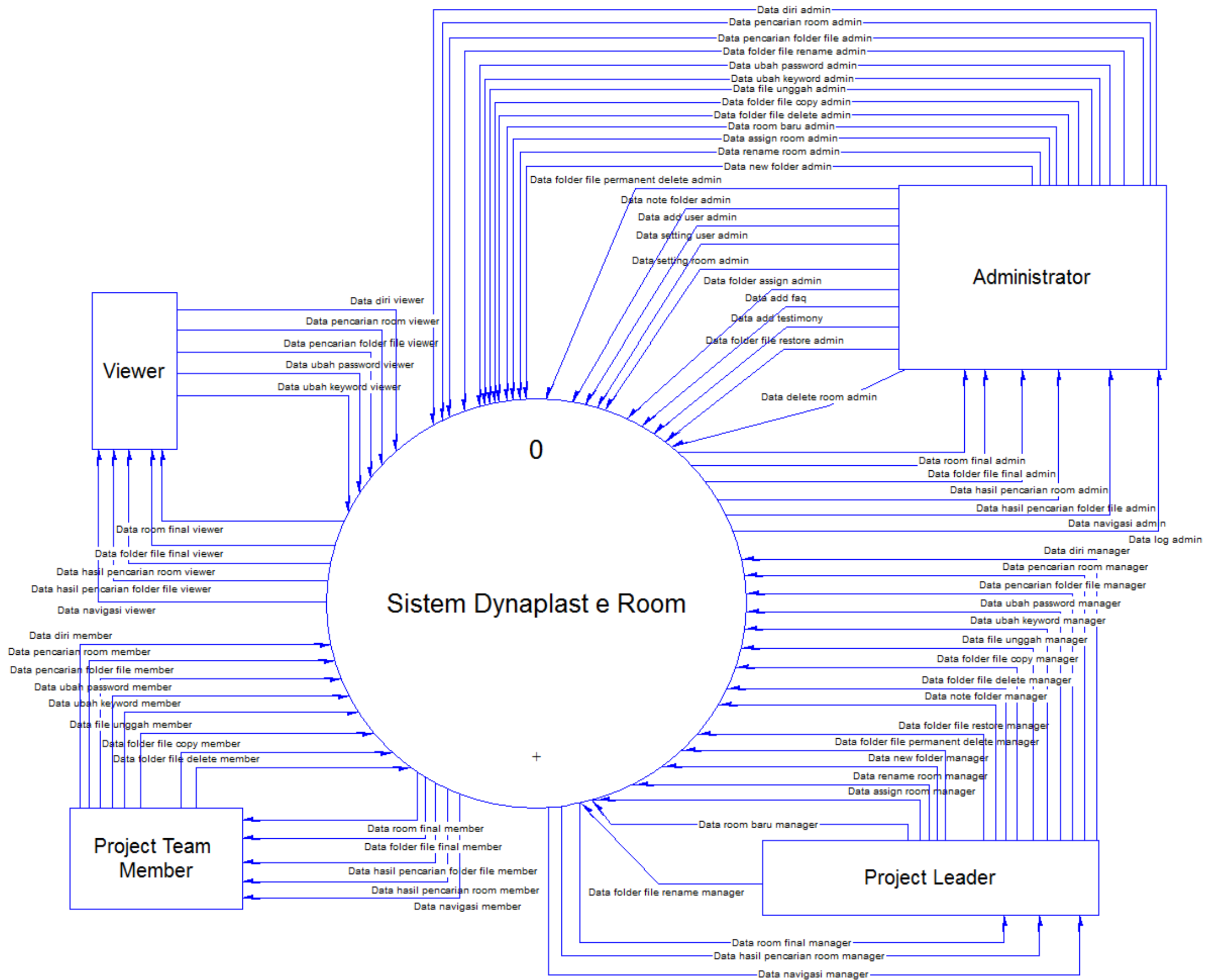
1. Menggunakan *authentication* pengguna berdasarkan surat elektronik yang dimiliki oleh pengguna dimana penambahan pengguna dilakukan oleh administrator.
2. Pembagian tanggung jawab yang bisa dilakukan berdasarkan posisi yang telah ditentukan.
3. Memiliki fasilitas dimana administrator dapat menonaktifkan sebuah akun dan mengaktifkan kembali akun yang nonaktif.
4. Pembagian dua status pada *room* yakni, *active* dimana isi *room* dapat diubah dan *deactive* dimana isi *room* sudah tidak dapat diubah.
5. Memiliki fasilitas pembatasan akses pengguna, baik pada *room* maupun *folder*.
6. Memiliki pengaturan untuk mengubah informasi maupun batasan pada *user*, *room*, dan *folder*.
7. Menggunakan notifikasi melalui surat elektronik secara otomatis dan berkala untuk memberikan informasi log kegiatan pada *room* dimana pengguna memiliki akses serta kegiatan lain mengenai akun pengguna. Untuk administrator, informasi mengenai keseluruhan log kegiatan akan dikirimkan, selain itu informasi terhadap keadaan ukuran dari sebuah *room* jika sudah hampir penuh juga diberikan.
8. Memiliki fasilitas untuk melakukan pengunggahan *file* dari sisi pengguna ke *server*.
9. Memiliki fungsi standar seperti pada layaknya sebuah *file explorer*, meliputi fungsi *create new room* dan *folder*, *cut* serta *paste* pada *folder* dan *file*, *rename folder* dan *file*, *restore* serta *permanent delete* pada

*folder* dan *file* yang ada di *recycle bin*, dan *select all* pada *folder* maupun *file*.

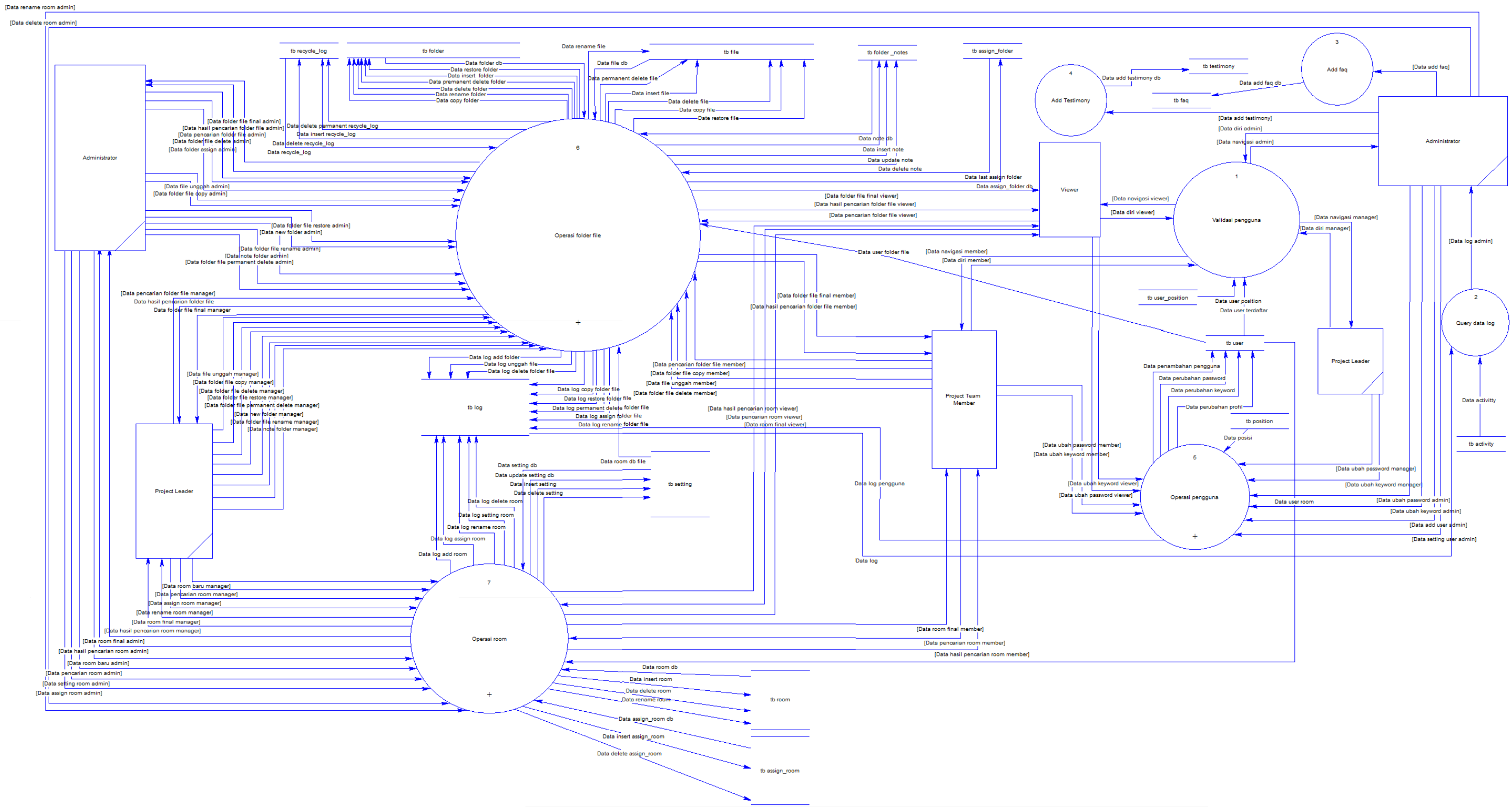
10. Fungsi standar yang ada disesuaikan dengan tingkatan direktori.
11. Pada tingkatan pertama direktori tidak diperbolehkan pengunggahan *file*.
12. Memiliki penampungan sementara untuk *folder* atau *file* yang dihapus sebelum kemudian dihapus secara permanen oleh administrator.
13. Memiliki *help* yang terdapat pada aplikasi dan bersifat statis, FAQ yang bisa ditambahkan oleh administrator, dan *testimonial* yang juga bisa ditambahkan oleh administrator.
14. Memiliki fasilitas dimana pengguna dapat memberikan pertanyaan dan testimoni kepada administrator.
15. Memiliki fasilitas *note* untuk memberikan informasi tambahan pada *folder* atau *file*.

### **3.4.2 Data Flow Diagram**

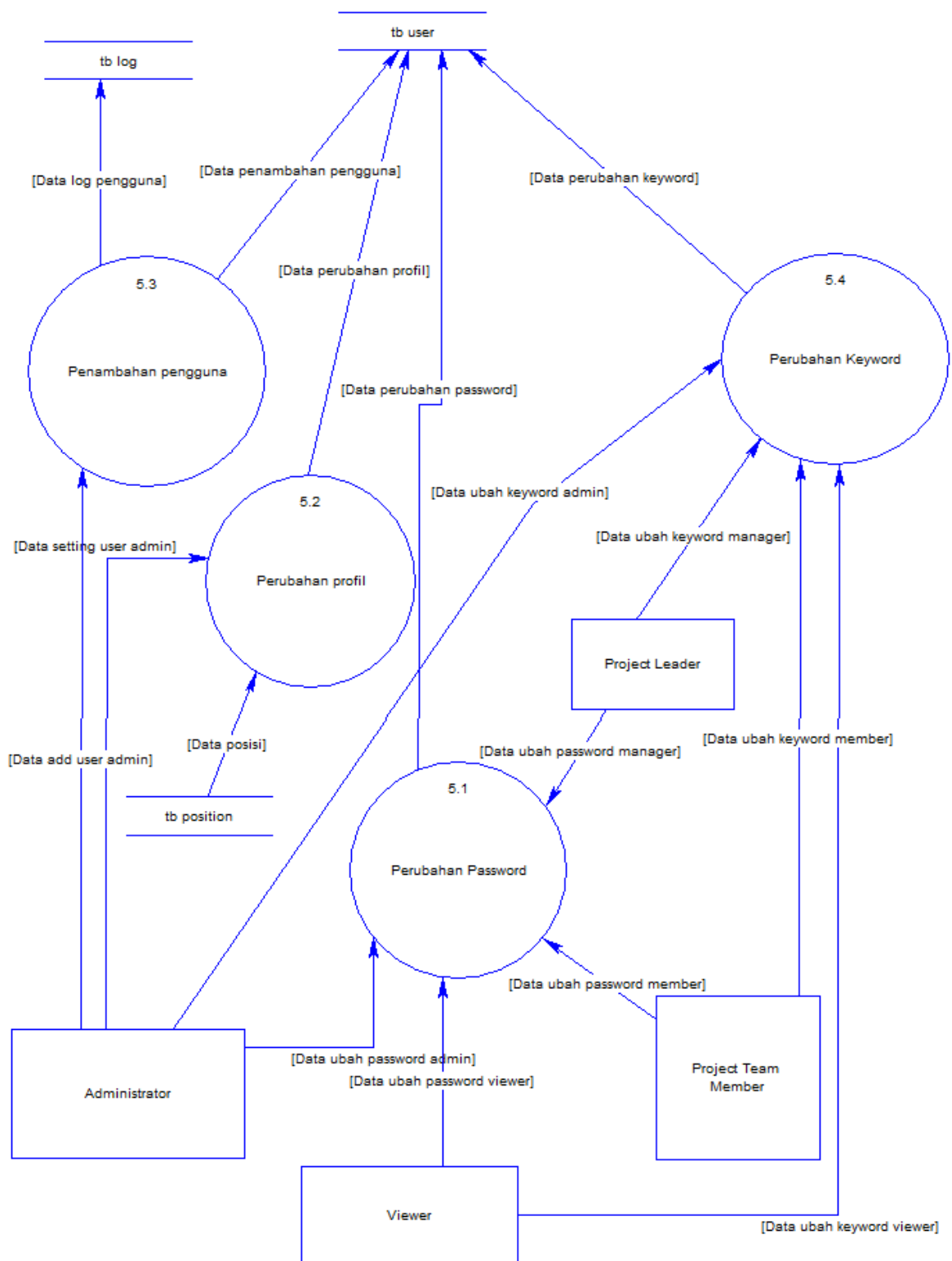
U  
M  
M  
N



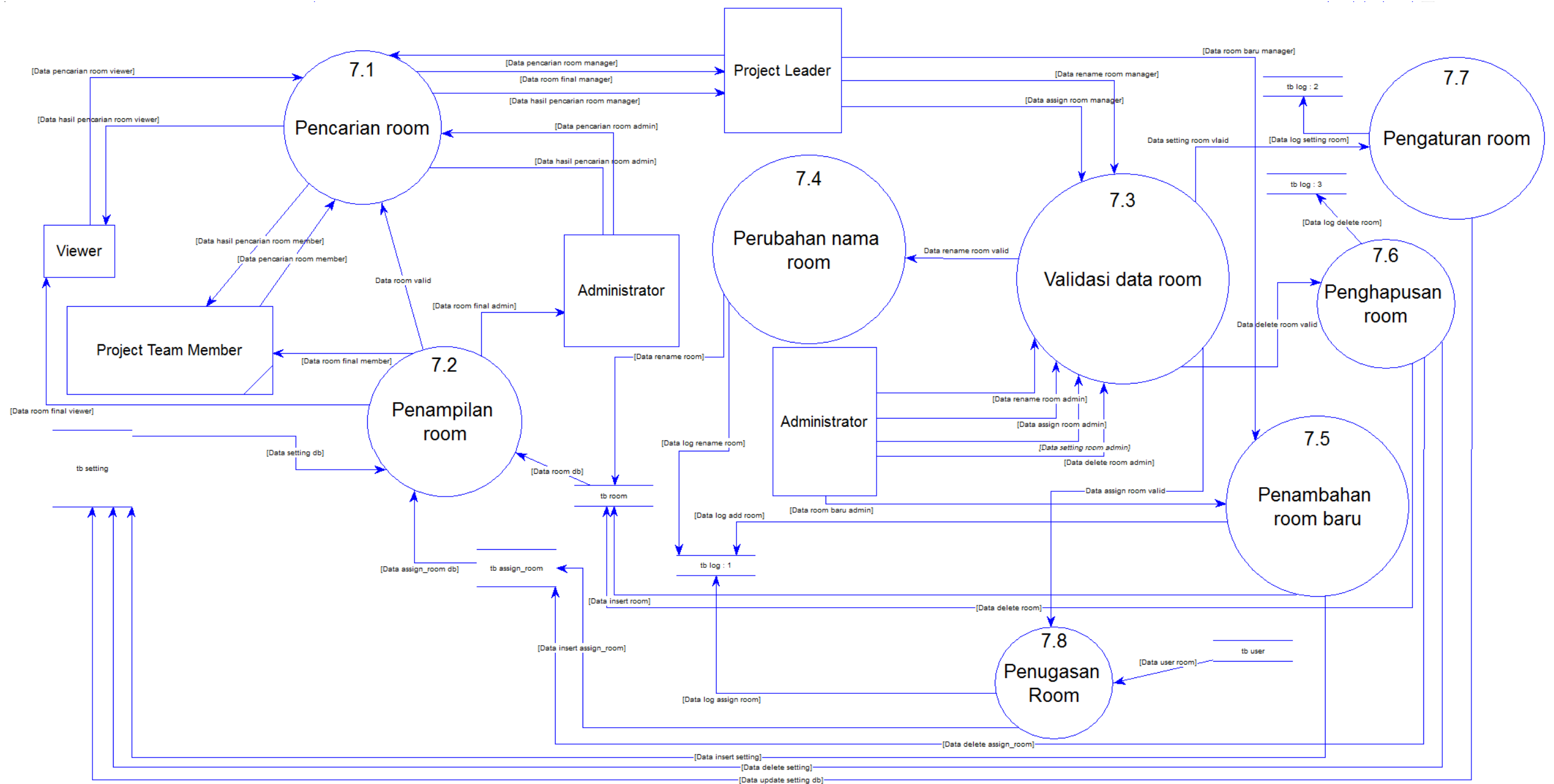
Gambar 3.2 Diagram Konteks



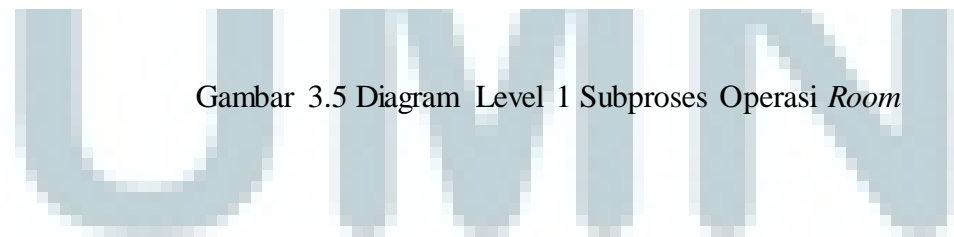
Gambar 3.3 Diagram Level 0

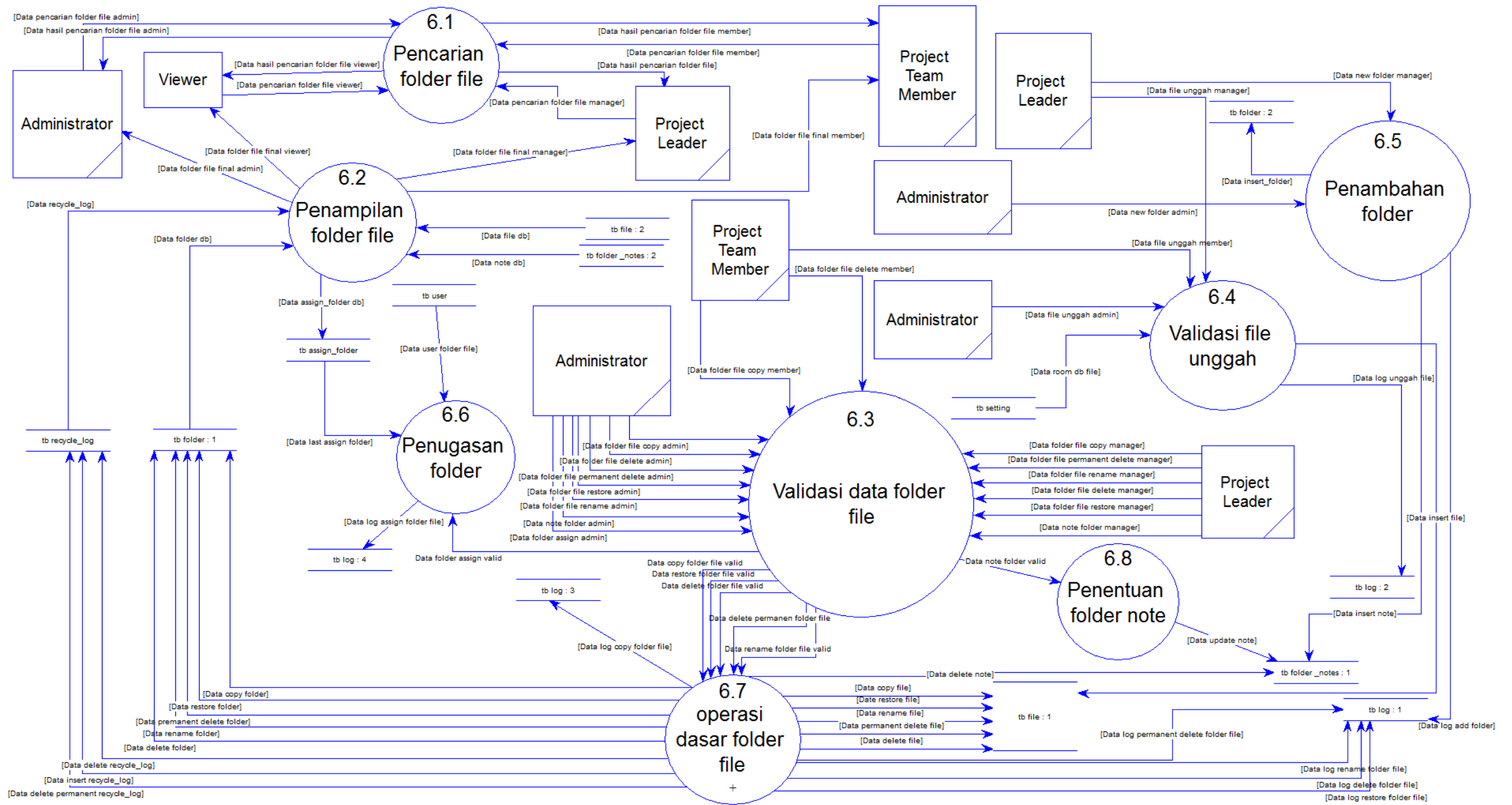


Gambar 3.4 Diagram Level 1 Subproses Operasi Pengguna

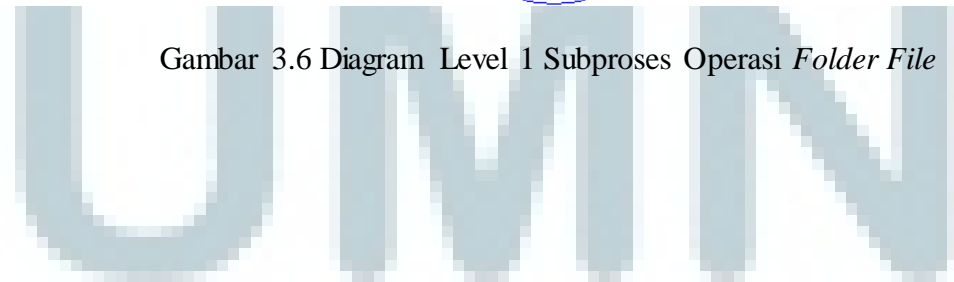


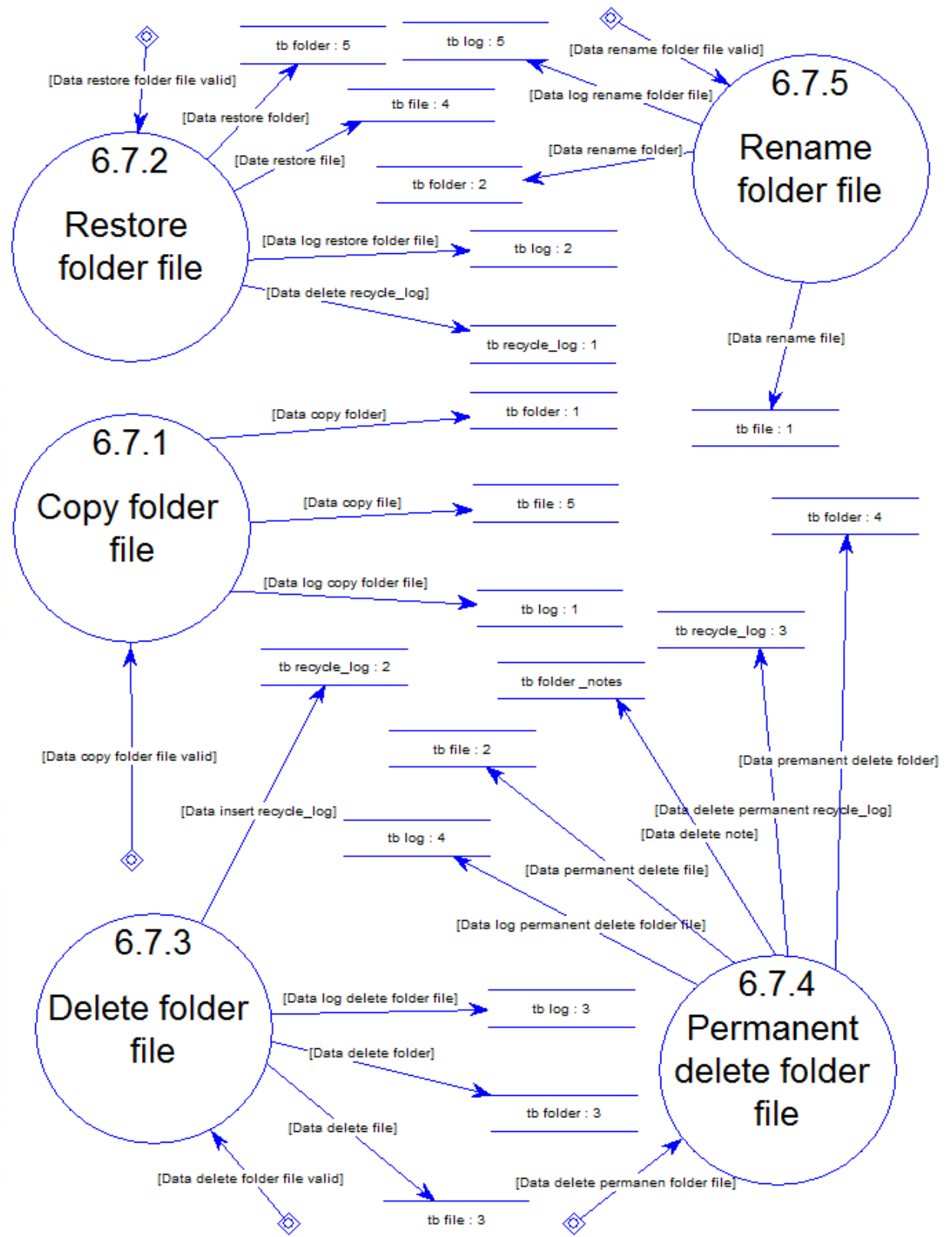
Gambar 3.5 Diagram Level 1 Subproses Operasi Room





Gambar 3.6 Diagram Level 1 Subproses Operasi *Folder File*





Gambar 3.7 Diagram Level 2 Subproses Operasi Dasar *Folder File*



Pada gambar 3.1 dapat ditunjukkan bahwa pada sistem Dyanaplast e-room ini, terdapat empat macam tingkatan pengguna, yaitu *viewer*, *project member*, *project leader*, dan administrator. *Viewer* hanya dapat melakukan *login*, perubahan yang hanya menyangkut pengguna itu sendiri, dan melihat serta melakukan pencarian pada *folder* dan *file*. *Project member* memiliki hak akses sama seperti *viewer*, namun diberikan fungsi tambahan untuk melakukan unggah *file* serta *copy* dan *delete folder* atau *file*. *Project leader* diberikan tambahan hak akses meliputi menambah *folder*, memberikan hak akses *folder* pada pengguna lain, memberikan catatan pada *folder*, melakukan *delete* permanen, melakukan *restore* pada *folder* atau *file*, membuat *room* baru serta mengatur hak aksesnya, dan melakukan *rename* pada *folder* atau *file*. Administrator dapat melakukan seluruh fungsi yang ada, meliputi menambah pengguna, melakukan perubahan informasi pada pengguna, menambah FAQ dan *help*, mengubah pengaturan *room*, melihat log, dan sebagainya.

Pada gambar 3.2 menunjukkan proses–proses besar yang ada, yaitu *add faq*, *add testimony*, *query data log* dimana administrator dapat melihat log yang ada, operasi pengguna, operasi *room*, operasi *folder file*, dan validasi pengguna untuk mengecek hak akses pengguna.

Pada gambar 3.3 menunjukkan proses–proses yang merupakan penjelasan lebih lanjut mengenai proses operasi pengguna pada gambar 3.3, pada diagram ini menunjukkan fasilitas–fasilitas yang menyangkut mengenai pengguna, hak akses pada fasilitas tersebut, dan data–data yang diperlukan pada masing–masing proses.

Pada gambar 3.4 menunjukkan proses–proses yang merupakan penjelasan lebih lanjut mengenai proses operasi *room* pada gambar 3.3, pada diagram ini terlihat fasilitas–fasilitas yang menyangkut mengenai pengaturan *room*, hak akses pada fasilitas tersebut, dan data–data yang diperlukan pada masing–masing proses. Pada diagram ini terdapat validasi yang berguna untuk menentukan mengenai data *room* yang diberikan benar dan ada pada lokasi yang benar atau tidak.

Pada gambar 3.5 menggambarkan proses – proses yang menjadi penjelasan lebih lanjut dari proses operasi *folder file* yang ada pada gambar 3.3, saat melakukan unggah, menambah *folder*, dan melakukan operasi dengan objek

*folder* maupun *file* akan dilakukan validasi, seperti jika ingin unggah maka akan dilihat nama yang dimasukkan sudah ada, atau pada operasi dengan objek *folder* atau *file*, *folder* atau *file* yang menjadi objek benar ada pada sistem. Validasi diperlukan karena aplikasi ini berbasis *web* dimana dapat diakses oleh banyak pengguna sekaligus.

### 3.4.3 Struktur Tabel

Berikut adalah struktur dari tabel *database* yang digunakan dalam aplikasi Dynaplast e-Room.

#### 1. Tabel *user*

Tabel *user* berfungsi untuk menyimpan data-data pengguna secara umum. Berikut adalah struktur tabel *user*.

Tabel 3.1 Tabel *user*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
id	int(11)	Not Null PRIMARY KEY	Nomor unik dari masing-masing pengguna
email	varchar(255)	Not Null UNIQUE	Surat elektronik pengguna dimana hanya dapat didaftarkan 1 kali
password	varchar(50)	Not Null	Kata kunci dari pengguna untuk masuk ke dalam aplikasi
first_name	varchar(15)	Not Null	Nama depan dari pengguna
last_name	varchar(15)	Not Null	Nama belakang dari pengguna
company	varchar(50)	Not Null	Perusahaan dimana pengguna sekarang terdaftar
keyword	varchar(35)		Kata kunci yang digunakan bila pengguna melupakan <i>password</i>

valid	int(11)	Not Null	Penunjuk apakah pengguna aktif atau nonaktif
-------	---------	----------	--

## 2. Tabel *position*

Tabel *position* digunakan untuk mendeskripsikan tingkatan akses yang ada pada aplikasi ini. Berikut adalah struktur tabel *position*.

Tabel 3.2 Tabel *position*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
id	int(11)	Not Null PRIMARY KEY	Nomor unik dari masing-masing tingkat akses
position	varchar(50)	Not Null	Nama untuk masing-masing tingkat akses

## 3. Tabel *user\_position*

Tabel *user\_position* digunakan untuk menyimpan data mengenai penentuan tingkatan akses untuk masing-masing pengguna. Berikut adalah struktur tabel *user\_position*.

Tabel 3.3 Tabel *user\_position*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
id	int(11)	Not Null PRIMARY KEY	Nomor urut dari masing-masing <i>record</i>
user_id	int(11)	Not Null UNIQUE	Nomor unik untuk mewakili pengguna
position_id	int(11)	Not Null	Nomor untuk mewakili tingkat akses

## 4. Tabel *room*

Tabel *room* digunakan untuk menyimpan informasi mengenai *room* yang sudah dibuat dalam aplikasi. Berikut adalah struktur tabel *room*.

Tabel 3.4 Tabel *room*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
room_id	int(11)	Not Null PRIMARY KEY	Nomor unik untuk masing-masing <i>room</i>
room_name	varchar(50)	Not Null	Nama <i>room</i>
created_date	datetime	Not Null	Tanggal dan waktu <i>room</i> dibuat
last_update	datetime	Not Null	Tanggal dan waktu terakhir kali <i>room</i> dimutakhirkan
contributor	int(11)	Not Null	Nomor pengguna yang membuat <i>room</i>

5. Tabel *setting*

Tabel *setting* digunakan untuk menyimpan informasi pengaturan yang dilakukan pada *room*. Berikut adalah struktur tabel *setting*.

Tabel 3.5 Tabel *setting*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
setting_id	int(11)	Not Null PRIMARY KEY	Nomor urut dari masing-masing <i>record</i>
room_id	int(11)	Not Null UNIQUE	Nomor unik untuk mewakili <i>room</i>
active	int(11)	Not Null	Status dari <i>room</i>
extension	varchar(3000)	Not Null	Jenis <i>file extension</i> yang dapat ditambahkan ke dalam <i>room</i>
limit_size	bigint(20)	Not Null	Besar ukuran maksimum dari <i>room</i>
limit_size_limit	int(11)	Not Null	Status batasan maksimum ukuran <i>room</i>
duplication	int(11)	Not Null	Jenis operasi jika ditemukan file ganda

upload_size	bigint(20)	Not Null	Besar ukuran maksimum dari <i>file</i> yang diunggah
upload_size_limit	int(11)	Not Null	Status batasan maksimum ukuran pengunggahan

6. Tabel *assign\_room*

Tabel *assign\_room* berfungsi untuk menyimpan informasi mengenai hak akses *room* kepada pengguna. Berikut adalah struktur tabel *assign\_room*.

Tabel 3.6 Tabel *assign\_room*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
assign_id	int(11)	Not Null PRIMARY KEY	Nomor urut dari masing-masing <i>record</i>
user_id	int(11)	Not Null	Nomor untuk mewakili pengguna
room_id	int(11)	Not Null	Nomor untuk mewakili <i>room</i>

7. Tabel *folder*

Tabel *folder* berfungsi untuk menyimpan informasi mengenai *folder* yang sudah dibuat dalam aplikasi. Berikut adalah struktur tabel *folder*.

Tabel 3.7 Tabel *folder*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
folder_id	int(11)	Not Null PRIMARY KEY	Nomor unik untuk masing-masing <i>folder</i>
folder_name	varchar(200)	Not Null	Nama <i>folder</i>
created_date	datetime	Not Null	Tanggal dan waktu <i>folder</i> dibuat
last_update	datetime	Not Null	Tanggal dan waktu <i>folder</i> terakhir kali dimutakhirkan
contributor	int(11)	Not Null	Nomor pengguna yang membuat <i>folder</i>

root_id	int(11)	Not Null	Nomor <i>room</i> dimana <i>folder</i> berada
parent_id	int(11)	Not Null	Nomor <i>folder</i> yang berada satu tingkat direktori di atas <i>folder</i>
in_recycle	int(11)	Not Null	Status <i>folder</i> apakah berada di <i>recycle bin</i>
is_recycle	int(11)	Not Null	Jenis <i>folder</i>
limit_access	int(11)	Not Null	Status <i>folder</i> mengenai pemberlakuan hak akses

8. Tabel *assign\_folder*

Tabel *assign\_folder* berfungsi untuk menyimpan hak akses *folder* kepada pengguna. Berikut adalah struktur tabel *assign\_folder*.

Tabel 3.8 Tabel *assign\_folder*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
assign_id	int(11)	Not Null PRIMARY KEY	Nomor urut dari masing-masing <i>record</i>
user_id	int(11)	Not Null	Nomor untuk mewakili pengguna
folder_id	int(11)	Not Null	Nomor untuk mewakili <i>folder</i>

9. Tabel *folder\_notes*

Tabel *folder\_notes* berfungsi untuk menyimpan informasi tambahan berupa teks pada *folder* yang ada. Berikut adalah struktur tabel *folder\_notes*.

Tabel 3.9 Tabel *folder\_notes*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
note_id	int(11)	Not Null PRIMARY KEY	Nomor urut dari masing-masing <i>record</i>
note_content	varchar(700)		Isi dari informasi tambahan

folder_id	int(11)	Not Null UNIQUE	Nomor untuk mewakili <i>folder</i>
-----------	---------	--------------------	------------------------------------

#### 10. Tabel *file*

Tabel *file* berfungsi untuk menyimpan informasi mengenai *file* yang sudah diunggah ke dalam aplikasi. Berikut adalah struktur tabel *file*.

Tabel 3.10 Tabel *file*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
file_id	int(11)	Not Null PRIMARY KEY	Nomor unik untuk masing-masing <i>file</i>
file_name	varchar(200)	Not Null	Nama <i>file</i>
created_date	datetime	Not Null	Tanggal dan waktu <i>file</i> dibuat
last_update	datetime	Not Null	Tanggal dan waktu <i>file</i> terakhir kali dimutakhirkan
contributor	int(11)	Not Null	Nomor pengguna yang mengunggah <i>file</i>
file_size	bigint(20)	Not Null	Ukuran dari <i>file</i>
folder_id	int(11)	Not Null	Nomor <i>folder</i> yang berada satu tingkat direktori di atas <i>file</i>
in_recycle	int(11)	Not Null	Status <i>file</i> apakah berada di <i>recycle bin</i>

#### 11. Tabel *activity*

Tabel *activity* berfungsi untuk menyimpan jenis aktivitas yang bisa dilakukan pengguna, meliputi aktivitas informasi pengguna maupun aktivitas terhadap objek-objek yang ada pada aplikasi. Berikut adalah struktur tabel *activity*.

Tabel 3.11 Tabel *activity*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
activity_id	int(11)	Not Null PRIMARY KEY	Nomor unik untuk masing-masing aktifitas
activity_name	varchar(30)	Not Null	Nama aktifitas

12. Tabel log

Tabel log berfungsi untuk menyimpan informasi aktivitas yang dilakukan pengguna saat menggunakan aplikasi, misalkan seperti membuat *folder* baru dan mengubah *password*. Berikut adalah struktur tabel log.

Tabel 3.12 Tabel log

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
log_id	int(11)	Not Null PRIMARY KEY	Nomor unik untuk masing-masing log
activity_id	int(11)	Not Null	Nomor untuk mewakili aktifitas yang dilakukan
user_id	int(11)	Not Null	Nomor untuk mewakili pengguna yang melakukan aktifitas
object_name	varchar(200)	Not Null	Nama dari objek aktifitas
room_id	int(11)	Not Null	Nomor untuk mewakili <i>room</i> yang terkait dengan aktifitas
date	datetime	Not Null	Tanggal dan waktu aktifitas dilakukan

13. Tabel *recycle\_log*

Tabel *recycle\_log* berfungsi untuk menyimpan informasi jika ada *folder* atau *file* yang sedang berada di *recycle bin*. Berikut adalah struktur tabel *recycle\_log*.



Tabel 3.13 Tabel *recycle\_log*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
log_id	int(11)	Not Null PRIMARY KEY	Nomor unik untuk masing-masing log
recycle_id	int(11)	Not Null	Nomor untuk mewakili <i>folder</i> yang berupa <i>recycle bin</i>
name	varchar(200)	Not Null	Nama sementara dari <i>folder</i> atau <i>file</i>
original_parent_id	int(11)	Not Null	Nomor untuk mewakili <i>folder</i> tempat sebenarnya <i>folder</i> atau <i>file</i> berada
obj_id	int(11)	Not Null	Nomor untuk mewakili <i>folder</i> atau <i>file</i>
user_id	int(11)	Not Null	Nomor untuk mewakili pengguna yang melakukan penghapusan
type	varchar(11)	Not Null	Jenis dari objek yang dihapus
delete_date	datetime	Not Null	Tanggal dan waktu penghapusan dilakukan

14. Tabel faq

Tabel faq berfungsi untuk menyimpan informasi bantuan berupa *frequently asked question*, yaitu pertanyaan dan jawaban. Berikut adalah struktur tabel faq.

Tabel 3.14 Tabel faq

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
faq_id	int(11)	Not Null PRIMARY KEY	Nomor unik untuk masing-masing <i>record</i>

question	varchar(300)	Not Null	Pertanyaan yang dijawab pada <i>record</i> yang sama
answer	varchar(800)	Not Null	Bantuan berupa jawaban atas pertanyaan pada <i>record</i> yang sama

#### 15. Tabel *testimony*

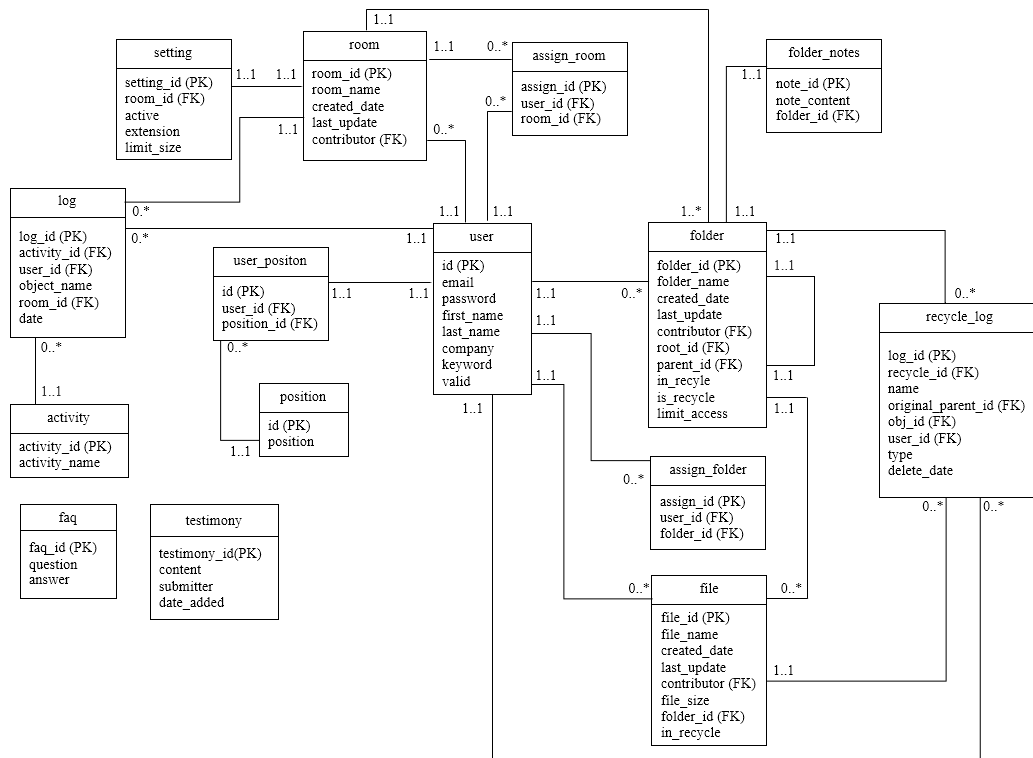
Tabel *testimony* berfungsi untuk menyimpan data-data testimoni yang bisa dimasukkan oleh *administrator*. Berikut adalah struktur tabel *testimony*.

Tabel 3.15 Tabel *testimony*

NAMA FIELD	TIPE	CONSTRAINT	KETERANGAN
testimony_id	int(11)	Not Null PRIMARY KEY	Nomor unik untuk masing-masing <i>record</i>
content	text	Not Null	Konten dari testimoni berupa teks
submitter	varchar(31)	Not Null	Nama pengguna yang menuliskan testimoni
date_added	datetime	Not Null	Tanggal dan waktu testimoni dimasukkan

#### 3.4.4 Entity Relationship Diagram

Deskripsi hubungan antara tabel-tabel yang digunakan dalam aplikasi ini dapat digambarkan dalam diagram berikut.



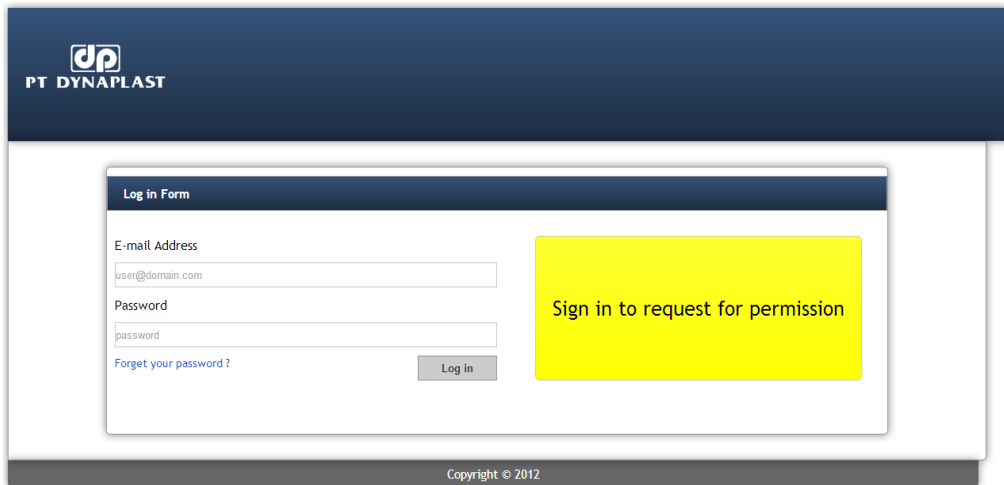
Gambar 3.8 Entity relationship diagram

### 3.4.5 Prototipe

Dalam pengerjaan aplikasi terdapat perubahan terhadap *interface* yang terjadi baik karena penyesuaian dengan gambaran perusahaan ataupun untuk lebih memudahkan pengguna dalam berinteraksi dengan aplikasi. Seluruh *form* yang ada pada aplikasi akan memberikan pesan berhasil atau gagal sesuai dengan informasi atau data yang diberikan pengguna. Berikut adalah *interface* beserta dengan perubahan yang ada pada aplikasi Dynaplast e-Room.

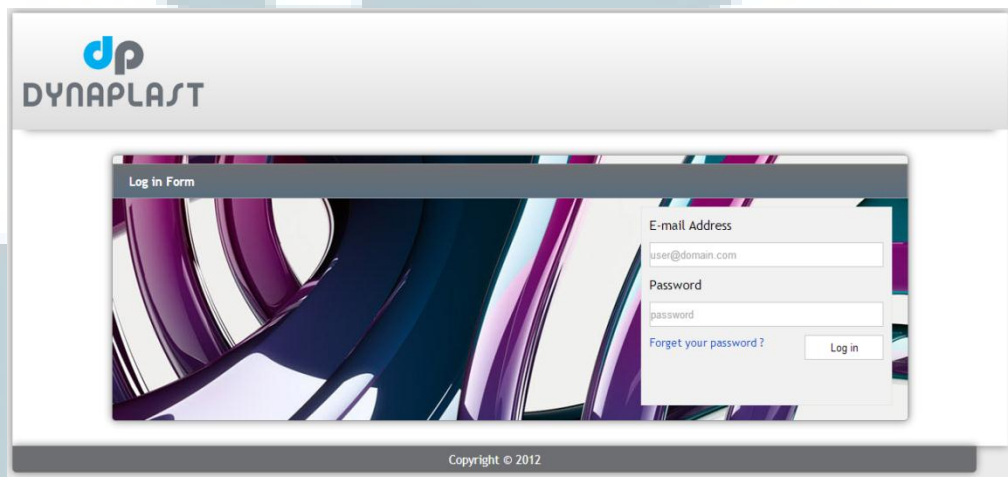
#### 1. Halaman *Login*

Secara umum halaman *login* digunakan sebagai *authentication* pengguna, dimana pengguna harus memasukkan data pengenal berupa alamat surat elektronik yang terdaftar, serta *password* yang sesuai.



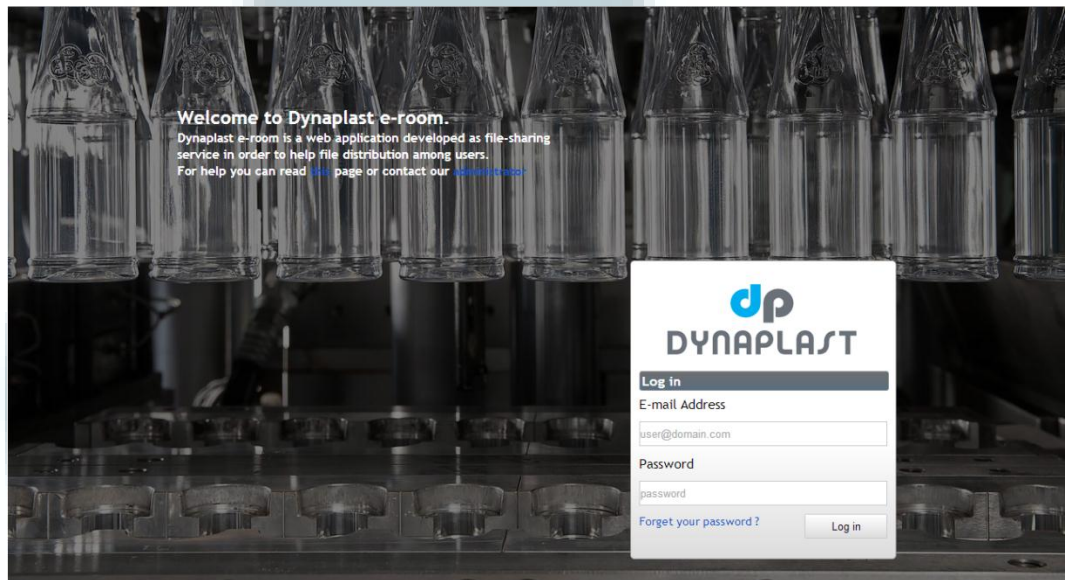
Gambar 3.9 Halaman *Login Prototype Pertama*

Pada *prototype* halaman *login* pertama, logo yang digunakan pada halaman ini menggunakan logo PT Dynaplast yang lama, dimana pada saat itu PT Dynaplast mengubah logonya menjadi yang baru, sehingga perubahan untuk menyesuaikan dengan *brand image* perusahaan yang baru perlu dilakukan. Selain itu, desain yang masih kaku menjadi alasan untuk mengubahnya.



Gambar 3.10 Halaman *Login Prototype Kedua*

Pada *prototype* kedua, logo yang baru telah digunakan dan pengecilan terhadap *form login* telah dilakukan, namun karena dianggap kurang mewakili PT Dynaplast, maka perlu diubah kembali.

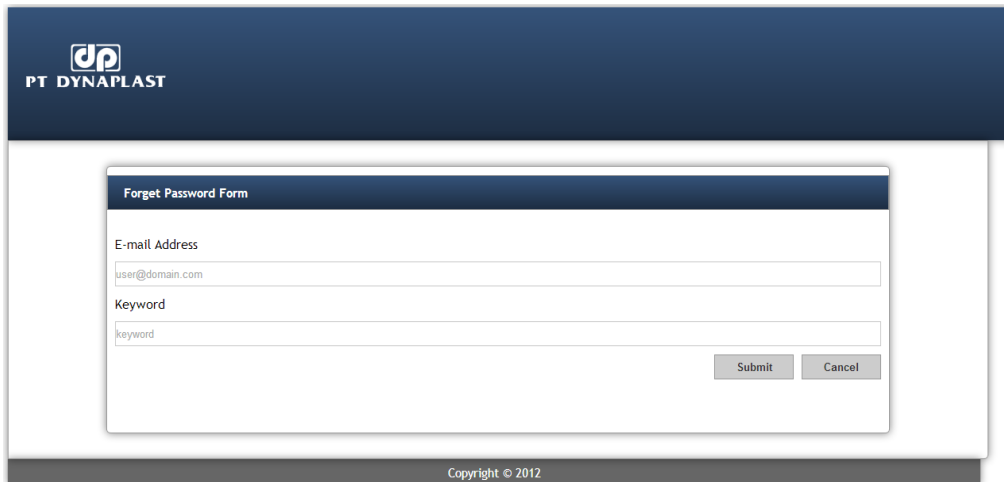


Gambar 3.11 Halaman *Login Akhir*

Pada desain akhir halaman login dengan gambar latar mesin *fold* plastik milik Dynaplast ini dilengkapi dengan tautan untuk melihat halaman *help* maupun tautan untuk mengirim pesan melalui surat elektronik kepada administrator. Jika proses *login* berhasil, maka pengguna akan dilanjutkan ke halaman *room manager*, dan jika gagal, maka pesan kesalahan akan ditampilkan untuk memberikan informasi mengenai kesalahan masukkan pengguna.

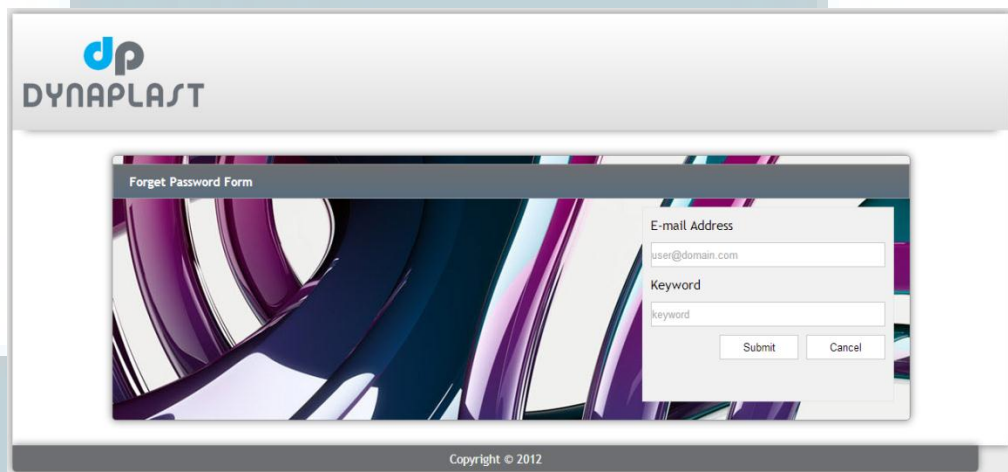
## 2. Halaman *Forget Password*

Halaman ini berfungsi untuk pengguna, bila pengguna tidak mengingat *password* yang digunakan untuk *authentication*. Dalam halaman *forget password*, pengguna perlu memasukkan alamat surat elektronik yang sudah terdaftar, dan juga *keyword* yang sudah ditentukan sebelumnya.



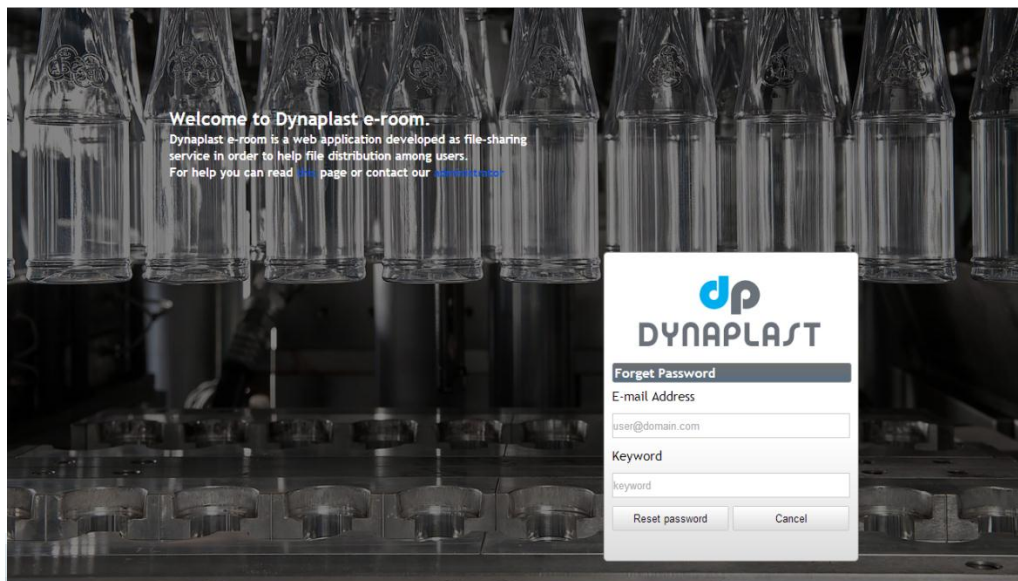
Gambar 3.12 Halaman *Forget Password Prototype* Pertama

Pada *prototype* pertama, *text field* terlalu luas sehingga sangat menyulitkan pengguna dalam menggunakannya. Selain itu, perubahan logo kembali menjadi alasan untuk diubah.



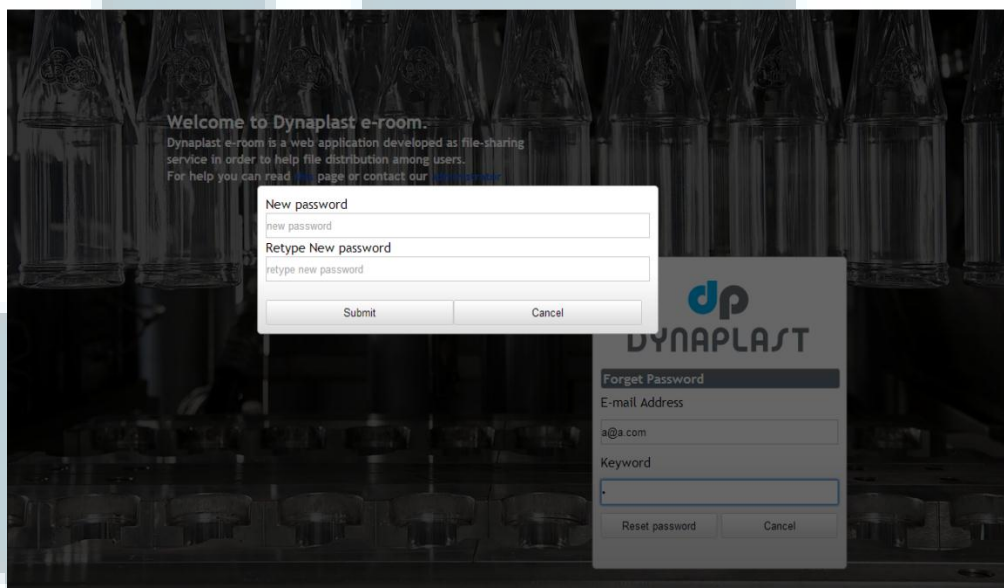
Gambar 3.13 Halaman *Forget Password Prototype* Kedua

Mengikuti halaman *login prototype* kedua, halaman *forget password prototype* kedua juga diubah dengan mengecilkan daerah *text field*.



Gambar 3.14 Halaman *Forget Password* Akhir

Penyesuaian kembali dilakukan pada halaman *forget password*. Perubahan ini dilakukan sesuai dengan perubahan yang dilakukan pada halaman *login*.



Gambar 3.15 *Forget Password Form*

Jika alamat surat elektronik dan *keyword* benar, maka pengguna akan dibawa kepada *form* untuk memasukkan *password* yang baru, jika salah maka akan

muncul pesan kesalahan mengenai informasi tidak benar yang dimasukkan oleh pengguna.

### 3. Halaman *First Setting*

Halaman *first setting* akan ditampilkan bila pengguna belum pernah menentukan *keyword*. Secara *default keyword* pada setiap akun pengguna adalah kosong.

PT DYNAPLAST

Welcome to Dynaplast e-Room  
It's seems that you had not set your keyword  
Keyword is used if you forget your password

New password  
new password

Retype New password  
retype new password

New keyword  
new keyword

Retype new keyword  
retype new keyword

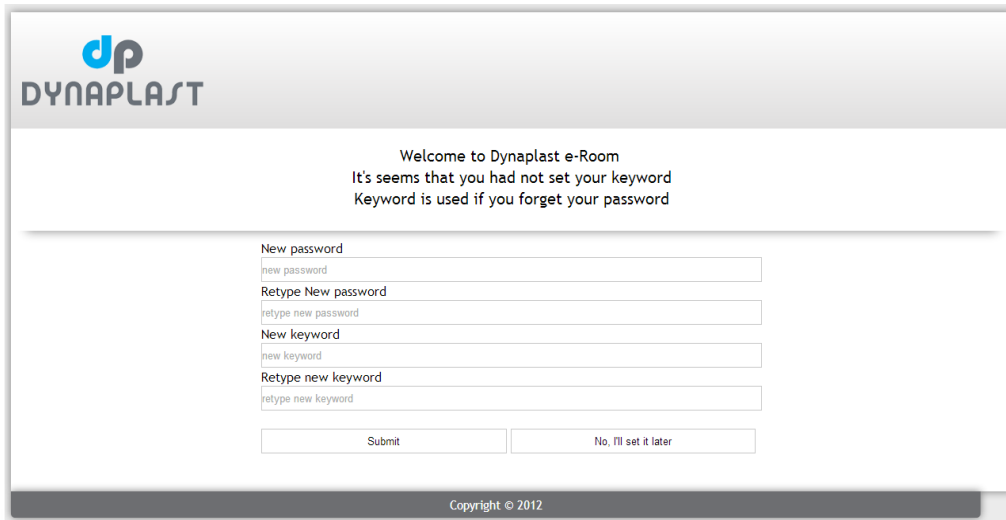
Submit No, I'll set it later

Copyright © 2012

Gambar 3.16 Halaman *First Setting Prototype* Pertama

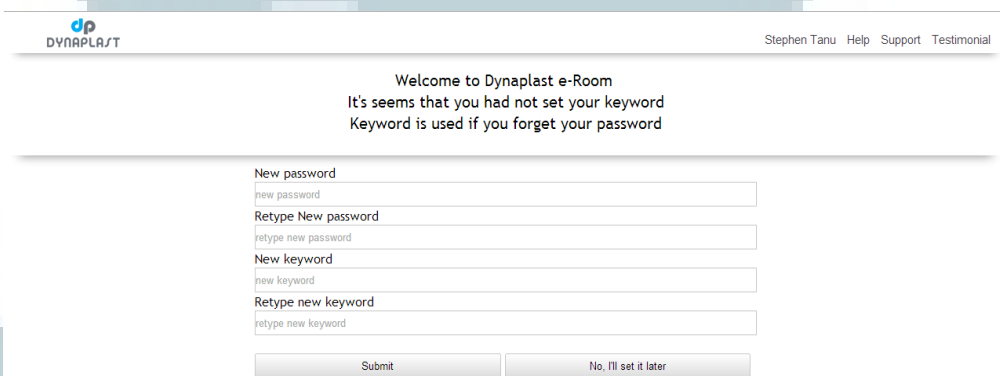
Pada halaman *first setting prototype* pertama, logo dan warna yang digunakan masih menggunakan gambaran perusahaan yang lama, namun selebihnya sudah disetujui.





Gambar 3.17 Halaman *First Setting Prototype Kedua*

Pada halaman *first setting prototype* kedua, logo dan warna yang digunakan sudah menggunakan gambaran perusahaan yang baru, namun logo dianggap terlalu besar sehingga menutupi layar pengguna maka perlu diperkecil.



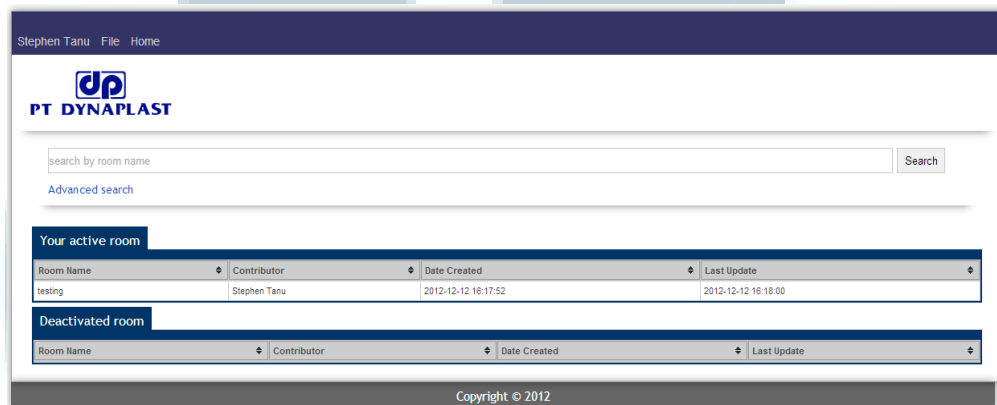
Gambar 3.18 Halaman *First Setting Akhir*

Pada halaman *first setting* akhir ukuran logo telah diperkecil serta ditambahkan navigasi di bagian atas.

#### 4. Halaman *Room Manager*

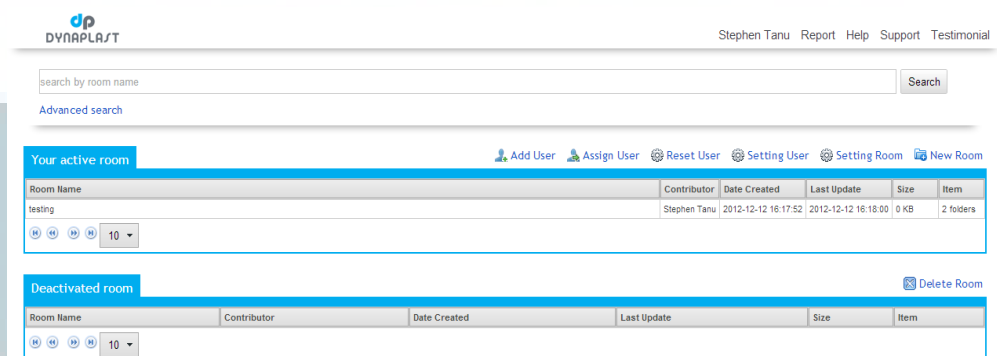
Halaman *room manager* berfungsi untuk menampilkan *room* yang telah dibuat oleh pengguna yang telah terdaftar dan memiliki akses untuk fungsi

penambahan *room*. *Room* yang ditampilkan pada halaman ini telah melalui proses penyaringan sehingga hanya pengguna yang berhak yang dapat melihat *room* dimana pengguna didaftarkan. Sesuai dengan *user requirement*, *room* dibagi dua jenis, aktif dan nonaktif, begitu pula *interface* menampung kebutuhan itu.



Gambar 3.19 Halaman *Room Manager Prototype*

Pada halaman *Room Manager Prototype* pertama terdapat navigasi pada bagian atas, pencarian pada bagian tengah, dan bagian pendaftaran *room*, baik yang aktif maupun yang non aktif.



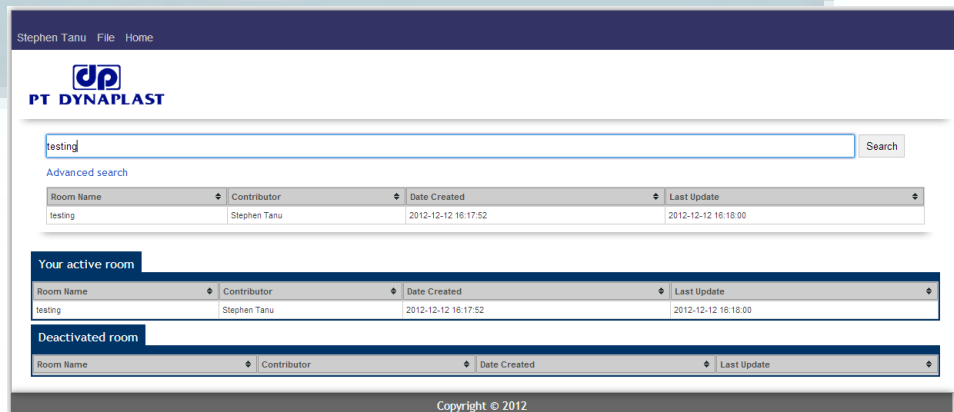
Gambar 3.20 Halaman *Room Manager Akhir* dan Menu Administrator

Pada halaman *room manager* akhir, warna dan logo disesuaikan dengan gambaran perusahaan yang baru sesuai dengan pengarahannya dalam bentuk

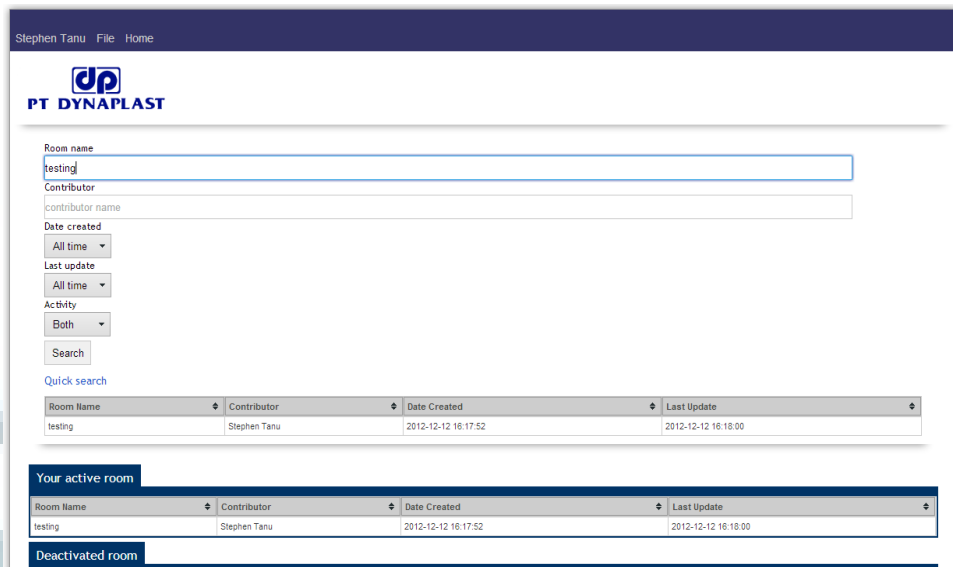
dokumen digital yang telah diberikan. Selain itu, halaman ini juga dilengkapi oleh *pagination* untuk memudahkan pengguna dalam melihat *room* yang ada. Informasi yang ditampilkan juga ditambah. Pada bagian desain *room manager* akhir dapat dibuka dengan *double click* pada baris *room* yang akan dibuka. Pada bagian *room manager* dapat dibagi menjadi beberapa bagian yang lebih kecil. Berikut adalah penjelasan dan gambar perkembangan bagian tersebut.

a. Bagian Pencarian

Pada bagian pencarian terdapat dua fungsi pencarian yang ada, pencarian pertama adalah pencarian cepat dimana pengguna cukup memasukkan nama atau bagian dari nama *room* yang ingin dicari, dan pencarian kedua adalah pencarian lanjutan dimana pengguna diberikan opsi yang lebih banyak sehingga pencarian menjadi lebih terarah.

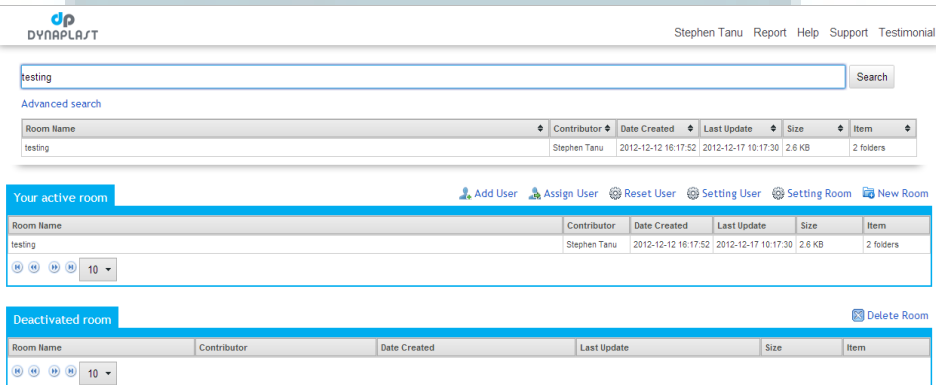


Gambar 3.21 *Room Manager Quick Search Prototype*

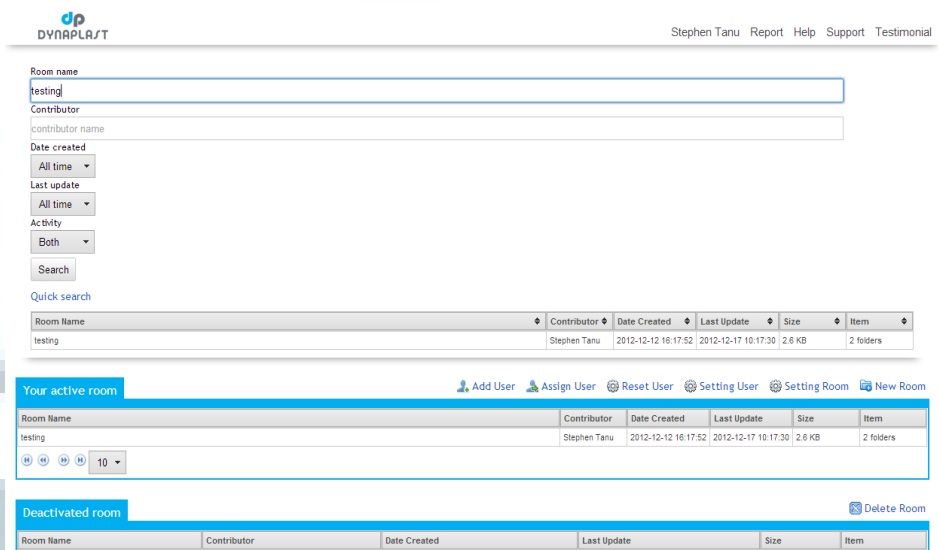


Gambar 3.22 Room Manager Advanced Search Prototype

Pada bagian pencarian *prototype* ini masih menggunakan logo dan warna lama dari gambaran perusahaan sehingga perlu diubah.



Gambar 3.23 Room Manager Quick Search Akhir

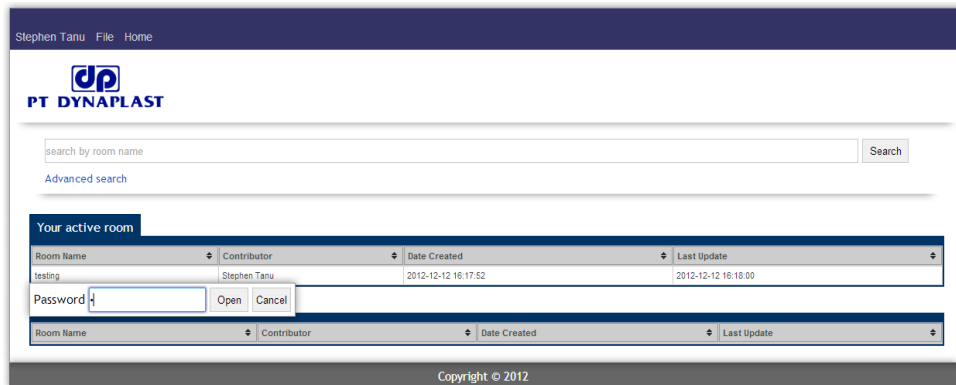


Gambar 3.23 Room Manager Advanced Search Akhir

Pada bagian pencarian akhir ini telah menggunakan logo dan warna baru dari gambaran perusahaan. Selain itu, pada sisi sistem diubah sehingga pencarian akan mementingkan *room* yang memiliki nama terlebih dahulu, dan yang menjadi bagian nama *room* berikutnya.

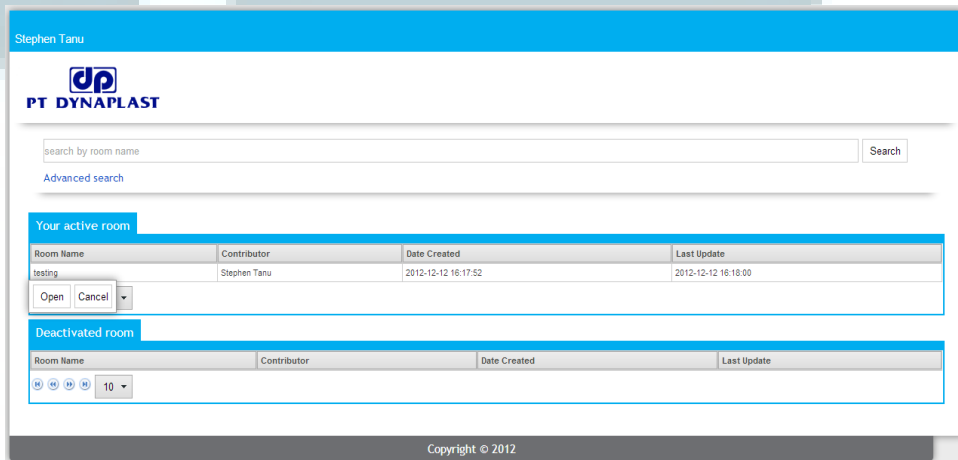
b. Bagian Opsi Menu *Room*

Bagian opsi menu *room* ini berfungsi menjadi daftar menu operasi yang bisa dikerjakan terhadap *room* dimana opsi dibuka. Opsi *room* ini berubah sesuai dengan perubahan *user requirement*. Opsi menu *room* pada bagian akhir dapat dibuka dengan *right click* pada baris *room* yang akan menjadi target operasi.



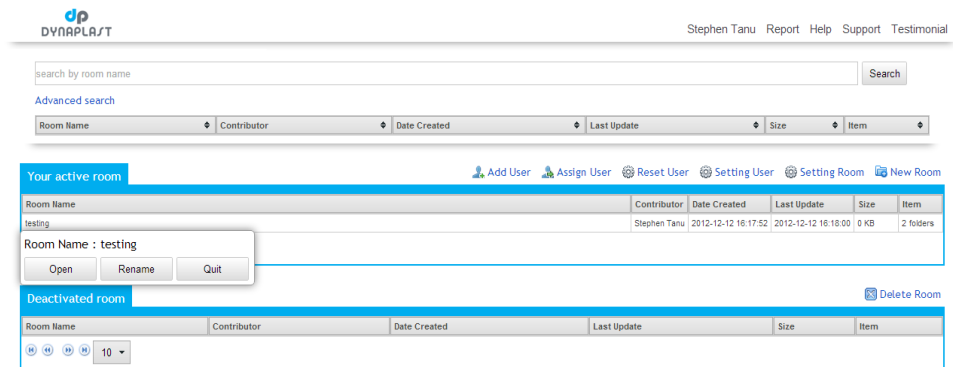
Gambar 3.24 Opsi Menu *Room Prototype* Pertama

Pada opsi menu *room prototype* pertama, *user requirement* pada saat itu adalah akses setiap *room* dibatasi dengan pemberian *password* pada setiap *room* yang ada.



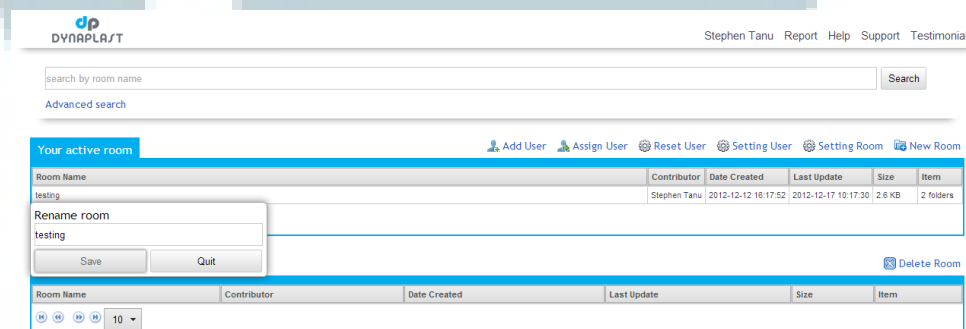
Gambar 3.25 Opsi Menu *Room Prototype* Kedua

Pada opsi menu *room prototype* kedua, penggunaan *password* pada *room* sudah tidak digunakan, diganti dengan pemberian akses pada pengguna.



Gambar 3.26 Opsi Menu *Room* Akhir Administrator

Pada opsi menu *room* akhir, ditambahkan operasi *rename* pada *room*, pada opsi menu *roomakhir* ini juga ditambahkan komponen *rename room form* yang terbuka jika tombol *rename* ditekan. *Form* yang dimaksud sebagai berikut.



Gambar 3.27 *Rename Room Form*

Jika nama yang dimasukkan oleh pengguna benar, maka akan diberikan pesan berhasil, dan jika gagal akan ditampilkan pesan gagal, sesuai dengan kegagalan yang dilakukan.

c. Bagian Menu Operasi *Room*

Bagian menu operasi *room* terdiri dari beberapa operasi dimana masing-masing menu ini memiliki *form* tersendiri. Berikut adalah menu dan *form* dengan tingkatan akses administrator.

The image shows a web application interface with a menu bar containing 'Add User', 'Assign User', 'Reset User', 'Setting User', 'Setting Room', and 'New Room'. Below the menu is a table with two columns: 'Contributor' and 'Last Update'. The 'Contributor' column contains the name 'Stephen Tanu'. Overlaid on the right side of the table is a modal form titled 'E-mail'. The form contains the following fields:
 

- 'E-mail': A text input field containing 'email@domain.com'.
- 'Name': Two text input fields, the first labeled 'first name' and the second labeled 'middle, last name'.
- 'Company Name': A text input field containing 'company name'.
- 'Positon': A dropdown menu with 'Viewer' selected.
- 'Add' and 'Quit': Two buttons at the bottom of the form.

Gambar 3.28 *Add User Form*

*Form* ini berguna bagi administrator untuk melakukan penambahan pengguna. Pada aplikasi ini, sesuai dengan *user requirement*, semua pengguna hanya bisa ditambahkan oleh administrator. Jika informasi yang dimasukkan benar, maka aplikasi akan mengirim surat elektronik kepada alamat yang didaftarkan dimana isi surat elektronik itu adalah *password* dasar hasil acakan aplikasi, dan jika terdapat kesalahan maka akan ditampilkan pesan kesalahan seperti pada *form* lainnya.

U  
M  
M  
N



Gambar 3.29 Assign User Form

Form ini berfungsi untuk memberikan hak akses terhadap satu atau sekelompok pengguna pada suatu *room* tertentu. Fungsi ini dapat diakses oleh administrator dan *project leader*.

Gambar 3.30 Reset User Form

Form ini berfungsi untuk mengubah *password* pengguna menjadi huruf-huruf yang diacak oleh aplikasi dan hanya bisa digunakan oleh pengguna yang memiliki akses administrator.

The image shows a web application interface with a navigation bar containing icons and labels for 'Add User', 'Assign User', 'Reset User', 'Setting User', 'Setting Room', and 'New Room'. Below the navigation bar is a table with two columns: 'Contributor' and 'Last Update'. The first row of the table contains the name 'Stephen Tanu'. A modal window titled 'Setting User' is overlaid on the table. The modal contains the following fields and controls:

- Email:** A dropdown menu with a '-' symbol.
- User activity:** Two radio buttons, 'active' (selected) and 'deactive'.
- Name:** Two text input fields labeled 'first name' and 'middle, last name'.
- Company Name:** A text input field labeled 'company name'.
- Position:** A dropdown menu with 'Viewer' selected.
- Buttons:** 'Save' and 'Quit' buttons at the bottom.

Gambar 3.31 *Setting User Form*

*Form* ini berfungsi agar administrator dapat mengubah informasi pengguna yang sebelumnya sudah terdaftar. Jika akun dinonaktifkan, maka secara otomatis *password* akun tersebut akan diubah menjadi huruf acakan aplikasi.

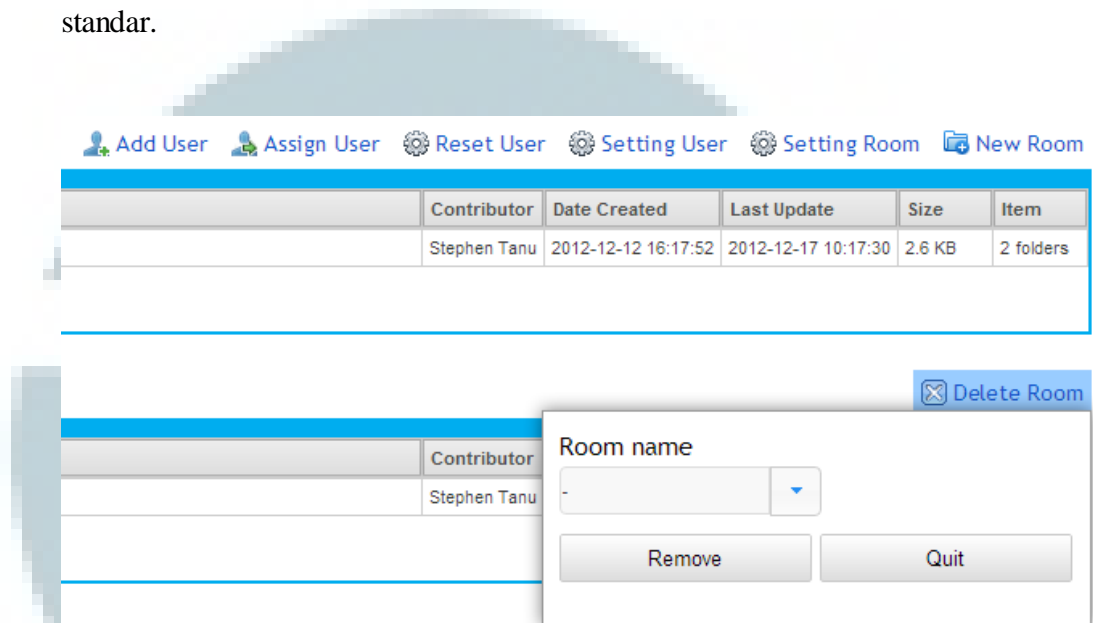
UMMN

Gambar 3.32 Setting Room Form

Form ini berfungsi bagi administrator untuk mengubah pengaturan pada sebuah room atau seluruh room yang telah dibuat sebelumnya.

Gambar 3.33 New Room Form

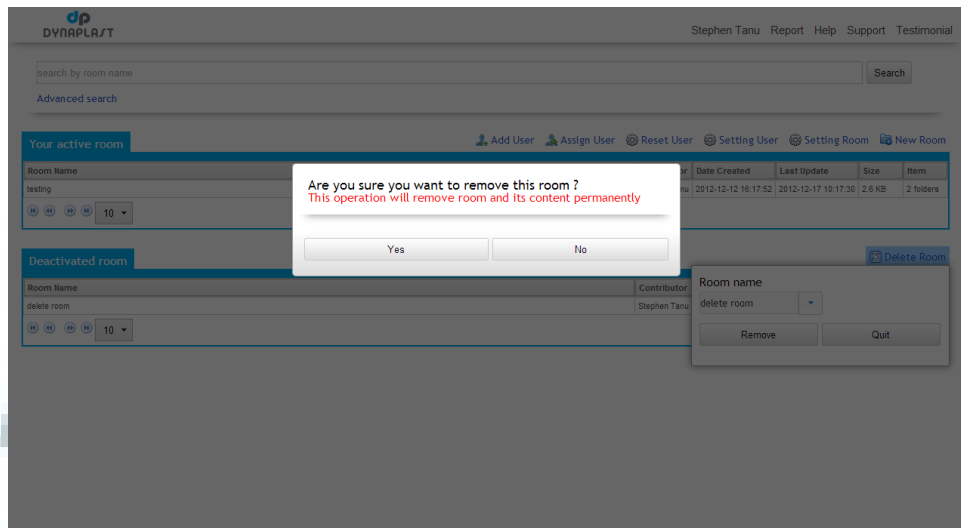
*Form* ini berfungsi untuk melakukan penambahan *room*. Selain batas maksimum dari *room* dan batas ukuran yang bisa diunggah per *file*, seluruh opsi pengaturan *room* yang lainnya adalah pengaturan *room* standar.



Gambar 3.34 Delete Room Form

*Form* ini berfungsi bagi administrator untuk menghapus *room* dengan status nonaktif sesuai dengan *user requirement*. Hal ini perlu dilakukan dengan alasan keterbatasan ukuran media penyimpanan pada *server* yang disediakan.

U  
M  
M  
N



Gambar 3.35 Konfirmasi *Delete Room*

Sebelum *room* benar-benar dihapus secara permanen akan ditampilkan konfirmasi kembali, hal ini dilakukan untuk memastikan bahwa pengguna sadar dan mengetahui sehingga tidak terjadi kesalahan.

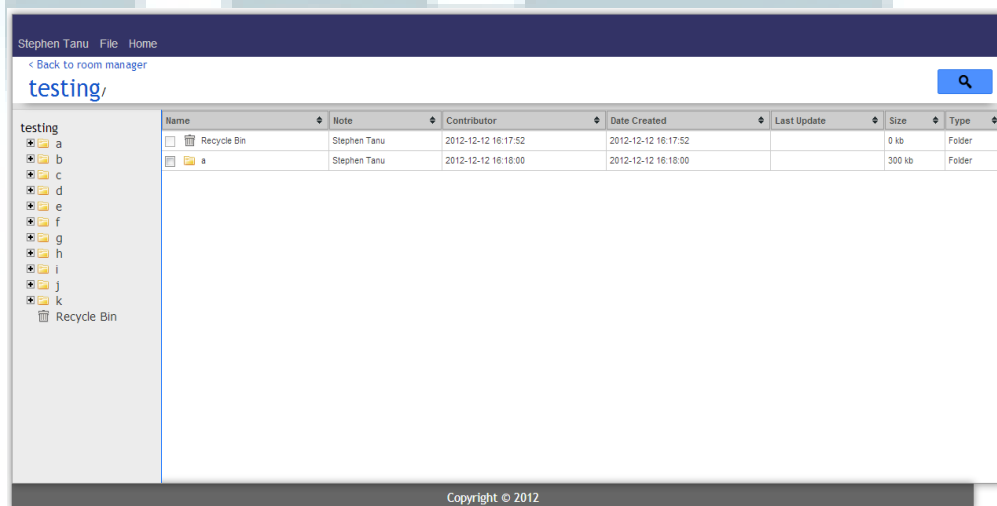
#### 5. Halaman *File Manager*

Halaman *file manager* berfungsi untuk menampilkan seluruh *folder* yang telah dibuat dan *file* yang telah diunggah sebelumnya. Menurut *user requirement*, *folder* dibagi menjadi dua jenis dimana terdapat *folder* yang bisa diakses oleh semua pengguna yang memiliki akses terhadap *room* dimana *folder* berada, dan *folder* yang hanya bisa diakses oleh pengguna tertentu tanpa memedulikan tingkatan aksesnya.

Terdapat beberapa pola navigasi yang bisa dilakukan pada halaman desain akhir *file manager* dalam penggunaannya. Berikut adalah beberapa pola interaksi yang bisa dilakukan.

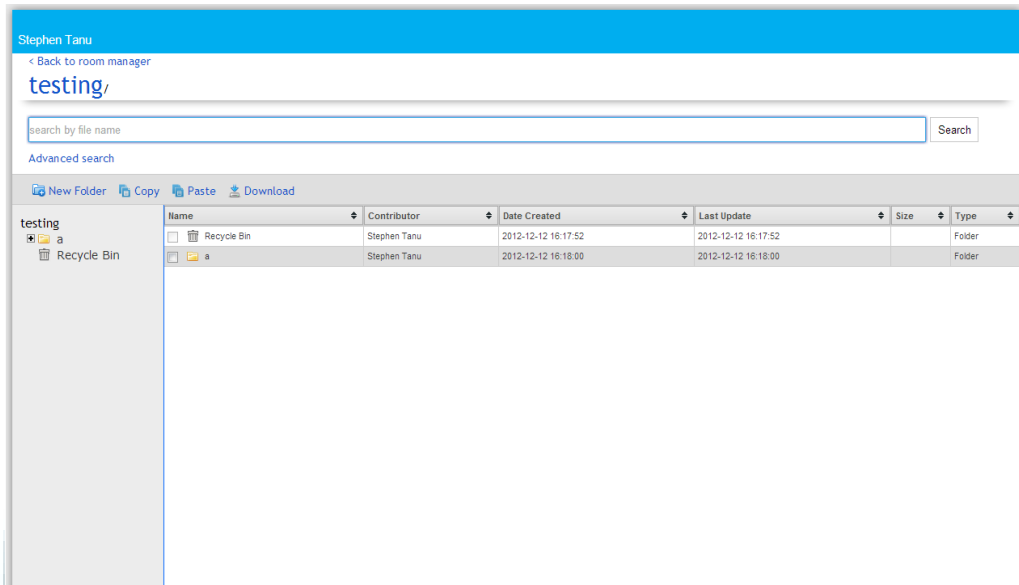
- a. Membuka *folder* dengan menggunakan *double click* pada baris *folder* yang akan dibuka.
- b. Membuka *folder* dengan melakukan *click* pada *tree* yang ada di sisi kiri.
- c. Membuka *folder* melalui opsi menu *folder*.

- d. Naik satu tingkat direktori dengan *click* pada *address bar* dari aplikasi di atas bagian pencarian.
- e. Mengunduh *file* dengan menggunakan *double click* pada baris *file* yang ingin diunduh.
- f. Mengunduh *file* dengan melakukan *click* pada *tree* di sisi kiri.
- g. Mengunduh *file* melalui opsi menu *file*.
- h. Menampilkan gambar beberapa format *file* gambar yang telah diunggah dengan melakukan *click* pada *iconfile*.
- i. Memberikan *check* yang ada di kiri dari nama *folder* atau *file* untuk dilakukan operasi sesuai dengan tingkatan pengguna.



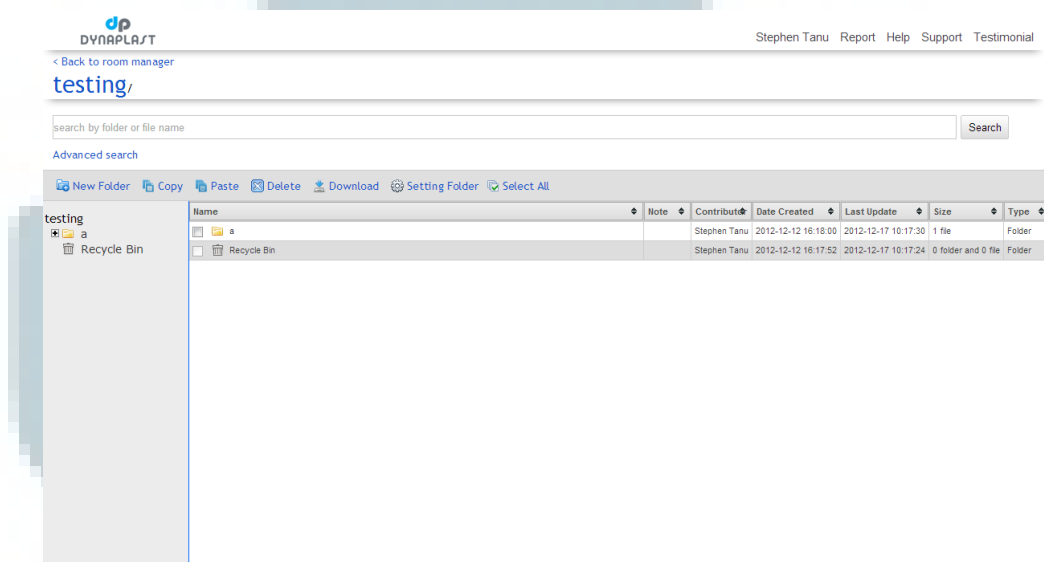
Gambar 3.36 Halaman *File Manager Prototype* Pertama

Halaman *file manager prototype* pertama memiliki kekurangan dalam bidang pencarian, dimana pada *prototype* pertama ini hanya bisa dilakukan pencarian berdasarkan nama *folder* atau *file* yang akan dicari. Selain itu, seperti pada kebanyakan *prototype* pertama lainnya, *prototype* ini masih menggunakan gambaran lama perusahaan.



Gambar 3.37 Halaman *File Manager Prototype* Kedua

Pada halaman *file manager prototype* kedua terjadi perubahan pada bagian pencarian seperti pada *room manager*. Pencarian pada *file manager* juga ada dua yaitu cepat dan lanjutan, namun pada halaman ini juga perlu ditambahkan logo perusahaan sehingga perlu diubah.



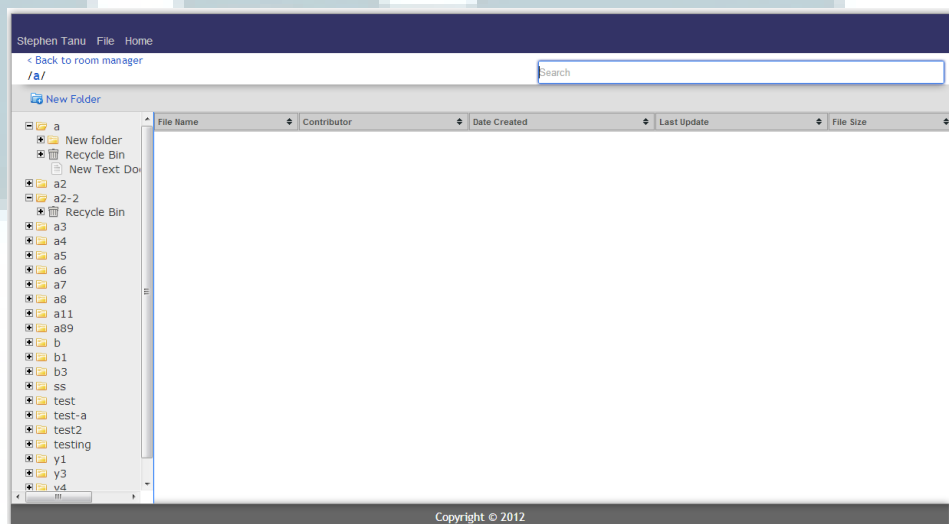
Gambar 3.38 Halaman *File Manager* Akhir

Halaman *file manager* akhir mengalami perubahan pada penampilan urutan kolom *size* untuk *folder*, dimana ditampilkan jumlah *folder* dan *file* yang ada di dalam *folder* tersebut. Selain itu, penambahan logo serta perubahan letak menu navigasi ke kiri juga dilakukan.

Pada bagian *file manager* dapat dibagi menjadi beberapa bagian yang lebih kecil. Berikut adalah penjelasan dan gambar perkembangan bagian tersebut.

a. Bagian pencarian

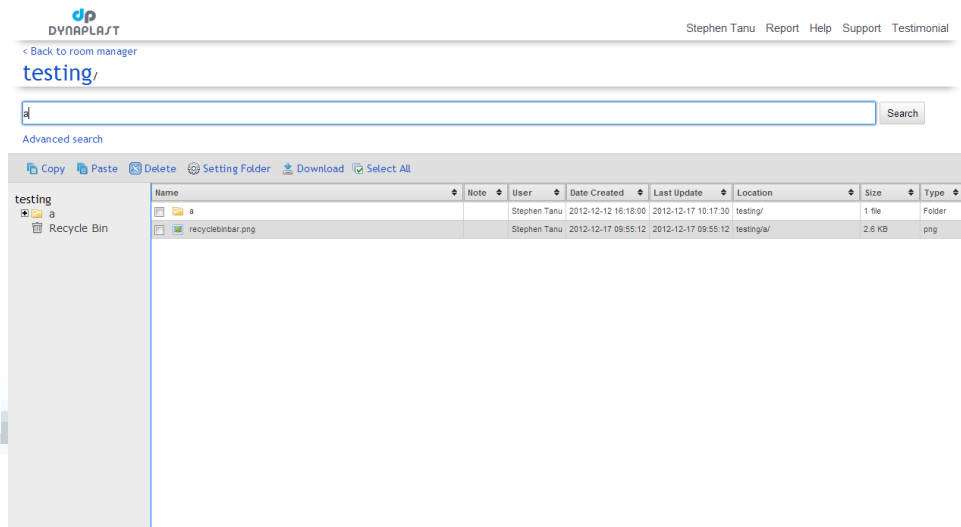
Bagian pencarian pada *file manager* seperti pada *room manager*, namun pencarian dilakukan pada *folder* dan atau *file* secara bersamaan. Hasil dari pencarian ditunjukkan pada sisi kanan dari *tree*.



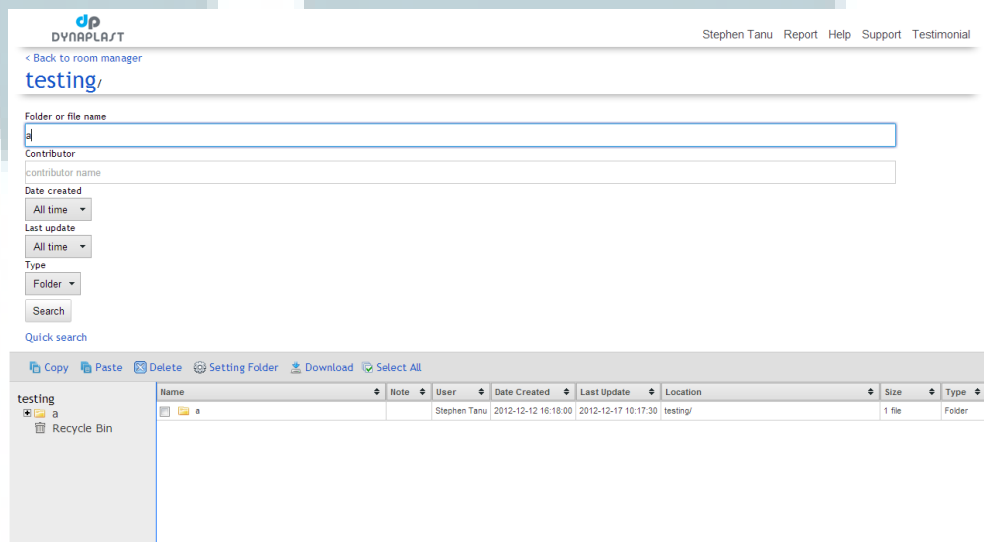
Gambar 3.39 *File Manager Quick Search Prototype*

Pada *prototype* pertama, pencarian yang dapat dilakukan hanya pencarian cepat dimana hanya nama dari *folder* dan *file* yang dicocokkan.





Gambar 3.40 File Manager Quick Search Akhir

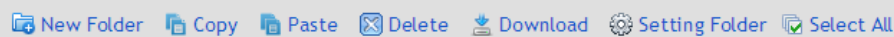


Gambar 3.41 File Manager Advanced Search Akhir

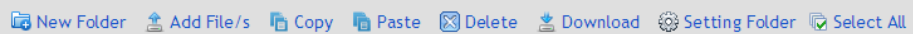
Pada tahap desain akhir, pada bagian pencarian yang ada pada halaman *room manager* terdapat dua jenis pencarian yang bisa dilakukan baik pencarian cepat maupun pencarian lanjutan, namun pada bagian pencarian lanjutan *file manager*, ditambahkan opsi pencarian mengenai jenis objek yang akan dicari.

b. Bagian *Operation Box*

Bagian *operation box* berfungsi untuk menampilkan menu mengenai operasi yang bisa dilakukan pada *file manager*, operasi yang ditampilkan tergantung dari tingkatan akses pengguna. Bagian *operation box* ini juga berbeda untuk masing–masing jenis dimana pengguna berada.



Gambar 3.42 *Operation Box* Tingkatan Pertama Direktori (Administrator)



Gambar 3.43 *Operation Box* Tingkatan Kedua Direktori dan Selanjutnya (Administrator)



Gambar 3.44 *Operation Box* Pencarian (Administrator)



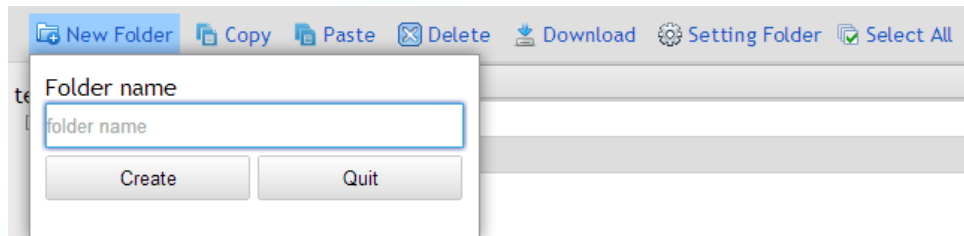
Gambar 3.45 *Operation Box Recycle Bin* (Administrator)

Pada *operation box* tingkatan pertama jika dibandingkan dengan tingkatan kedua dan selanjutnya, tidak memiliki menu untuk mengunggah *file*. Hal ini dilakukan untuk menyesuaikan dengan *user requirement* dimana pada tingkatan pertama tidak diperbolehkan adanya *file*.

c. Bagian *Operation Form*

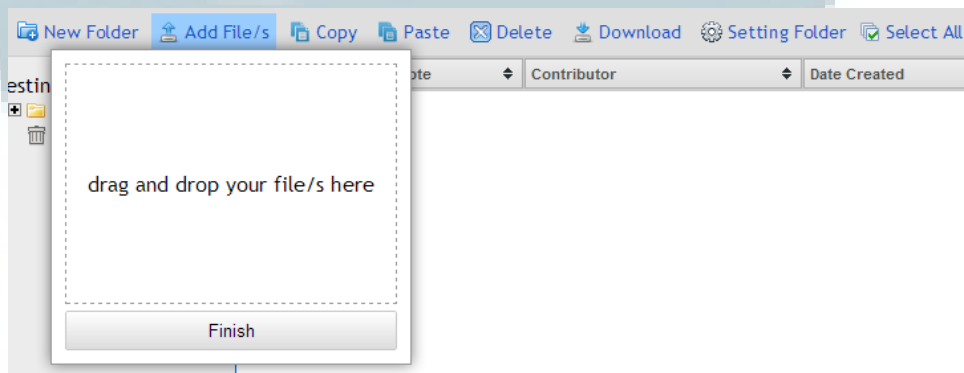
Dari menu operasi yang ada pada bagian *operation box*, terkadang diperlukan informasi dari pengguna sehingga operasi dapat berjalan dengan benar. Untuk itu, sebagai wadah informasi yang dimasukkan pengguna, *operation form* ini diperlukan. Berikut beberapa *form* yang

dapat ditampilkan dengan *click* pada menu yang ada pada halaman *file manager*.

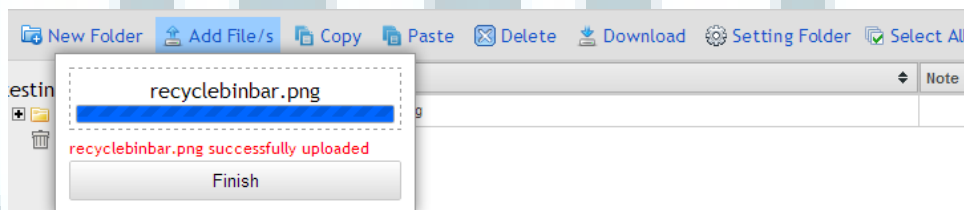


Gambar 3.46 *New Folder Form*

*New folder form* berfungsi untuk menambahkan *folder* baru pada tingkatan direktori dimana pengguna berada pada saat itu.



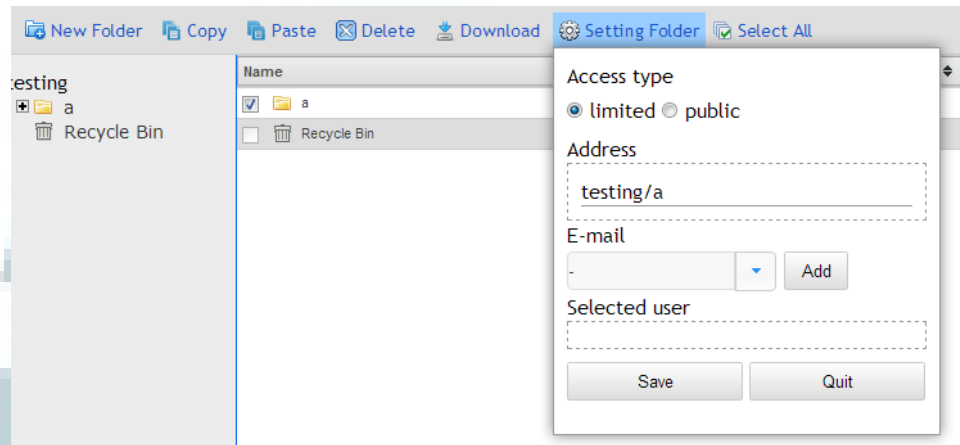
Gambar 3.47 *Form Pengunggahan File*



Gambar 3.48 *Pengunggahan File Berhasil*

*Form* pengunggahan *file* berguna bagi pengguna untuk menambahkan *file* yang dimiliki ke dalam *server* dengan melakukan *drag and drop* sebuah

atau beberapa *file* dari komputer ke area pada aplikasi yang telah ditentukan.

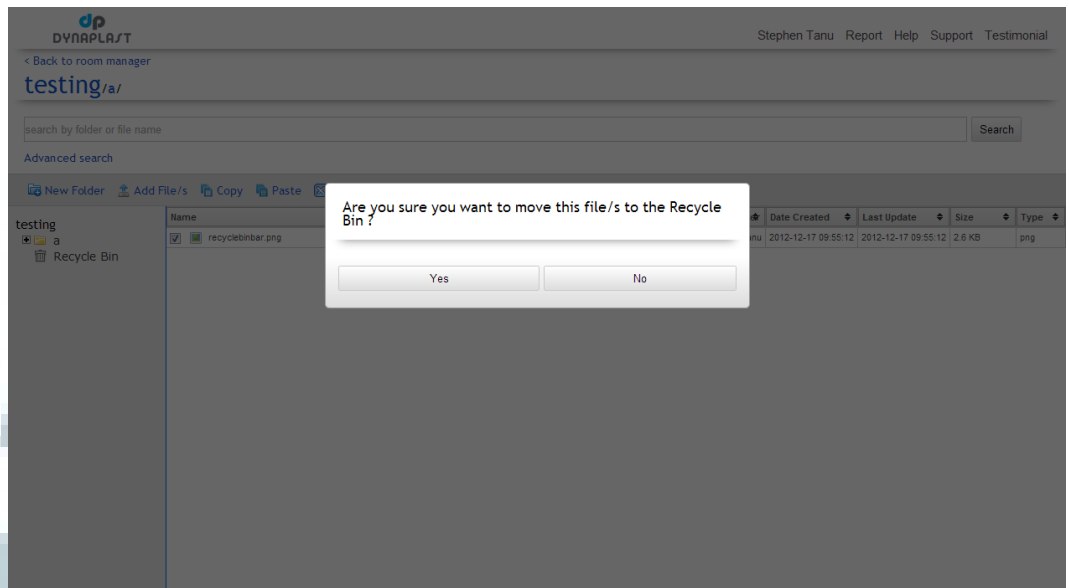


Gambar 3.49 Form Pengaturan Folder

Form pengaturan *folder* berfungsi untuk mengubah tipe akses dari sebuah *folder*. Jika dapat dilihat oleh semua *user* yang memiliki akses terhadap *room* dapat dipilih *public*, dan jika ingin dibatasi pada pengguna yang telah ditentukan, maka dipilih *limited*.

d. Bagian Kotak Dialog *Operation*

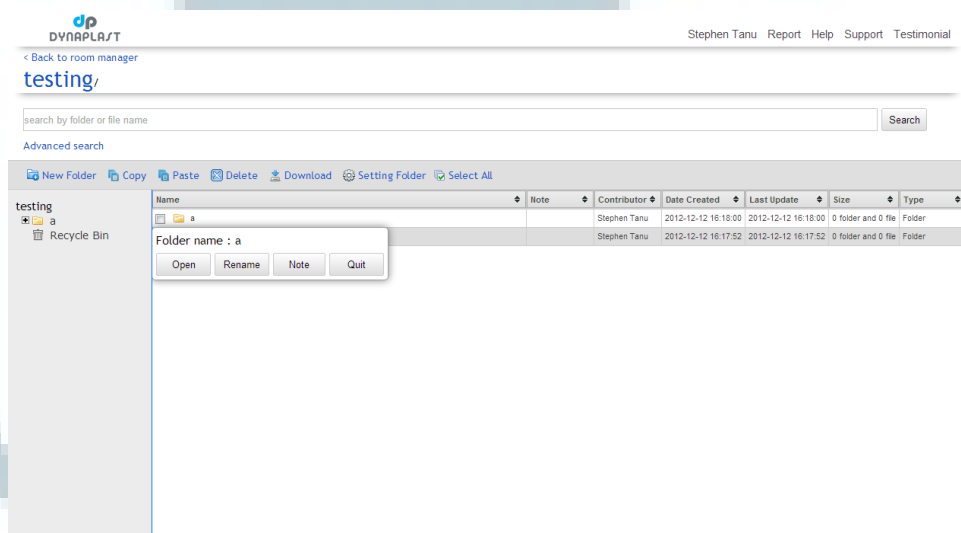
Selain dalam bentuk *form*, pengguna juga diberikan kotak dialog jika operasi yang dilakukan adalah operasi yang permanen dan dapat berakibat fatal jika terjadi kesalahan. Berikut adalah dialog operasi *delete folder* atau *file*.



Gambar 3.50 Dialog Konfirmasi *Delete Folder dan File*

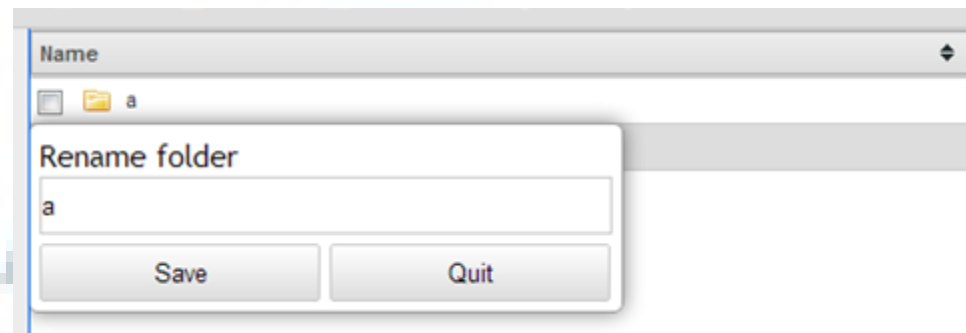
e. Bagian Opsi Menu *Folder dan file*

Bagain opsi menu *folder dan file* dapat diakses dengan melakukan *right click* pada baris *folder* atau *file* yang diinginkan. Opsi menu *folder dan file* berisi mengenai operasi yang bisa dilakukan pada objek yang dipilih.

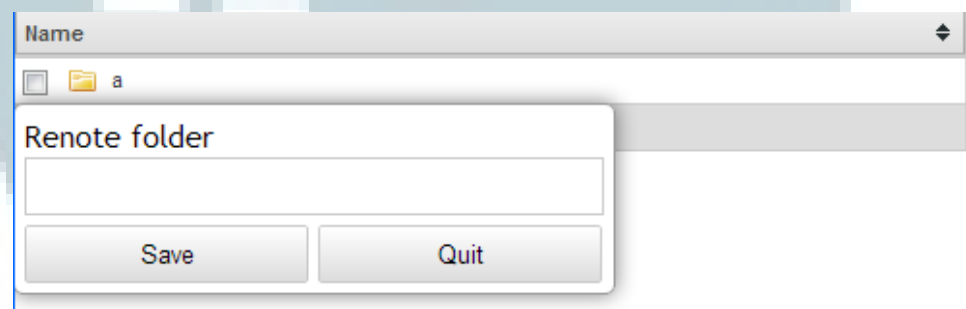


Gambar 3.51 Opsi Menu *Folder Administrator*

Berikut adalah *form* yang bisa diakses pengguna dengan melakukan *click* pada tombol *rename* pada opsi menu.



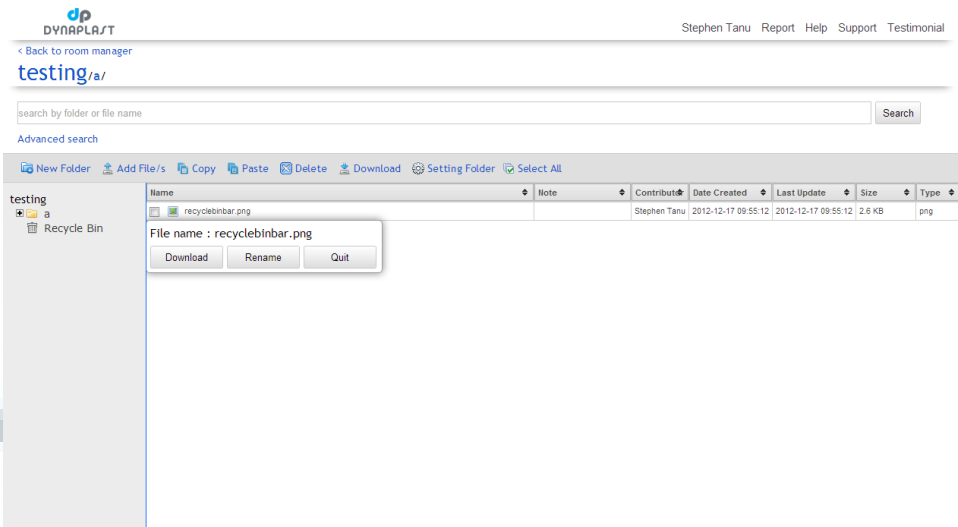
Gambar 3.52 *Form Rename Folder*



Gambar 3.53 *Form Rename Folder*

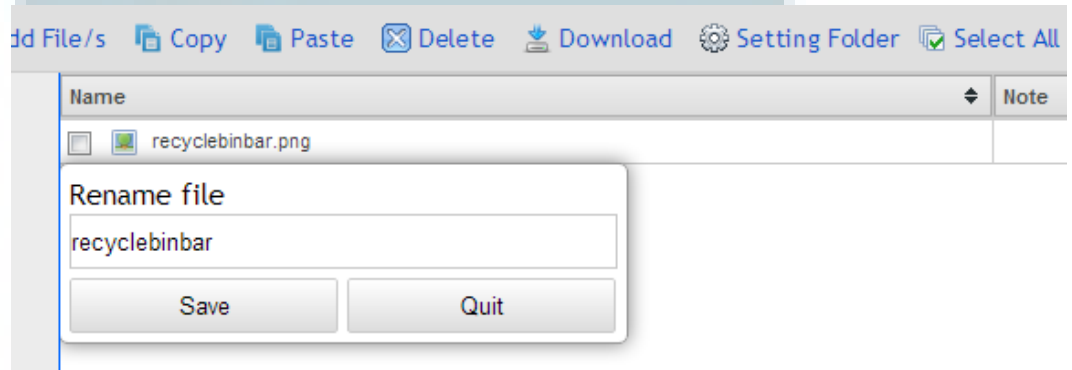
Berikut adalah opsi menu untuk *file* menu operasi yang ada disesuaikan dengan objek yang dipilih.

UMMN



Gambar 3.54 Opsi Menu *File Administrator*

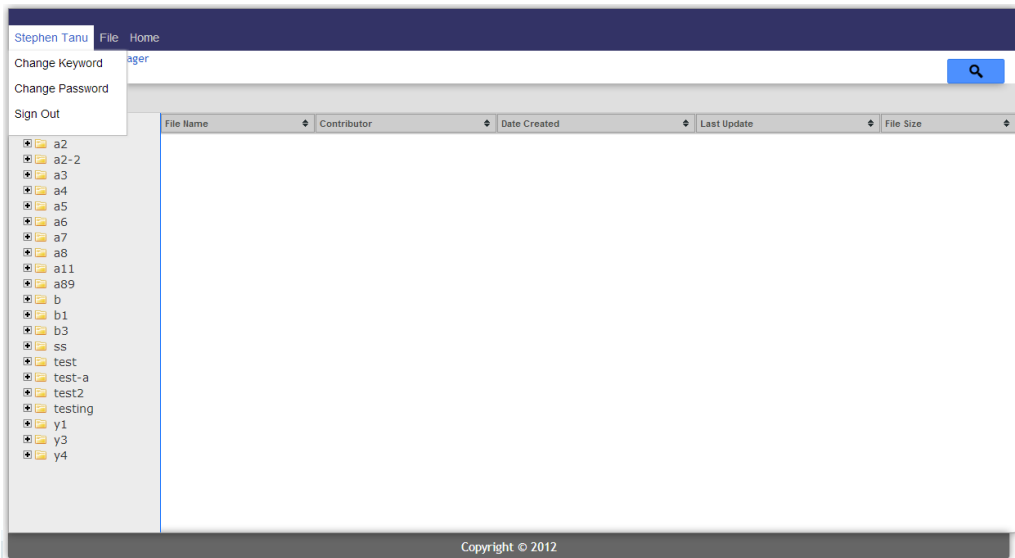
Berikut adalah *form rename file* yang dapat ditampilkan melalui *click* pada tombol *rename* pada opsi menu.



Gambar 3.55 *Form Rename File*

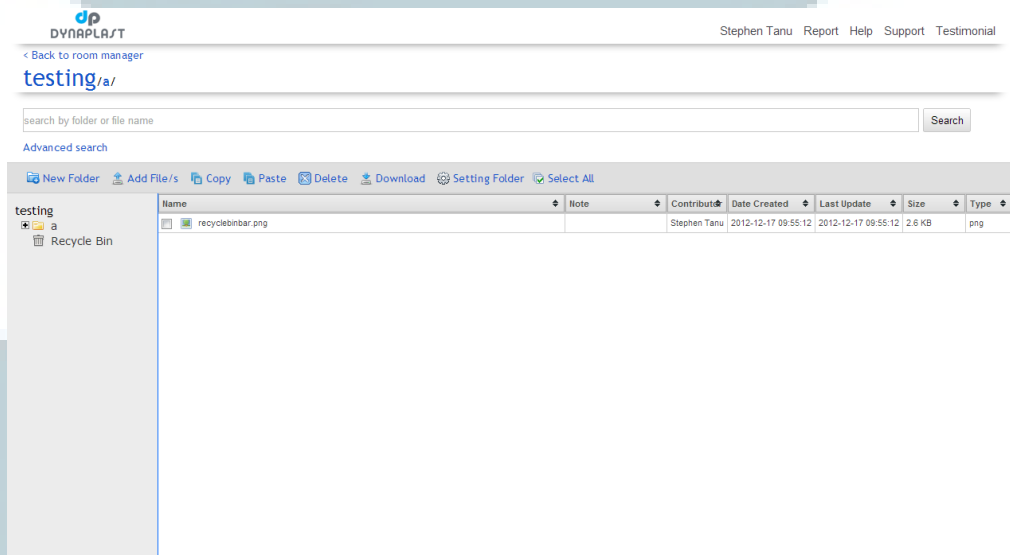
## 6. Menu Navigasi

Menu navigasi selalu terletak dibagian atas dari aplikasi bersama dengan logo Dynaplast.



Gambar 3.56 Menu Navigasi *Prototype* dan Sub Menu Pengguna

Pada *prototype* pertama letak navigasi masih ada di sebelah kanan dan tidak memiliki logo Dynaplast, sehingga perlu diubah.



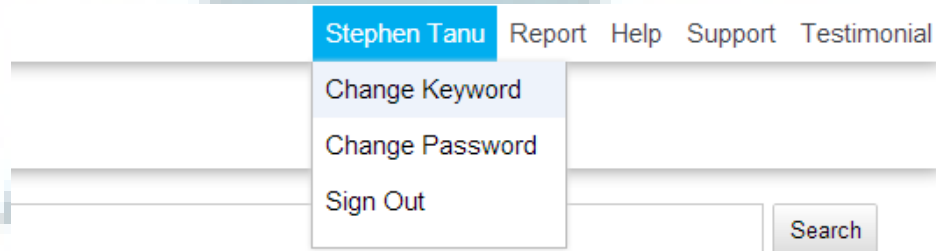
Gambar 3.57 Menu Navigasi Akhir

Menu navigasi akhir dilengkapi dengan logo perusahaan, dan perubahan letak serta penambahan menu sesuai dengan perubahan *user requirement* yang ada.



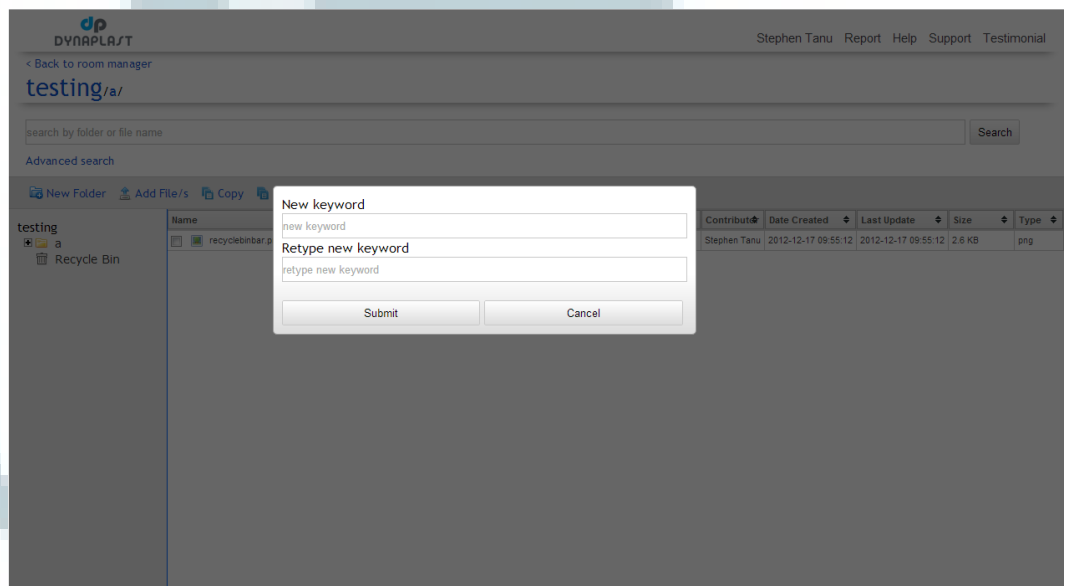
Berikut adalah sub menu navigasi yang dapat ditampilkan dengan meletakkan kursor pada menu yang ada.

a. Sub Menu Pengguna

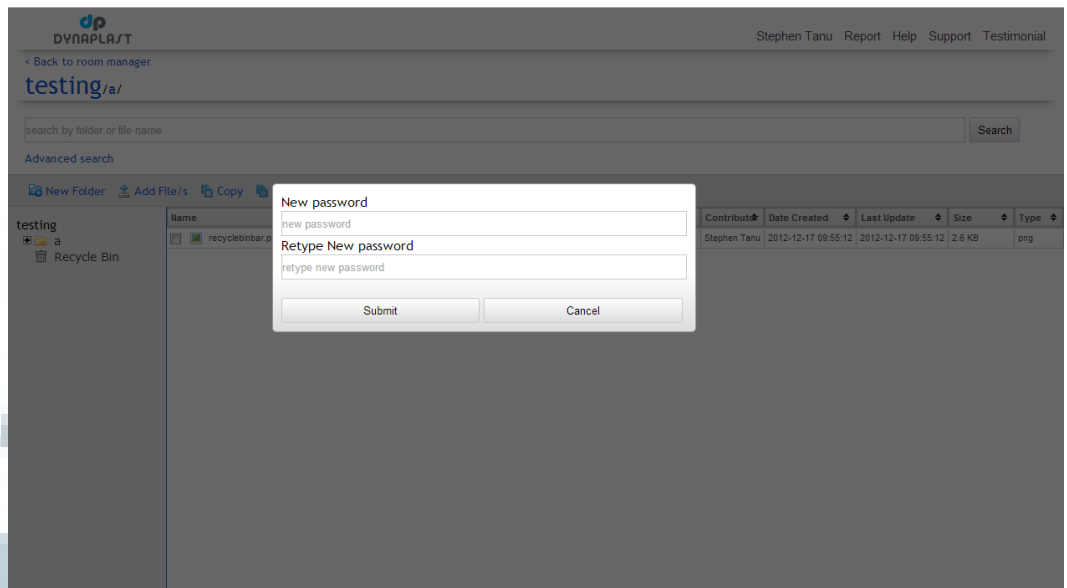


Gambar 3.58 Sub Menu Pengguna

Sub menu pengguna berfungsi untuk menjadi tautan dalam menjalankan fungsi perubahan *password*, *keyword*, dan keluar dari aplikasi. Jika pengguna melakukan *click* pada sub menu *change keyword* dan *change password*, maka *form* yang bersesuaian akan ditampilkan. Berikut adalah *form* yang dimaksud.

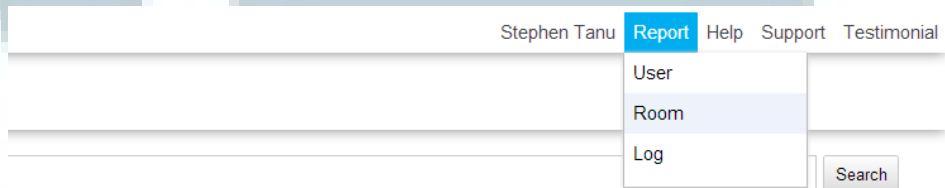


Gambar 3.59 Form Change Keyword



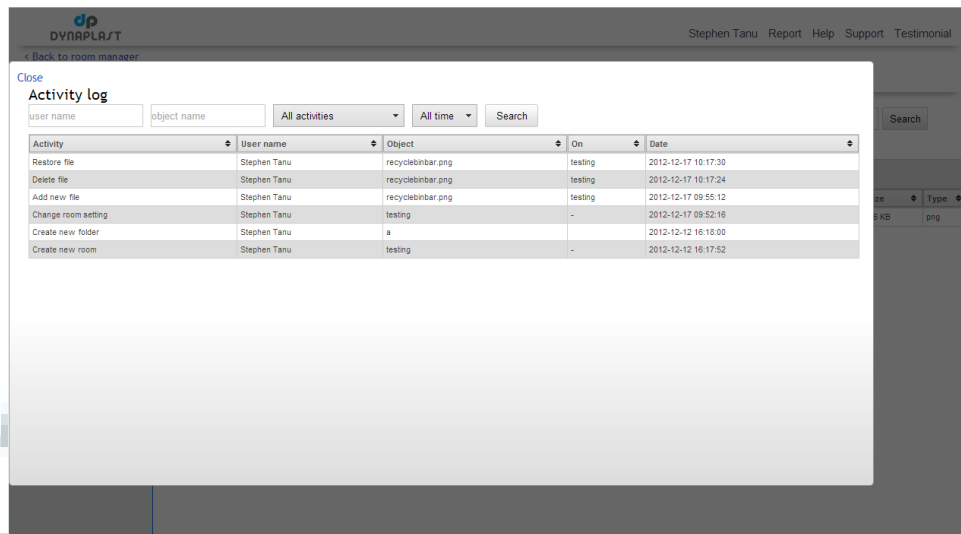
Gambar 3.60 *Form Change Password*

b. Sub Menu *Report*



Gambar 3.61 Sub Menu *Report*

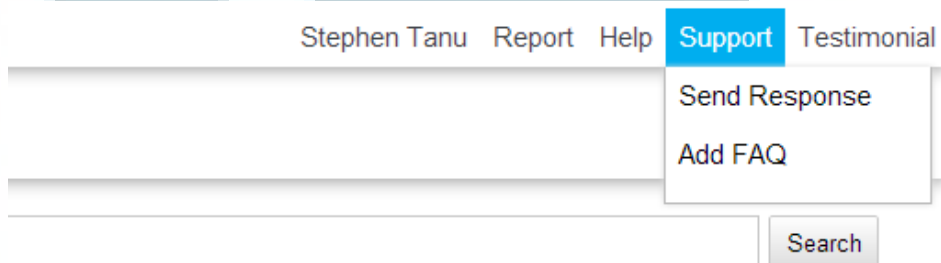
Sub menu *Report* berfungsi sebagai tautan untuk menampilkan log pengguna, *room*, dan aktifitas yang dilakukan oleh pengguna. Sub menu *report* hanya dapat diakses oleh pengguna dengan tingkatan akses administrator. Berikut adalah tampilan *report* log jika log dipilih.



Gambar 3.62 Report Log

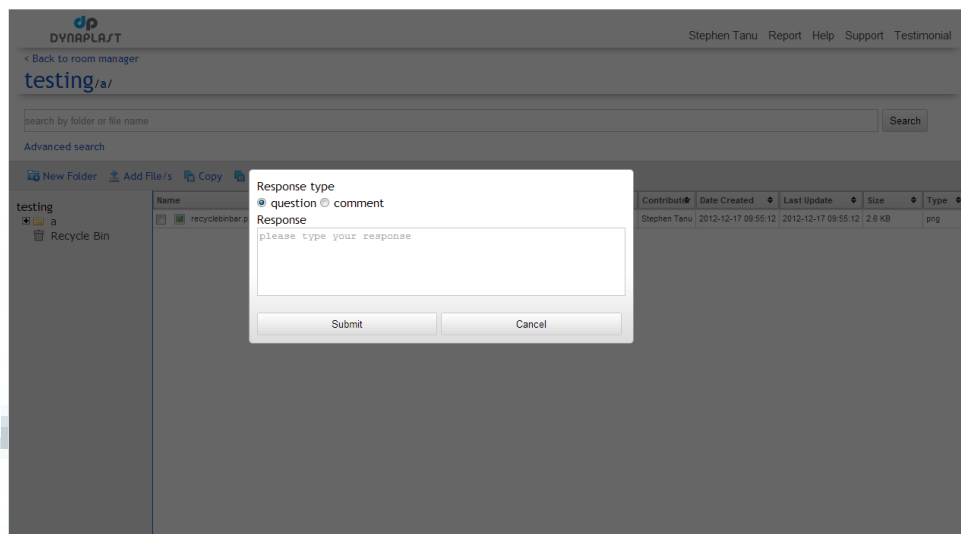
Pada *report log*, pengguna juga dapat melakukan pencarian sehingga dapat memudahkan dalam pembacaan log kegiatan yang terjadi.

c. Sub Menu *Support*



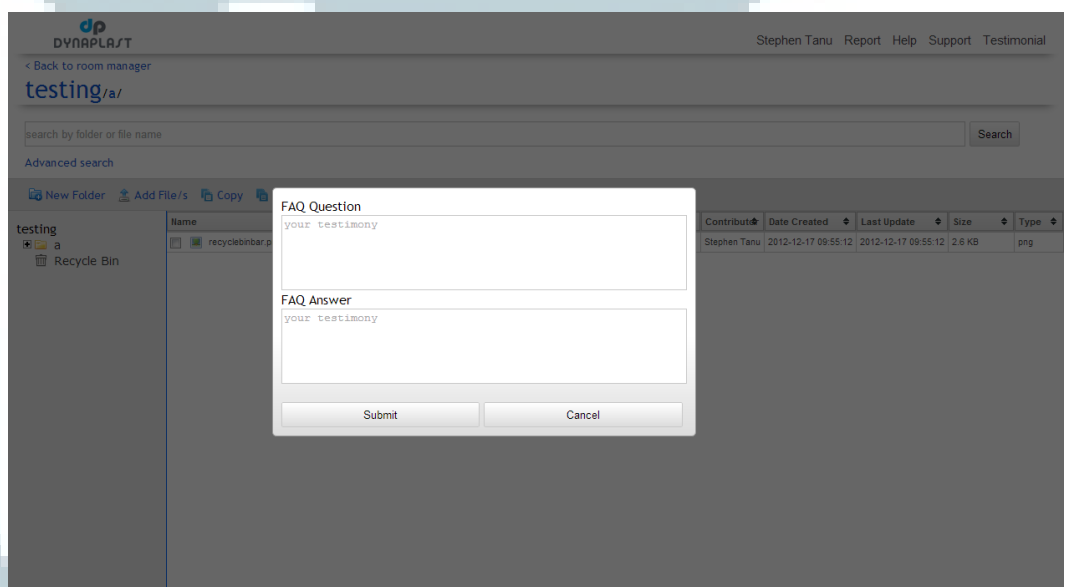
Gambar 3.63 Sub Menu *Support*

Sub menu *send response* berfungsi untuk melakukan operasi dimana pengguna dapat mengirimkan respon atas penggunaan aplikasi, baik berupa pertanyaan maupun komentar kepada administrator.



Gambar 3.64 *Send Response Form*

*Form send response* dapat diakses oleh seluruh pengguna, dengan tujuan utama pengguna dapat memberikan pertanyaan jika ada hal yang tidak dimengerti atau salah mengenai pengalaman dalam penggunaan aplikasi.

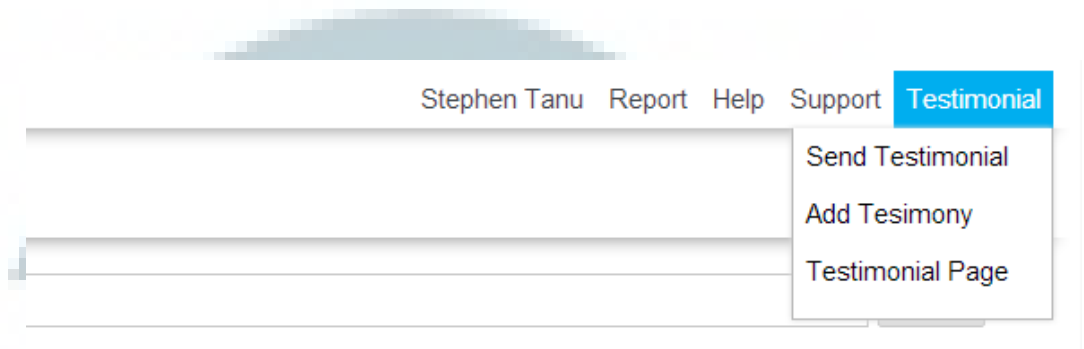


Gambar 3.65 *Add FAQ Form*

*Form add FAQ* hanya dapat diakses pengguna dengan tingkat akses administrator, dengan tujuan utama menambah informasi mengenai

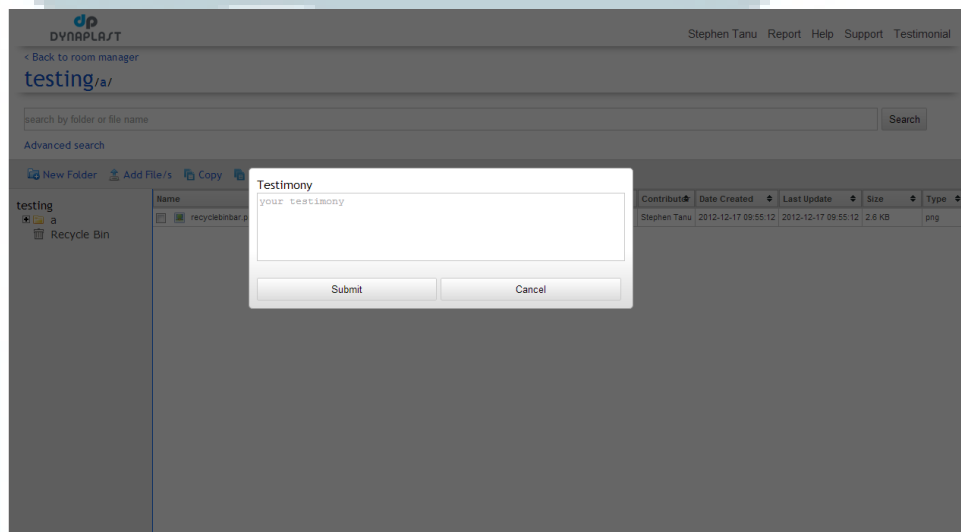
aplikasi, fungsi ini digunakan sebagai solusi karena fungsi *help* yang bersifat tetap.

d. Sub Menu *Testimonial*



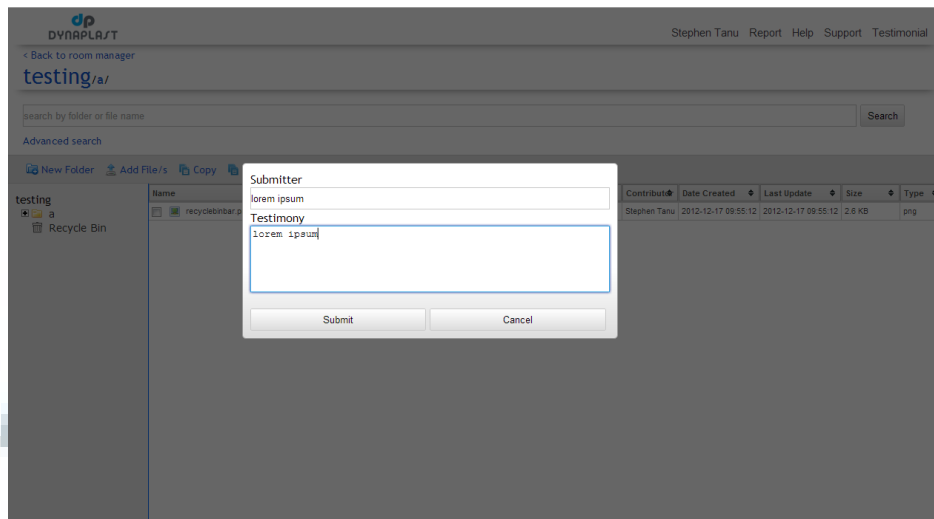
Gambar 3.66 Sub Menu *Testimonial*

Sub menu *Testimonial* berfungsi untuk melakukan operasi yang berkaitan dengan testimoni. Berikut adalah *form* yang tampil jika sub menu *send testimonial* dan *add testimony* dipilih.



Gambar 3.67 *Send Testimonial Form*

*Form send testimonial* berfungsi untuk mengirimkan testimonium pengguna kepada administrator dalam pengalaman penggunaan aplikasi.



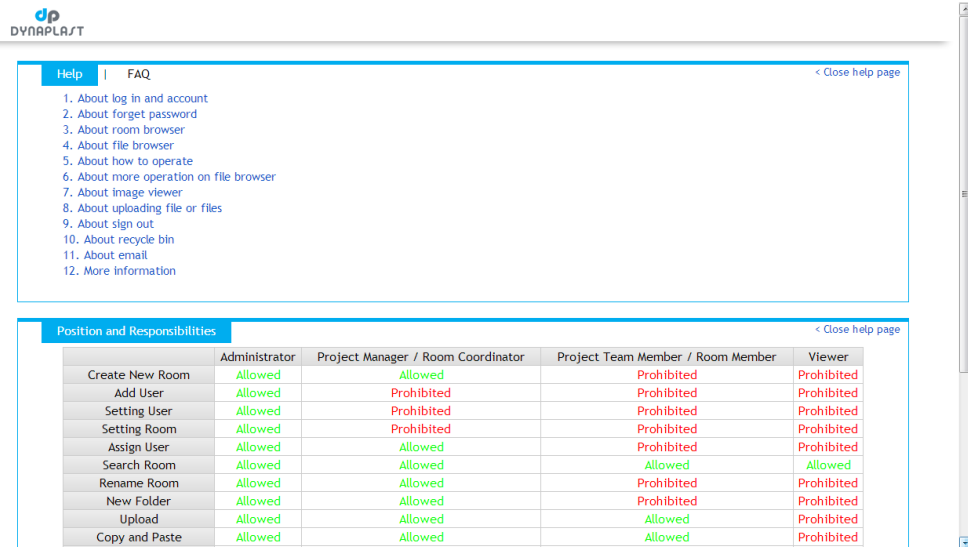
Gambar 3.68 Add Testimony Form

*Form add testimony* berguna bagi administrator untuk menambahkan testimonium. Sub menu *testimonial page* berfungsi untuk mengantarkan pengguna menuju halaman *Testimonial*.

#### 7. Halaman Bantuan

Halaman bantuan berfungsi untuk memberikan informasi pada pengguna mengenai cara penggunaan aplikasi serta hak akses yang dimiliki.

UMMN

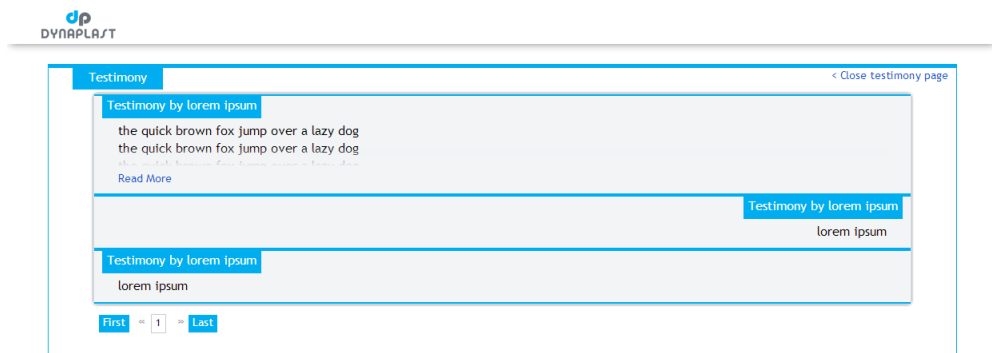


Gambar 3.69 Halaman Bantuan

Secara umum halaman bantuan dibagi menjadi tiga bagian utama, yaitu bagian *help* yang berisi bantuan yang bersifat tetap namun informasi yang diberikan berupa tulisan dan gambar, bagian FAQ yang berisi tulisan yang dapat ditambahkan oleh administrator, dan bagian posisi dan tanggung jawab yang bersifat tetap untuk memberikan informasi apa yang bisa dilakukan oleh pengguna dalam penggunaan aplikasi.

#### 8. Halaman *Testimonial*

Halaman *testimonial* adalah halaman yang berisikan testimonium yang telah dimasukkan oleh administrator.



Gambar 3.70 Halaman *Testimonial*

### 3.4.6 Sarana yang Digunakan

Dalam melakukan pengembangan aplikasi ini peserta magang tentu tidak terlepas dari sarana-sarana pendukung. Berikut adalah sarana-sarana yang menunjang pengembangan aplikasi Dynaplast e-Room.

1. HTML merupakan singkatan dari *hyper text markup language*. HTML berfungsi untuk memberikan aturan dalam pertukaran teks, grafik, suara, video, dan elemen multimedia lainnya (Gary B. Shelly, Denise M. Woods, William J. Dorin, 2009).
2. CSS merupakan singkatan dari *cascading style sheets* berguna untuk membantu presentasi dokumen yang ada (Meyer, 2006). Dengan digunakan CSS, maka tampilan dari aplikasi yang berbasis *web* ini menjadi lebih atraktif.
3. PHP merupakan *server-side scripting* yang menjadi bagian penting dalam aplikasi Dynaplast e-Room. PHP digunakan dalam proses *create*, *read*, *update*, dan *delete* dalam tabel *database*, pembacaan *file* yang ada pada *server*, dan untuk pengiriman notifikasi surat elektronik kepada pengguna.
4. JavaScript merupakan *scripting language*, yang pada aplikasi berfungsi untuk menghasilkan *web* yang dinamis dan juga validasi informasi.
5. *Library* jQuery merupakan *library* JavaScript yang berguna pada aplikasi untuk menghasilkan animasi serta memudahkan penggunaan AJAX.



6. IDE Adobe Dreamweaver CS 5 yang memudahkan peserta magang dalam melakukan pengembangan aplikasi karena telah tersedianya fasilitas pendukung.
7. *File* .htaccess untuk memberikan aturan pengaksesan pada *file* aplikasi yang telah dibuat.
8. *Server* yang menggunakan SUSE Linux Enterprise Server 11 sebagai sistem operasinya.
9. XAMPP yang digunakan sebagai *server* penguji pada tahap awal aplikasi dibangun, serta LAMPP yang digunakan sebagai *server* intranet perusahaan pada *server* berbasis Linux yang sudah disediakan.
10. WinSCP yang digunakan untuk melakukan pertukaran data dalam jaringan.

### 3.5 Kendala yang Ditemukan

Berikut adalah kendala yang ditemukan oleh peserta magang selama pengembangan aplikasi berlangsung dari awal hingga selesai.

1. Revisi *user requirement* baik dalam sisi tampilan yang disebabkan oleh perubahan logo dan gambaran perusahaan maupun perubahan serta penambahan fungsi aplikasi yang tidak jarang melibatkan *database*. Hal ini menyebabkan pada tiap revisi diperlukan *reset*, serta perubahan dalam sisi *database* yang ada pada *server*.
2. Perubahan ekosistem dari pengembangan pada sistem operasi Windows dan pada sistem operasi berbasis Linux dalam menyatakan alamat direktori.
3. *Server* yang disediakan datang terlambat, sehingga dapat menyebabkan terjadi ruang pada linimasa pengembangan aplikasi.
4. Terjadinya kekurangan daya untuk menghidupkan *server* yang baru saja tiba, sehingga dapat mengakibatkan terjadinya ruang pada linimasa pengembangan aplikasi.

### 3.6 Solusi atas Kendala yang Ditemukan

Berikut adalah solusi atas kendala yang ditemukan peserta magang untuk mengurangi efek yang ditimbulkan oleh kendala yang terjadi dalam pengembangan aplikasi.

1. Membuat *backup* serta penomoran pada setiap tingkatan penyelesaian aplikasi yang berguna dalam menentukan fungsi yang sudah berjalan dengan baik, yang perlu diubah maupun yang perlu dihapus. Selain itu setiap *backup* dapat digunakan sebagai referensi jika terjadi kesalahan pada fungsi yang harus diubah.
2. Melakukan peninjauan serta pengetestan aplikasi dalam sistem operasi berbasis Linux. Jika ditemukan kesalahan, maka kesalahan tersebut diperbaiki dan kemudian dicatat untuk menjadi referensi peninjauan selanjutnya.
3. Mencari serta mencatat informasi mengenai pengaturan aplikasi pada *server* berbasis Linux sehingga ketika *server* tiba, peserta magang dapat langsung mempraktikkan informasi yang sudah dikumpulkan.
4. Memilih dan melanjutkan pengembangan aplikasi yang tidak memiliki dampak yang besar pada *server* XAMPP.

UMMN