# ANALISIS DAN DESAIN SISTEM

Pada bab ini dibahas mengenai desain sistem dan teknologi apa saja yang digunakan dalam sistem, proses kerja sistem, analisa yang dilakukan, permasalahan yang terjadi dan solusi yang digunakan.  Proses analisis dan desain dilakukan dengan tujuan untuk memberikan gambaran besar bagaimana sistem bekerja dan saling berkomunikasi sehingga implementasi pengkodean menjadi lebih mudah dilakukan.

## Desain Aplikasi

*Bussiness Intelligence* bertujuan untuk mengubah data mentah menjadi informasi yang dapat beguna untuk analisis bisnis atau pengambilan keputusan. Aplikasi yang menerima data mentah dan mengolah data tersebut sehingga menjadi informasi diharapkan dapat membantu pengguna dalam pengambilan keputusan. Selain itu fitur untuk melakukan rekomendasi metode prediksi diharapkan dapat membantu pengguna awam yang ingin melakukan analisis data.

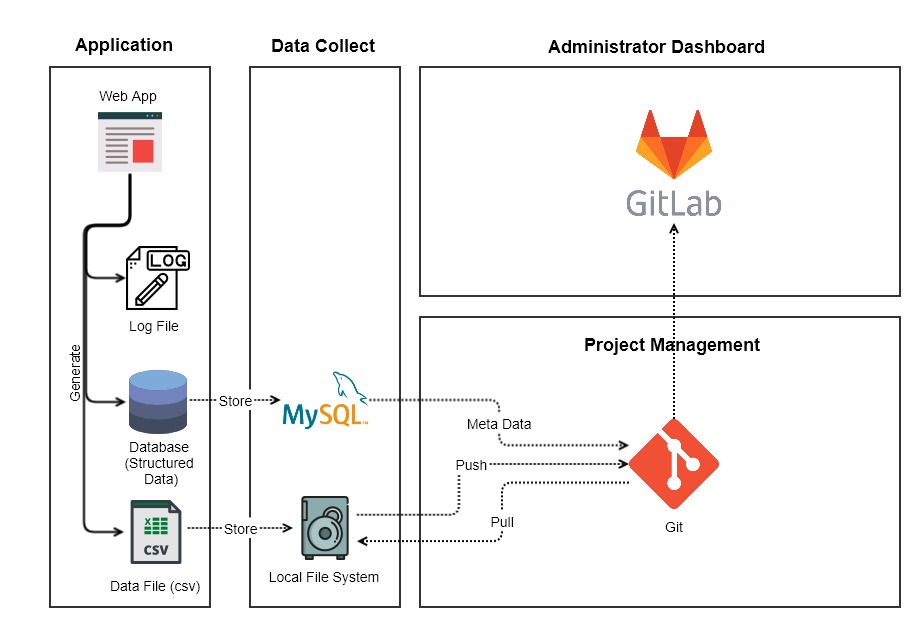
Aplikasi yang dibuat memiliki banyak fitur untuk melakukan analisa data, fitur – fitur utama adalah:

1. Registrasi dan pengaturan User
2. Menerima Data (Upload) dalam format CSV
3. Membuat dan mengatur projek untuk pengolahan data.
4. Melakukan rekomendasi metode prediksi.

Dengan adanya fitur – fitur utama ini maka diharapkan pengguna awam maupun berpengalaman dapat terbantu dalam melakukan analisa data. Selain itu program juga merupakan aplikasi berbasis web yang dapat diakses di manapun sehingga memudahkan pengguna.

## Desain Sistem dan Teknologi

Sistem informasi dalam mendukung pembuatan *project* untuk melakukan analisa memiliki banyak fitur untuk membantu proses analisa. Ketika melakukan analisa maka sebuah projek harus dibuat untuk mengatur data yang digunakan untuk analisa, *log* atau catatan apa saja yang terjadi dalam project, dan juga file – file yang dihasilkan projek seperti *image chart*. Selain itu projek juga diharapkan dapat di-*share* ke user lain tetapi dalam 1 perusahaan yang sama. Git merupakan teknologi yang digunakan untuk memenuhi fitur tersebut. Produk yang digunakan adalah Gitlab.



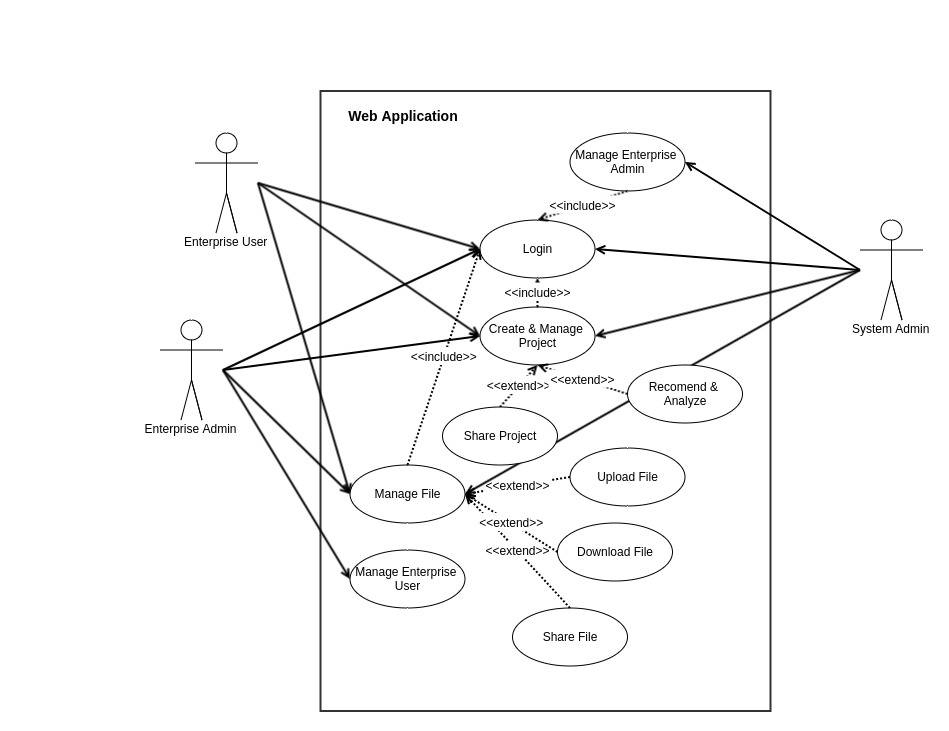
1. Desain Sistem dan Teknologi

Teknologi Git digunakan karena memiliki konsep pengaturan projek yang baik dan teratur. Hal ini terlihat dari proses *push, pull, dan commit* selain itu Git juka memiliki fitur *Branch* yang baik digunakan untuk melihat proses atau perubahan apa yang dilakukan tim kepada projek yang dibuat.

Gitlab merupakan aplikasis berbasis web yang digunakan untuk mendukung sistem pengaturan projek. Gitlab dipilih karena memiliki REST API yang lengkap beserta dokumentasinya. Dengan adanya bantuan API dari Gitlab maka proses sinkronisasi dengan aplikasi menjadi lebih mudah.

### Use Case Diagram

*Use case diagram* merupakan alat bantu untuk menggambarkan tingkah laku pengguna terhadap sistem. Pada aplikasi ini terdapat 3 fitur utama yaitu mengatur file seperti *upload, download, delete, share file*. Mengatur *user* atau pengguna yang dapat masuk kedalam sisten dan mengatur projek. Ketiga fitur utama ini masih dapat dijabarkan / dipecah menjadi fitur – fitur yang lebih mendasar.



1. Desain Use Case Diagram

Penjabaran dari fitur aplikasi ini:

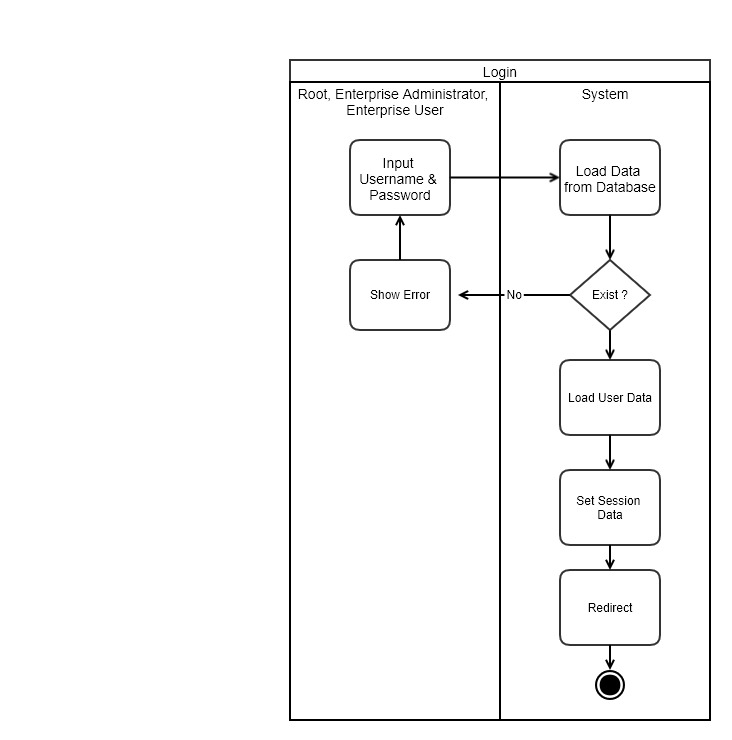
1. Login
2. Manage Enterprise User
3. Manage Enterprise Admin
4. Manage File
   1. Upload File
   2. Download File
   3. Share File
5. Create & Manage Project
   1. Share Project
   2. Recomend & Analyze

### Activity Diagram

*Activity diagram* digunakan untuk mengetahui secara detail aktivitas yang terjadi antara user dengan sistem. *Activity diagram* akan dibagi menjadi beberapa *activity* yang menjelaskan bagaimana dan apa yang terjadi terhadap sistem jika pengguna melakukan sesuatu ke sistem seperti login.

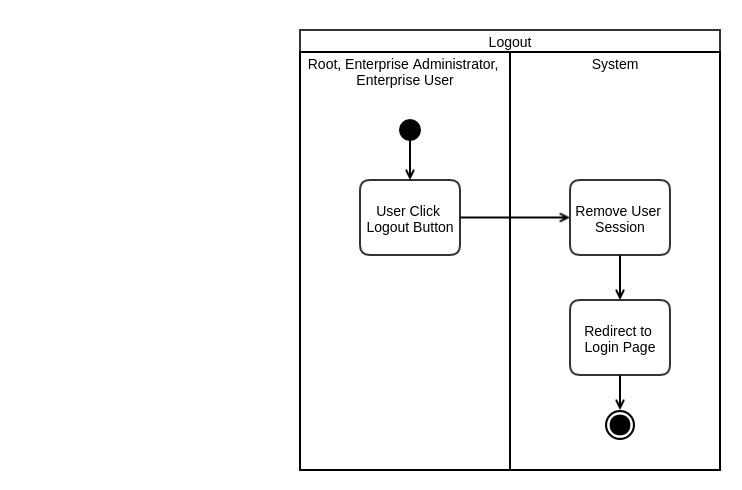
1. Activity Login

*Activity* ini dilakukan ketika pengguna ingin masuk kedalam sistem dan menggiunakan fitur – fitur yang ada di dalamnya. Pertama pengguna akan memasukan *username* dan *password* , kemudian sistem akan melakukan autentikasi terhadap input yang dilakukan pengguna. Jika input yang diberikan memiliki format yang benar dan pengguna terdaftar pada sistem, maka user akan dialihkan menuju halaman *home.* Selain itu pengguna akan mendapatkan pesan *error.*



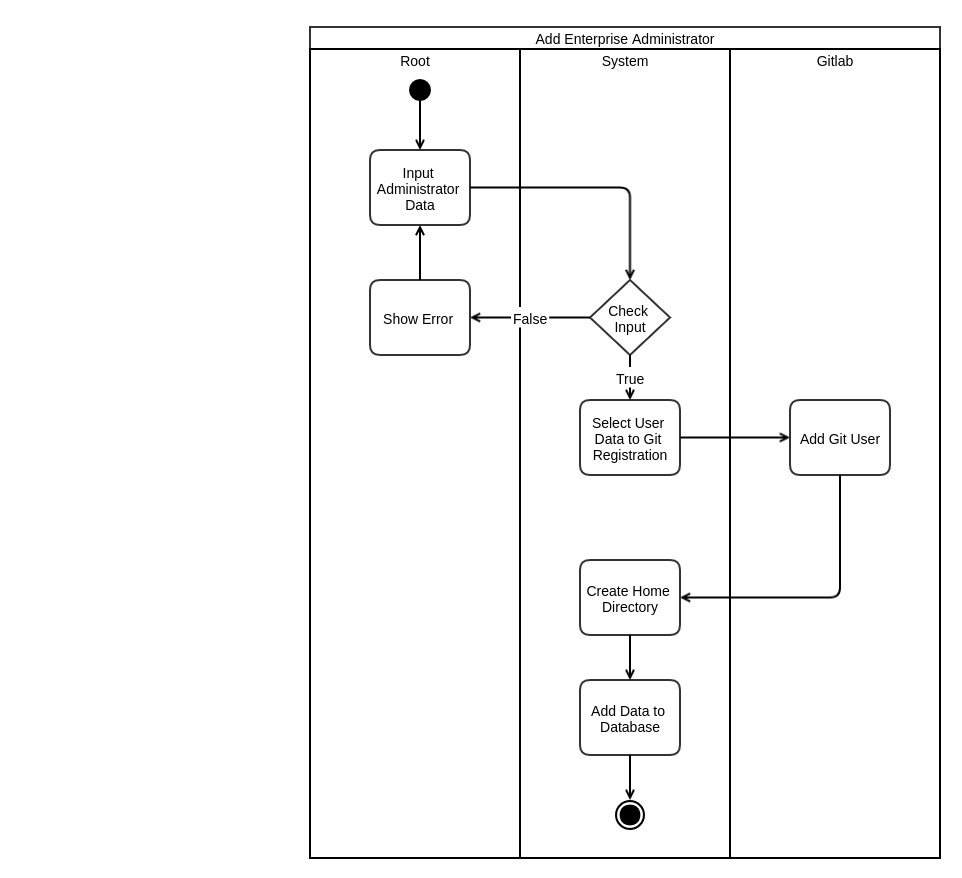
1. Activity Login
2. Activity Logout

*Activity* ini dilakukan ketika pengguna ingin keluar dari sistem. Sistem akan menghapus data sesi / *session* dari pengguna tersebut.



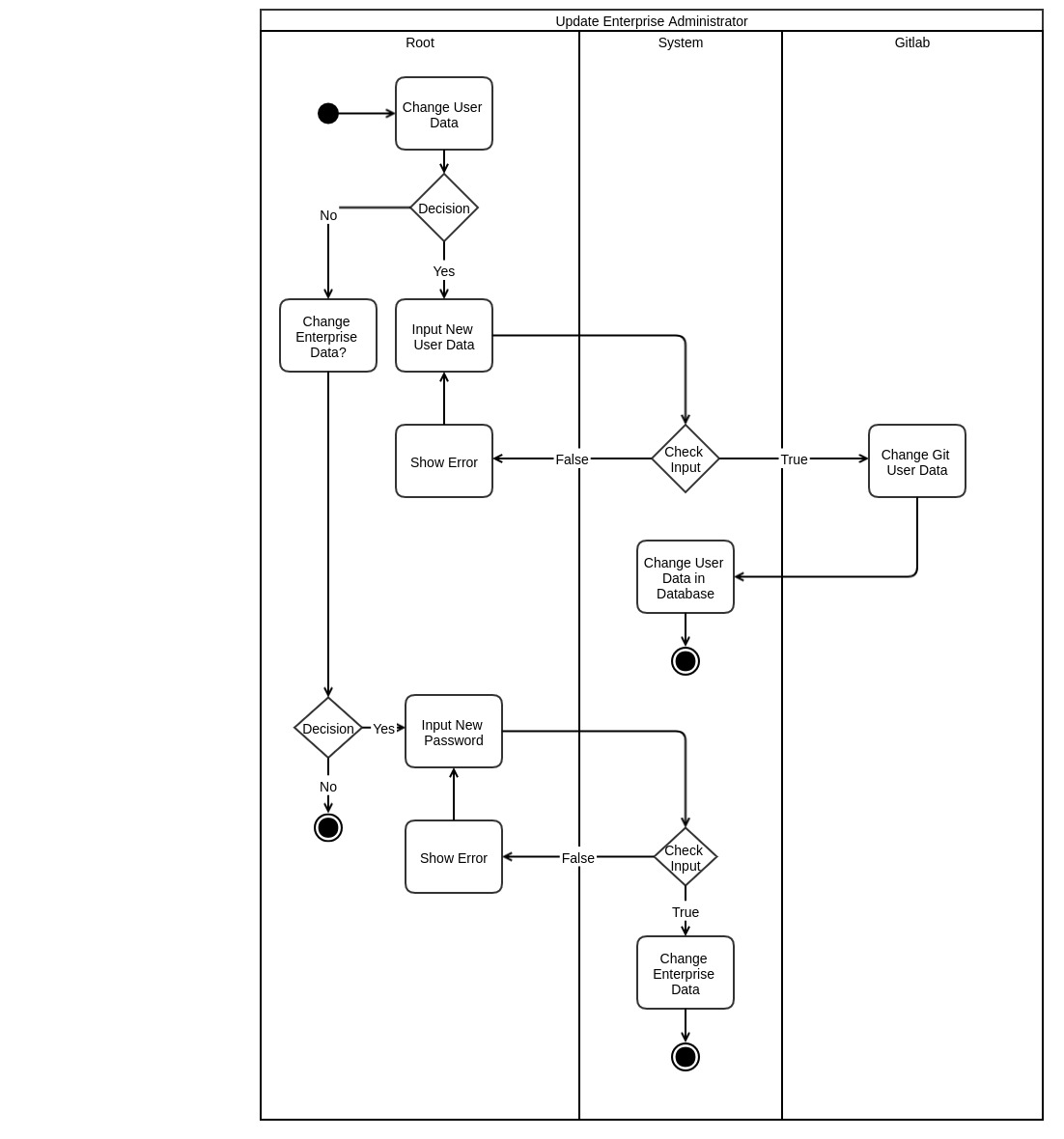
1. Activity Logout
2. Activity Add Enterprise Administrator

*Activity* ini dilakukan ketika pengguna, khususnya *System Administrator* ingin menambahkan *Enterprise Administrator*. *Enterprise Administrator* adalah pengguna / user yang memiliki otoritas untuk menambahkan pengguna / user pada perusahaanya. Ketika user tipe ini dibuat maka sistem juga akan otomatis menambahkan *home directory* sebagai tempat untuk menyimpan file – file yang dimiliki user dan *repository* sebagai tempat untuk mengatur projek.



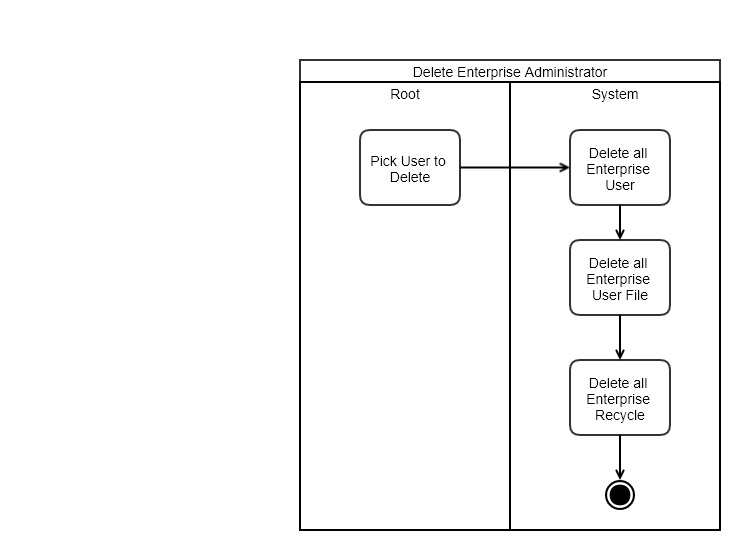
1. Activity Add Enterprise Administrator
2. Activity Edit Enterprise Administrator

*Activity* ini terjadi ketika pengguna khususnya *System Administrator* ingin mengubah data seorang pengguna atau user dengan tingkat otoritas *Enterprise Administrator*. Ketika melakukan pengubahan data, sistem juga akan melakukan *check* dan sinkronisasi dengan data yang ada pada Gitlab.



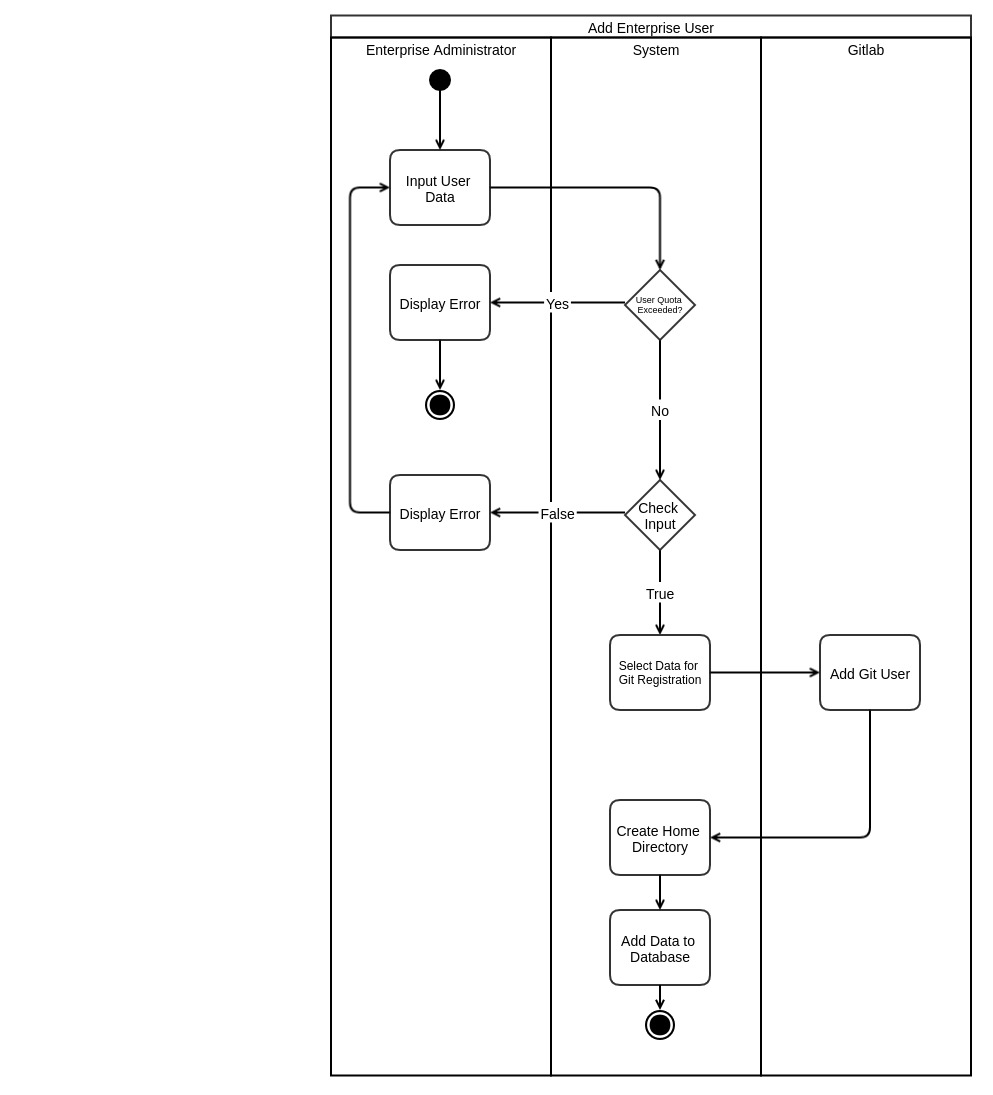
1. Activity Edit Enterprise Administrator
2. Activity Delete Enterprise Administrator

*Activity* ini terjadi ketika pengguna khususnya *System Administrator* ingin menghapus *Enterprise Administrator* / *Enterprise* itu sendiri. Ketika menghapus pengguna, maka semua data dan user milik perusahaan akan dihapus. Ketika proses penghapusan dilakukan sistem juga akan melakukan sinkronisasi dengan Gitlab. Sebelum proses penghapusan dijalankan pengguna / *System Administrator* akan diminta untuk melakukan konfirmasi penghapusan.



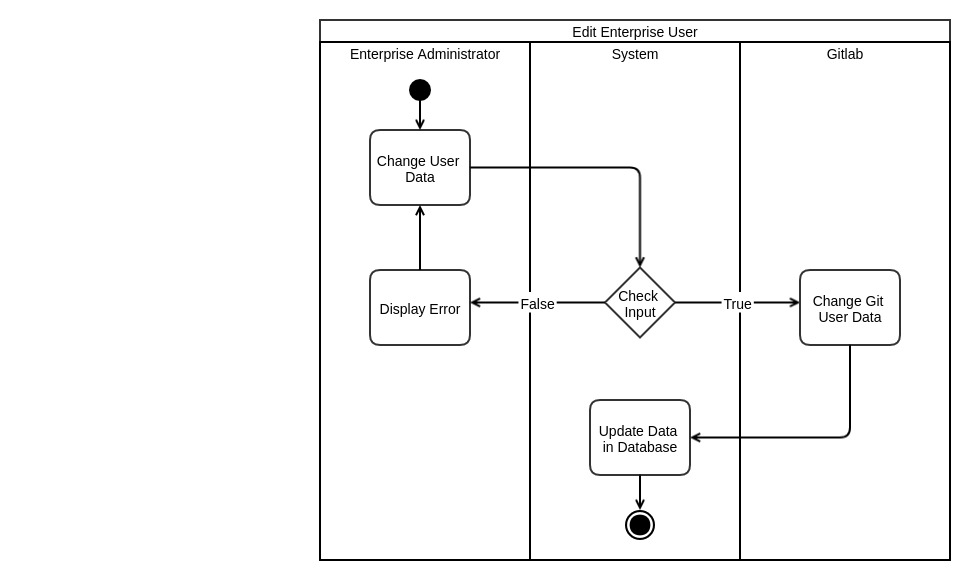
1. Activity Delete Enterprise Administrator
2. Activity Add Enterprise User

*Activity* ini terjadi ketika pengguna / user khususnya *Enterprise Administrator* menambahkan pengguna dalam perusahaanya. Ketika menambahkan pengguna pada perusahaan sistem akan melakukan pemeriksaan apakah perusahaan masih memiliki quota untuk menambahkan pengguna / user.



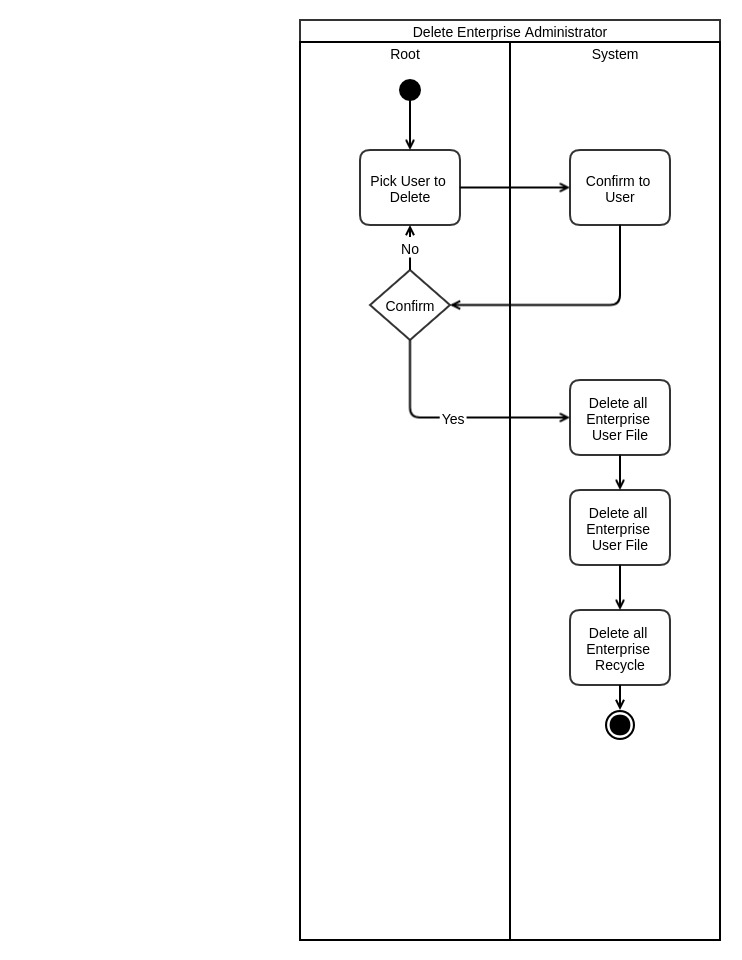
1. Activity Add Enterprise User
2. Activity Edit Enterprise User

*Activity* ini erjadi ketika pengguna / user khususnya *Enterprise Administrator* mengubah data pengguna dalam perusahaanya. Ketika pengubahan data dilakukan maka sistem akan melakukan sinkronisasi dengan data pada Gitlab.



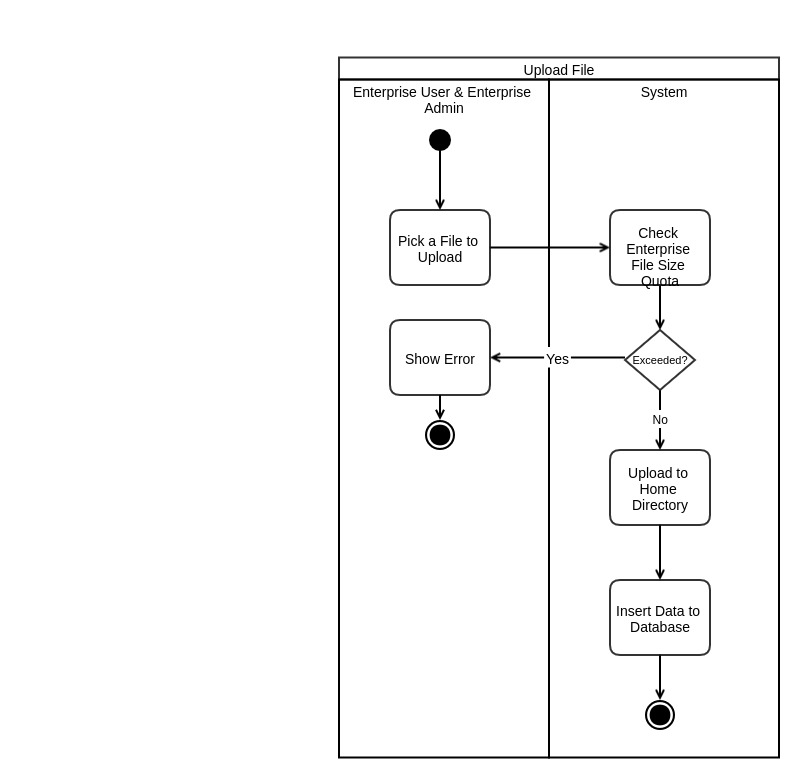
1. Activity Edit Enterprise User
2. Activity Delete Enterprise User

*Activity* ini erjadi ketika pengguna / user khususnya *Enterprise Administrator*  menghapus pengguna dalam perusahaanya. Ketika pengguna dihapus dari sistem, sistem akan menghapus juga *repository* pengguna pada Gitlab.



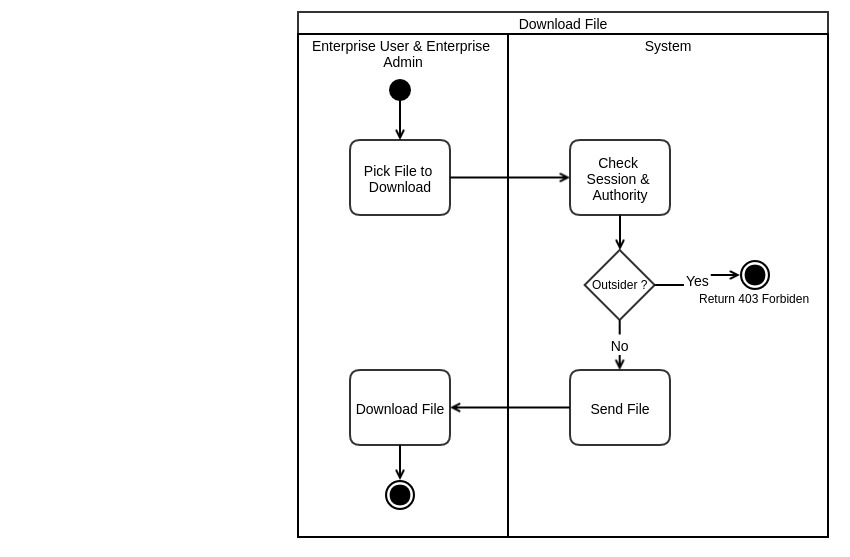
1. Activity Delete Enterprise User
2. Activity Upload File

*Activity* ini terjadi ketika pengguna / user ingin memasukan atau mengirimkan file menuju *home directory* pengguna pada sistem. File yang terdapat pada *home directory* dapat digunakan untuk membuat projek agar dapat dilakukan analisa lebih lanjut.



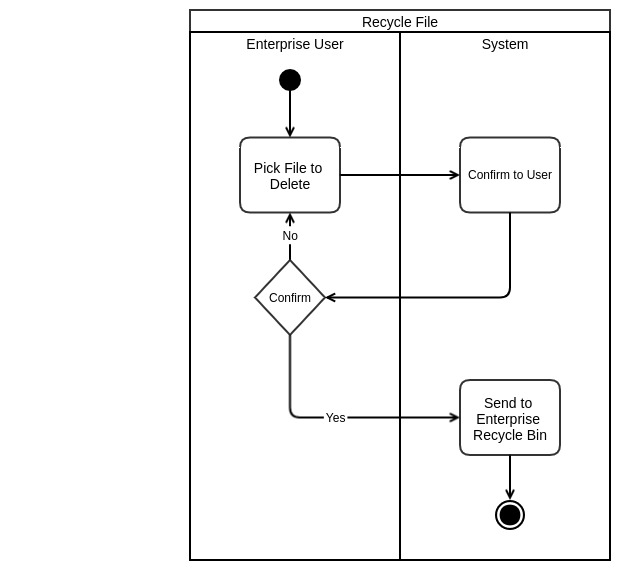
1. Activity Upload File
2. Activity Download File

*Activity* ini terjadi ketika pengguna / user ingin mengunduh atau me-*download* file dari *home directory* pengguna pada sistem. Ketika melakukan download user harus masuk kedalam sistem dan memiliki sesi. Jika pengguna / user tak memiliki sesi maka sistem akan menganggap permintaan / *request* dilakukan oleh orang di luar sistem.



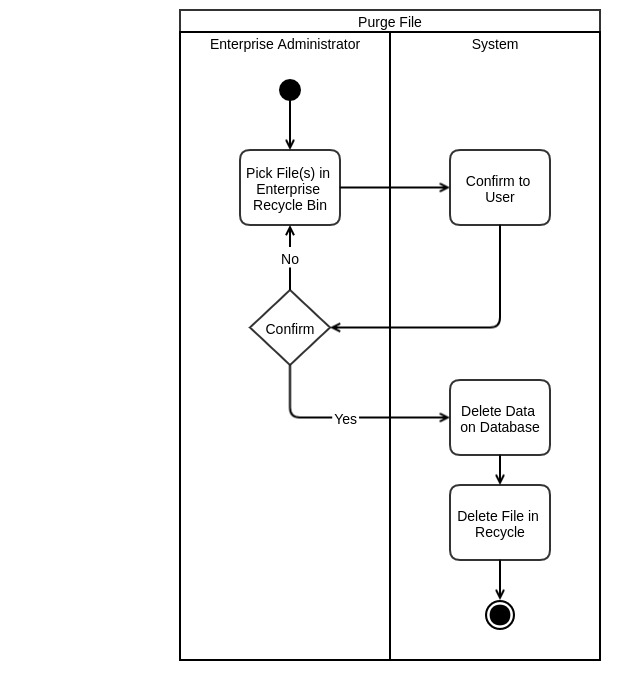
1. Activity Download File
2. Activity Recycle File

*Activity* ini terjadi ketika pengguna khususnya *Enteprise User* ingin menghapus file pada *home directory.* Ketika file dihapus maka akan masuk kedalam *Enterprise Recycle Bin.* Kondisi ini membuat ketika file tidak jadi dihapus, *Enterprise Administrator* dapat melakukan *restore*  untuk mengembalikan file menuju *home directory* pemilik file tersebut.



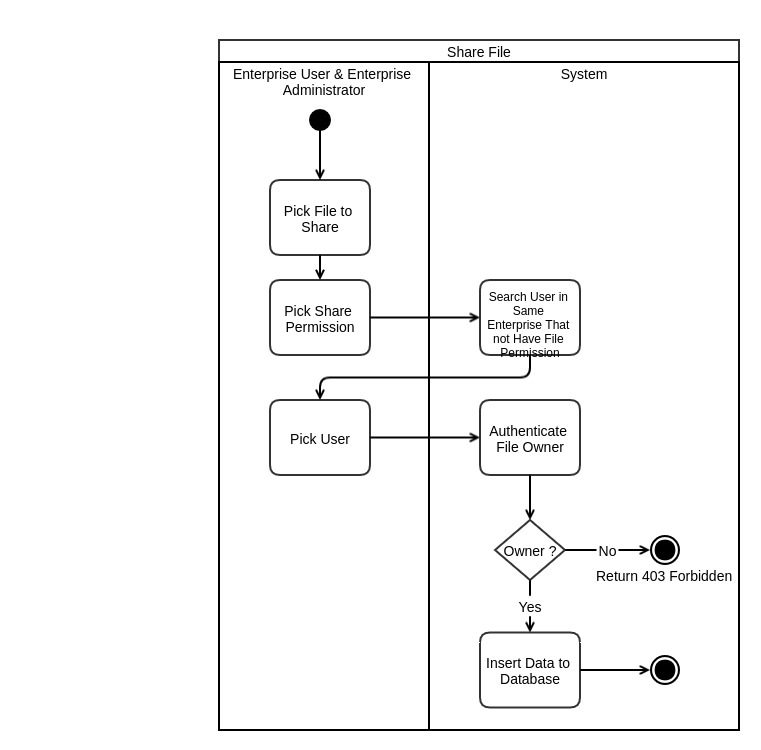
1. Activity Recycle File
2. Activity Purge File

*Activity* ini terjadi ketika pengguna khususnya *Enteprise Administrator* ingin menghapus file pada *Enterprise Recycle Bin.* Ketika file dihapus dari *Enterprise Recycle Bin* maka file tersebut akan benar – benar terhapus dan tidak dapat dikembalikan seperti semula.



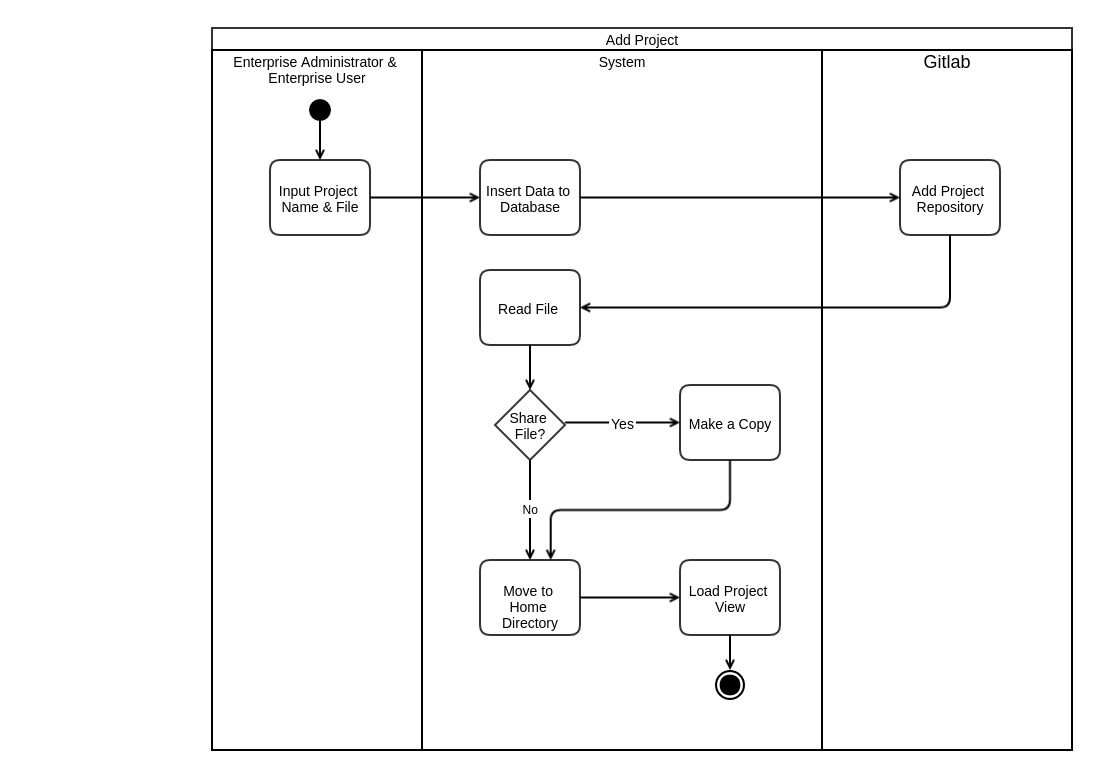
1. Activity Purge File
2. Activity Share File

*Activity* ini terjadi ketika pengguna / user ingin membagi akses file yang dimiliki terhadap pengguna lainnya dalam 1 perusahaan / *Enterprise* yang sama.



1. Activity Share File
2. Activity Create Project

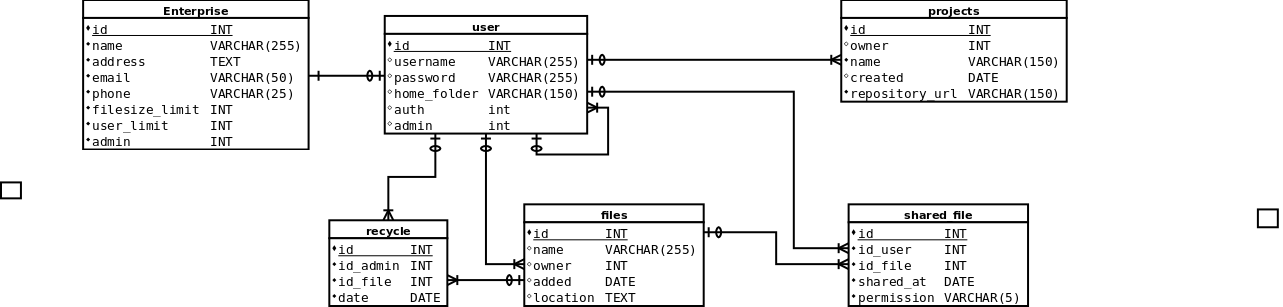
*Activity* ini terjadi ketika pengguna ingin membuat sebuah projek. Ketika pengguna membuat sebuah projek, maka sistem akan meminta Gitlab untuk membuatkan *repository* baru atas nama pengguna yang membuat projek baru.



1. Activity Create Project

## Desain *ERD (Entity Relationship Diagram)*

Tahap perancangan ERD dilakukan setelah tahap penentuan fitur dan aktivitas yang terjadi pada sistem dilakukan. ERD dirancang untuk memberikan gambaran mengenai desain database yang digunakan. ERD juga berguna sebagai dokumen yang membantu saat proses pengkodean.



1. Entity Relationship Diagram

Database yang telah didesain di atas memiliki tabel sebagai berikut:

Tabel 3.1 Tabel User

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id | INT | 11 | Id user |
| username | VARCHAR | 255 | Username pengguna |
| password | VARCHAR | 255 | Password pengguna |
| home\_folder | VARCHAR | 150 | Lokasi folder user pada local system |
| auth | INT | 11 | 1. System Admin  2. Enterprise Admin  3. Enterprise User |
| admin | INT | 11 | ID Administrator dari user, 0 jika user merupakan Enterprise Administrator / System Admin |

Tabel 3.2 Tabel Enterprise

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id | INT | 11 | ID Enterprise / Perusahaan |
| name | VARCHAR | 255 | Nama perusahaan |
| address | TEXT |  | Alamat perusahaan |
| email | VARCHAR | 50 | Email perusahaan |
| phone | VARCHAR | 25 | Telepon perusahaan |
| filesize\_limit | INT | 11 | Limit penggunaan l*ocal storage* dalam GB (Gigabyte) |
| user\_limit | INT | 11 | Limit berapa banyak user yang dapat dimiliki perusahaan |
| admin | INT | 11 | ID Enterprise Administrator. |

Tabel 3.3 Tabel Files

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id | INT | 11 | ID sebuah file |
| name | VARCHAR | 255 | Nama file |
| owner | INT | 11 | ID Pengguna yang memiliki file |
| added | DATE |  | Tanggal file memasuki sistem |
| location | TEXT |  | Lokasi file pada *local storage* |

Tabel 3.4 Tabel Shared File

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id | INT | 11 | Primary Key |
| id\_user | INT | 11 | ID User target user yang ingin dishare |
| id\_file | INT | 11 | ID File yang ingin dishare |
| shared\_at | DATE |  | Kapan *share* terjadi |
| permisssion | VARCHAR | 5 | r, untuk read  w, untuk read dan write |

Tabel 3.5 Tabel Recycle

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id | INT | 11 | Primary Key |
| id\_admin | INT | 11 | ID User Enterprise Administrator |
| id\_file | INT | 11 | ID File yang masuk kedalam Enterprise Recycle Bin |
| date | DATE |  | Kapan File masuk kedalam Enterprise Recycle Bin |

Tabel 3.6 Tabel Project

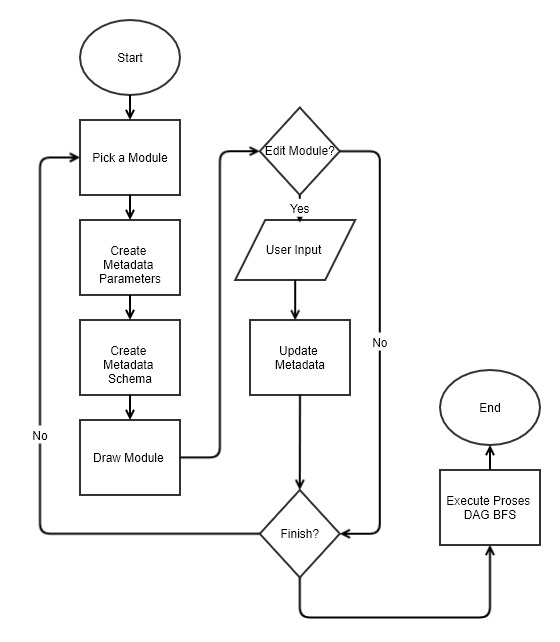
|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Ukuran** | **Keterangan** |
| id | INT | 11 | ID sebuah *project* |
| owner | INT | 11 | ID User pemilik *project* |
| name | VARCHAR | 150 | Nama *project* |
| created | DATE |  | Kapan *project* dibuat |
| repository\_url | VARCHAR | 150 | URL *Project Repository* pada Gitlab |

## Desain *Flowchart*

Perancangan *flowchart* dilakukan untuk memperjelas proses yang ada. Beberapa proses akan dijelakan dalam bentuk *flowchart* untuk memperjelas secara detail kegiatan yang terjadi di dalamnya. Desain *flowchart*  di bawah ini akan fokus menjelaskan proses yang terjadi ketika projek dibuat dan pengaturannya.

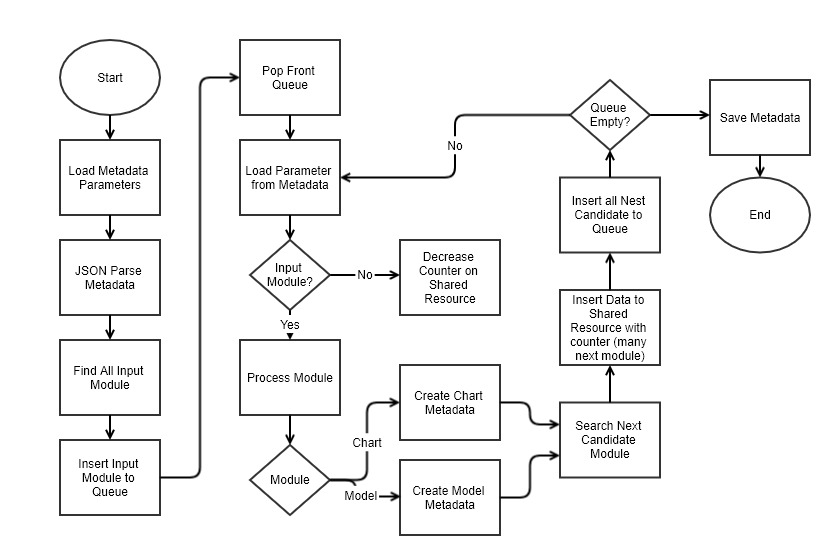
1. *Flowchart* *Directed Acyclic Graph (Interface)*

*Flowchart* ini menjelaskan bagaimana tampilan *data flow* terbuat, ketika *user* ingin me-*design* *data flow* untuk proses *pre-processing*.



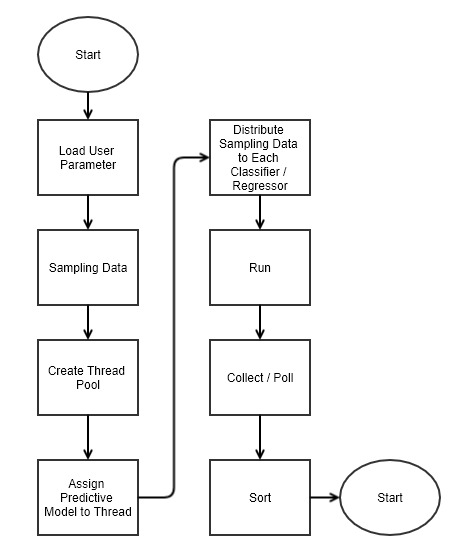
1. Flowchart DAG Interface
2. *Flowchart Breadth First Search Directed Acyclic Graph*

*Flowchart* ini berfungsi untuk menjelaskan proses mengartikan metadata dari proses desain *data flow* menjadi urutan process yang harus dijalankan oleh aplikasi.



1. Proses BFS pada DAG
2. *Flowchart Recomend & Analyze*

Pengguna ketika ingin mendapatkan rekomendasi metode prediksi maka pengguna dapat melakukan *setting* parameter terlebih dahulu. Ketika pengguna selesai melakukan setting sesuai keinginannya maka pengguna dapat meminta sistem untuk merekomendasikan metode.



1. Proses Sistem Rekomendasi

Pertama sistem akan melakukan sampling data, dimana data tersebut akan dianalisa. Sistem kemudian akan melakukan analisa terhadap beberapa metode lalu melakukan *ranking* metode berdasarkan kecepatan atau *accuracy* metode setelah dilakukan validasi.

Metode yang direkomendasi berdasarkan:

1. *Classification*
   1. *Decision Tree*
   2. *Naive Bayes*
   3. *Logistic Regression*
2. *Regression*
   1. *Lasso Regression*
   2. *Regression Tree*
   3. *Support Vector Machine*

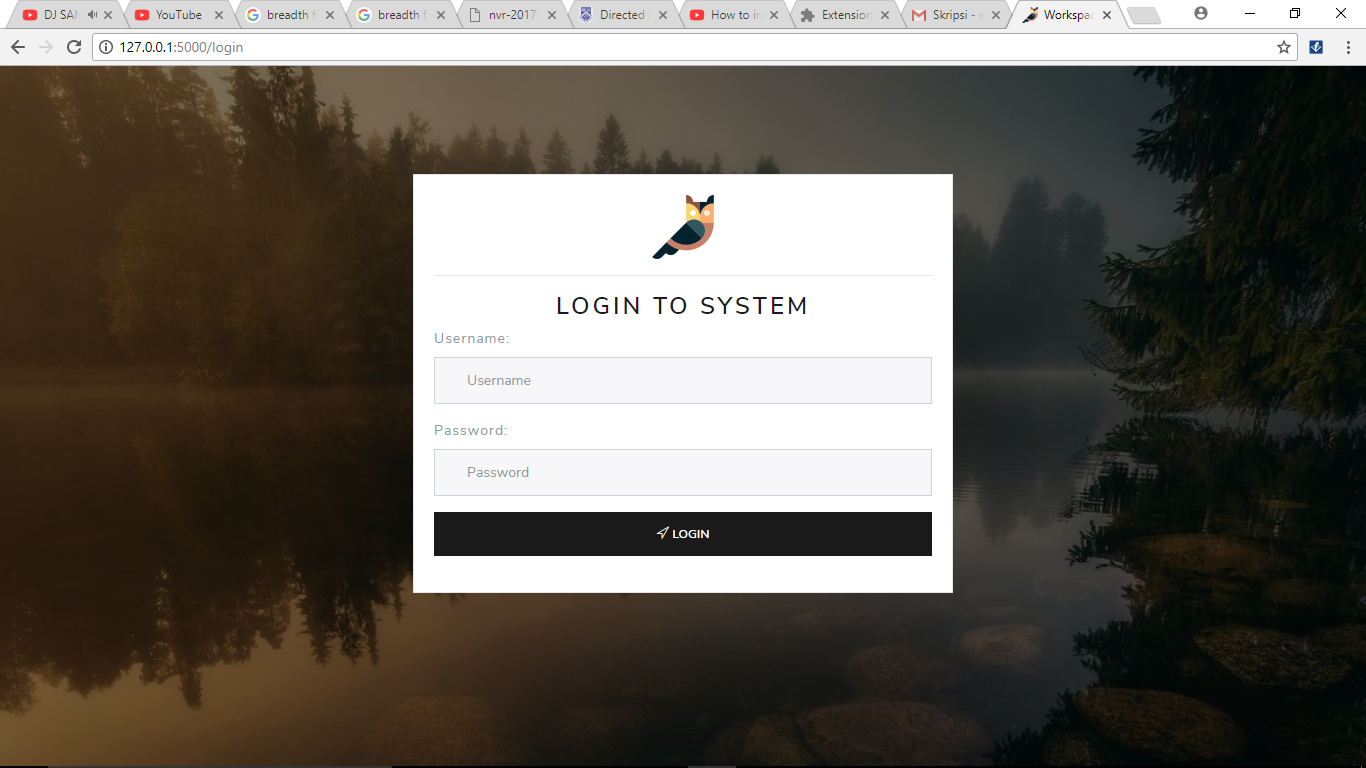
Metode untuk melakukan validasi adalah *cross-validation* dimana menggunakan 10 fold. Setelah proses validasi maka 10 *accuracy* yang didapat akan di rata – rata dan dicari *standard deviation* – nya. Tujuannya adalah untuk mendapatkan accuracy setelah validasi dan konsistensi model dalam memprediksi (*Robust*).

## Desain Tampilan Aplikasi

Pada bagian ini akan dijelaskan desain tampilan aplikasi yang terdiri atas tampilan halaman *login*, *layout* tampilan utama aplikasi, tampilan untuk membuat dan mengatur *project*, tampilan *manage user*, tampilan halaman pengaturan file pada *local storage*.

### Halaman *Login*

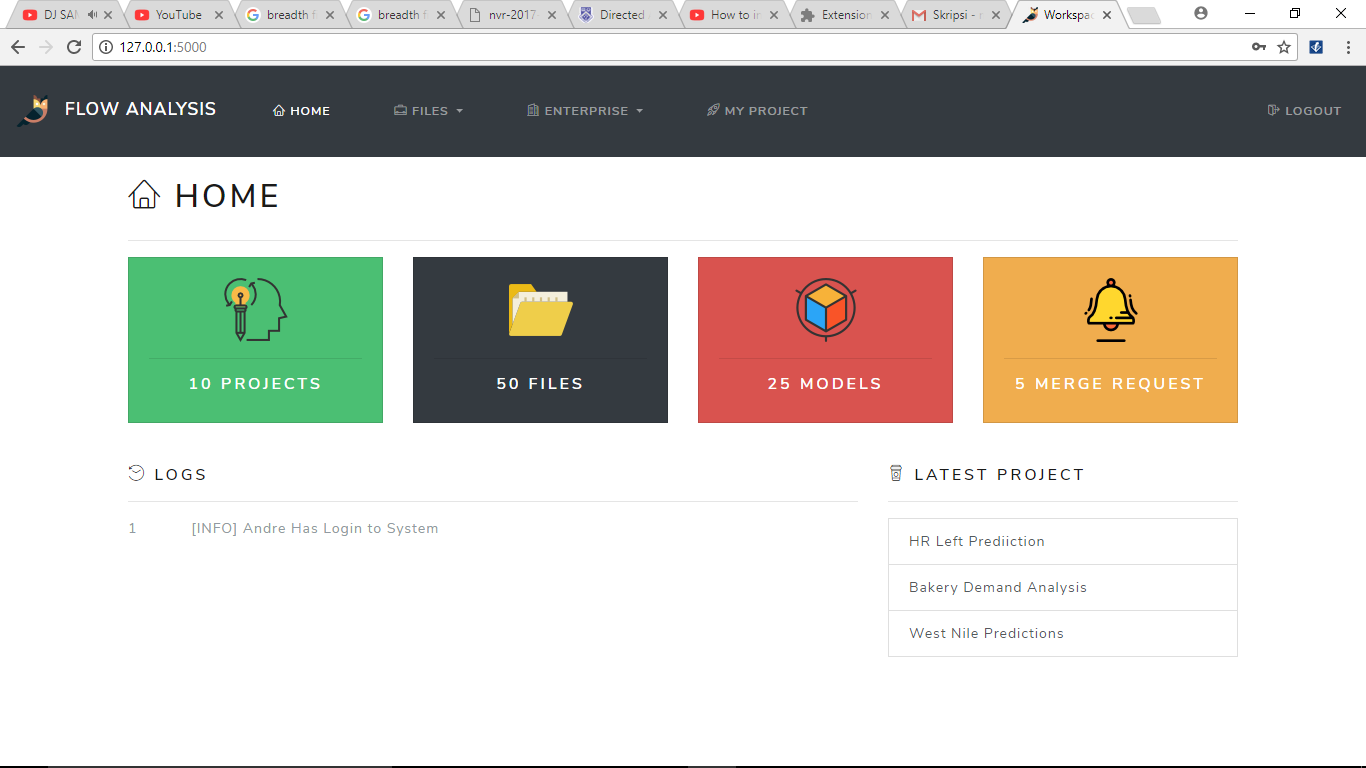
Halaman *Login* merupakan halaman pertama yang akan dikunjungi oleh pengguna. Pada halaman ini pengguna akan memasukkan 2 input yaitu *username* dan *password.*



1. Halaman *Login*

### Halaman *Dashboard*

Halaman *dashboard* merupakan halaman pertama yang akan dikunjungi pengguna ketika memasuki aplikasi atau setelah melakukan autentikasi pada halaman *login*. Halaman ini memberikan informasi singkat mengenai notifikasi dan beberapa *project* terakhir yang telah dikerjakan.



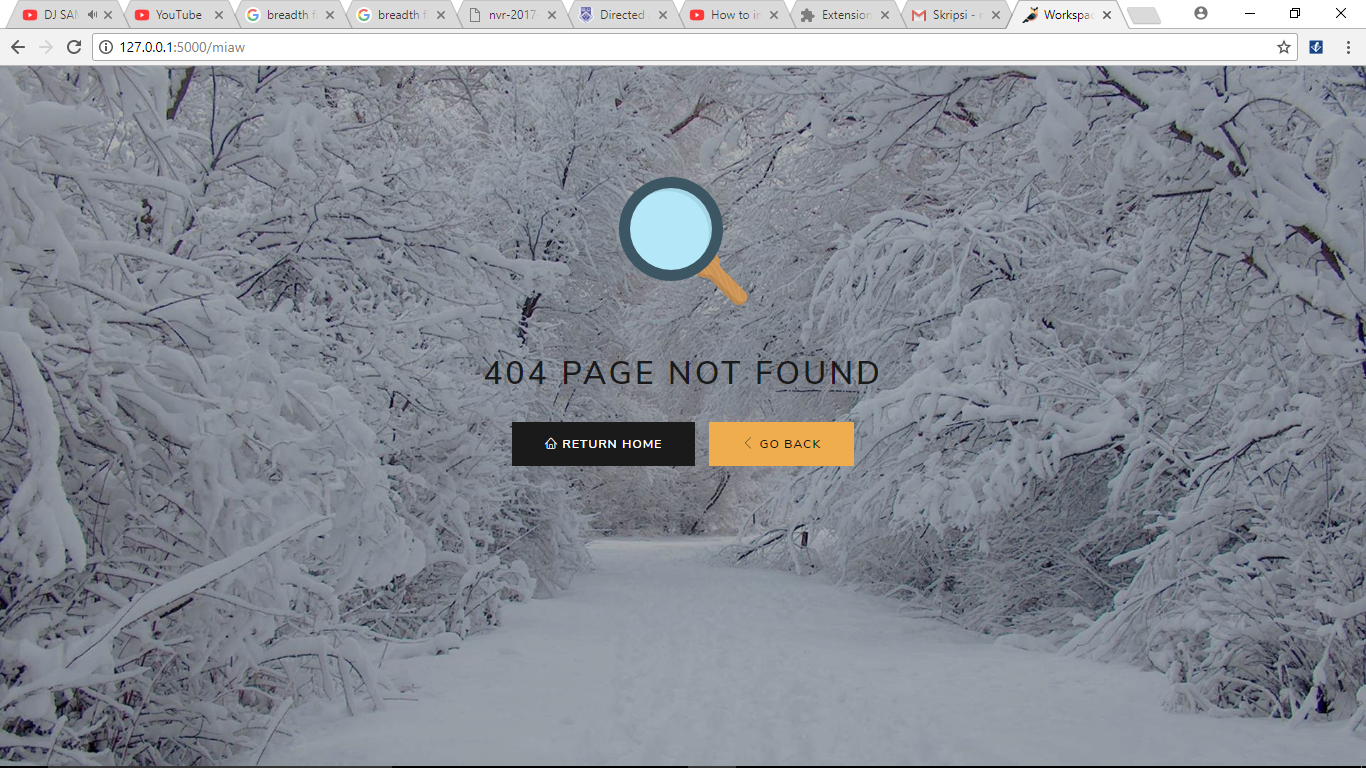
1. Halaman Dashboard

### Halaman *Error*

Halaman *error* adalah halaman kunstomisasi yang ditampilkan ketika pengguna mengalami *error* seperti 404 (*Not Found*). Halaman *error*  pada aplikasi ini terdiri atas:

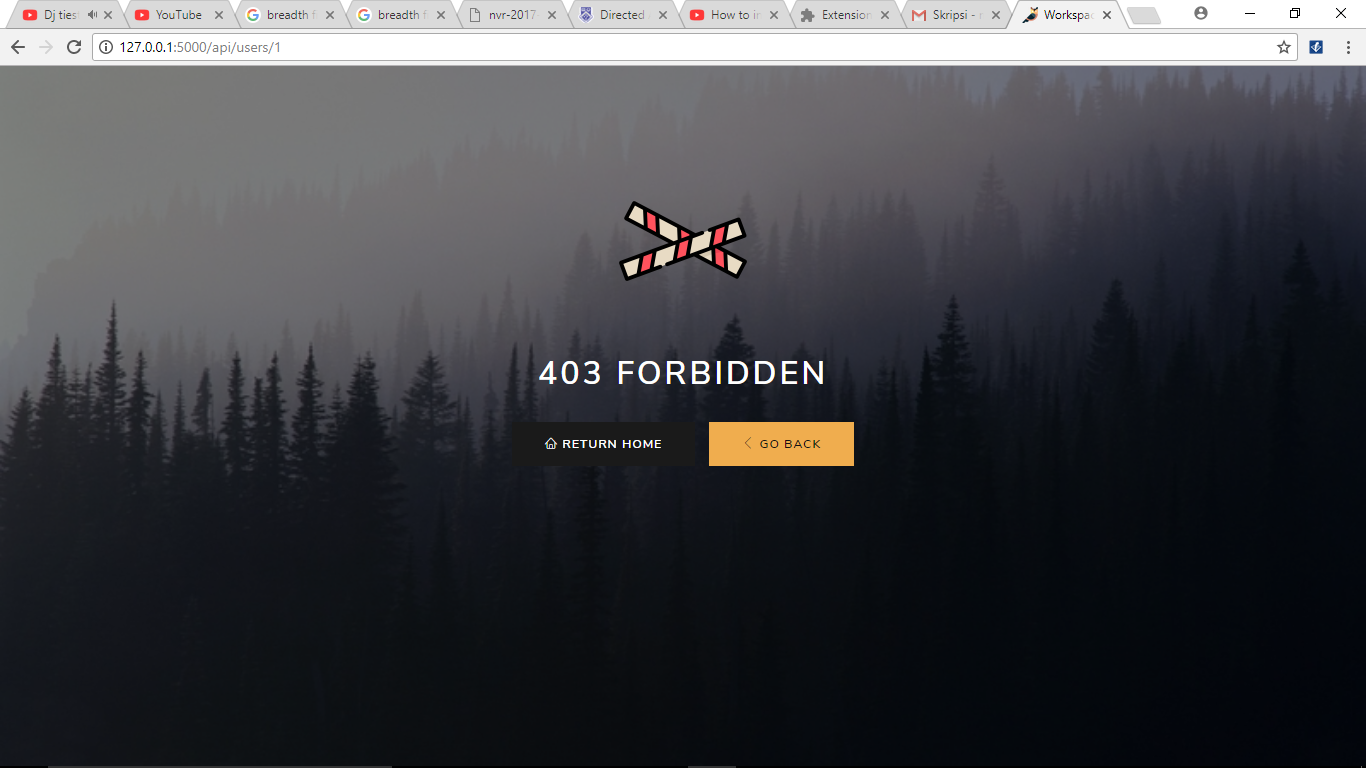
1. Halaman *Error 404 (Not Found)*

Halaman ini akan ditampilkan jika server merespon *request* pengguna dengan *status code* 404. Arti dari *error*  ini adalah pengguna mencoba untuk melakukan akses ke *URL* yang tidak terdapat pada sistem. Pada halaman ini pengguna dapat kembali ke 1 halaman sebelum halaman *error* atau dapat kembali ke halaman *home.*



1. Halaman Error 404 (Not Found)
2. Halaman *Error 403 (Forbidden)*

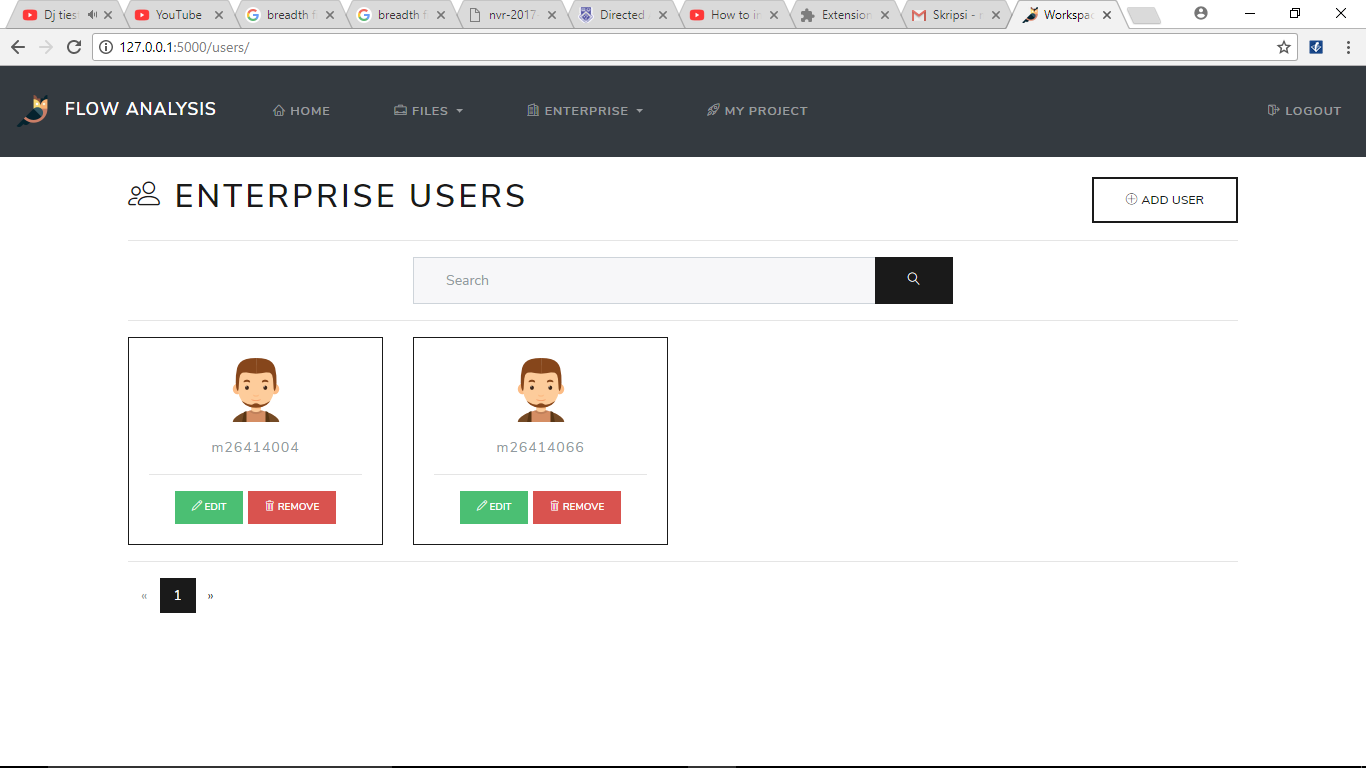
Halaman ini akan ditampilkan jika server merespon *request* pengguna dengan *status code* 403. Arti dari *error* ini adalah pengguna mencoba untuk melakukan akses kepada materi atau *resource*  pada sistem yang bukan miliki pengguna tersebut, Pengguna mencoba melakukan akses ke halaman yang tidak ditujukan kepada level *authority*  pengguna, atau pengguna mencoba untuk melakukan akses terhadap data yang tidak seharusnya bisa di dapatkan pengguna. *Error* ini juga dapat terjadi ketika ada sebuah akses yang meminta *resource /* data pada server namun akses tersebut tidak memiliki sesi pada sistem. Pada halaman ini pengguna dapat kembali ke 1 halaman sebelum halaman *error* atau dapat kembali ke halaman *home.*



1. Halaman Error 403 (Forbidden)

### Halaman *Manage User* (*System Admin* & *Enterprise Administrator*)

Halaman *manage user*  terdiri dari 2 macam halaman yang berbeda berdasarkan level autentikasi yang melakukan akses kepada halaman tersebut. Halaman ini bertujuan untuk melakukan *create, read, update, delete* terhadap *master* data *user*. Pengguna dengan level autentikasi *System Admin* dapat mengatur pengguna yang level autentikasinya *Enterprise Administrator.* Pengguna dengan level autentikasi *Enterprise Administrator* dapat mengatur pengguna yang level autentikasinya *Enterprise User.* Sedangkan pengguna dengan level autentikasi *Enterprise User* tidak dapat mengatur pengguna manapun.



1. Halaman Manage User untuk System Admin

Pada halaman ini pengguna dapat menambahkan pengguna / *user* dengan level autentikasi *Enterprise Administrator*. Ketika menambahkan *Enterprise Administrator*, pengguna juga akan memasukan data mengenai perusahaan / *Enterprise* yang diadministrasi pengguna *Enterprise Administrator*. Selain itu pengguna dapat melakukan *edit* data *Enterprise Administrator*. Pengguna juga dapat menghapus *Enterprise Administrator* jika diperlukan.

Ketika *System Admin*  ingin menekan tombol *add*  maka tampilan akan berubah seperti gambar di atas dan *System Admin*  dapat mulai memasukkan *input* data *Enterprise* dan *Enterprise Administrator*.

Setelah *System Admin* selesai melakukan input dan sistem tidak menemukan *error* pada input pengguna maka sistem akan menampilkan notifikasi yang menunjukkan bahwa *Enterprise Administrator* berhasil ditambahkan.

Ketika *System Admin* memilih menu *edit*, maka menu seperti gambar di atas akan muncul. *System Admin* dapat melakukan *edit* pada data pribadi *Enterprise Administrator* atau mengubah data *Enterprise* dari user tersebut.

Ketika *System Admin* memilih menu untuk menghapus *Enterprise Administrator* maka konfirmasi seperti gambar di atas akan muncul. Konfirmasi ini bertujuan untuk melakukan konfirmasi bahwa *System Admin* benar – benar ingin menghapus user tersebut.

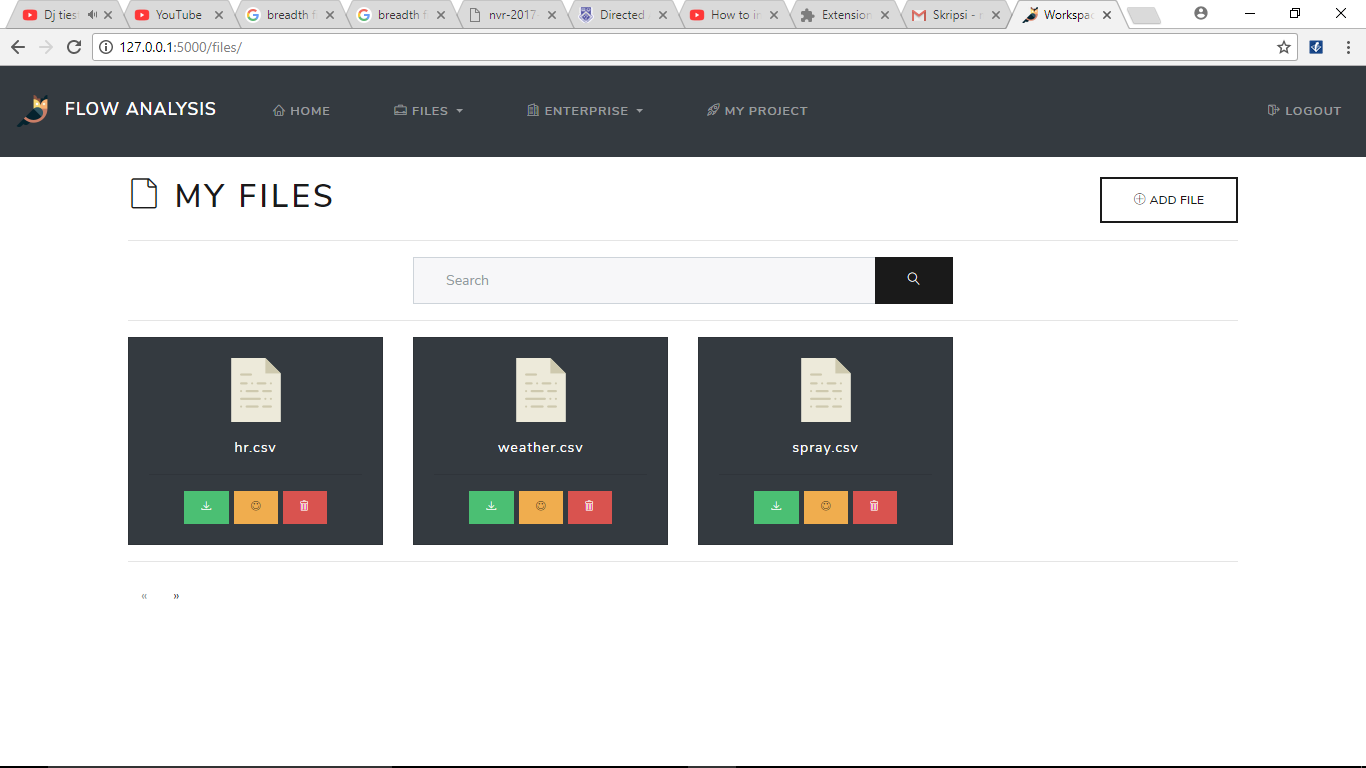
Pada halaman ini *Enterprise Administrator* dapat menambahkan *Enterprise User*, mengubah / *edit*  data *Enterprise User*, dan menghapus *Enterprise User.* Perbedaan tampilan pada Enterprise Administrator dan System Admin adalah pada *Add User* dan *Edit User*. Tampilan dan mekanisme penghapusan *user* sama seperti tampilan pada *System Admin.*

Pada menu ini *Enterprise Administrator* dapat menambahkan *Enterprise User*. Enterprise Administrator dapat memasukan input data *Enterprise User*, setelah *Enterprise User* dibuat maka akan muncul notifikasi seperti pada

Pada menu ini *Enterprise Administrator* dapat mengubah data *Enterprise User.* Menu ini dapat diakses ketika *Enterprise Administrator* menekan tombol *edit*.

### Halaman *Manage* *Files*

Halaman *Manage File* merupakan halaman yang digunakan untuk mengatur dan mengelola file yang dimiliki *user.* Pada halaman ini pengguna dapat melakukan *upload, download,* dan *share* file kepada pengguna lainya. Setiap kali pengguna melakukan *upload*  maka sistem akan memastikan bahwa file yang akan ter-*upload* tidak lebih dari kuota yang dimiliki perusahaan. Ketika *download* sistem juga akan memeriksa apakah akses yang datang memiliki sesi dan apakah sesi yang dimiliki memiliki ID yang sama dengan ID pemilik file yang ingin di-*download*. *Share* file merupakan fitur untuk memberikan akses kepada *user* lain dalam 1 perusahaan yang sama . Akses yang dapat diberikan terdiri dari 2 akses yaitu r (*read*) dan w(*read & write*)*.* User juga dapat menghapus *file* pada halaman ini. Ketika *user* menghapus *file* maka *file* tersebut akan masuk ke dalam *Enterprise Recycle Bin.*



1. Halaman Manage File

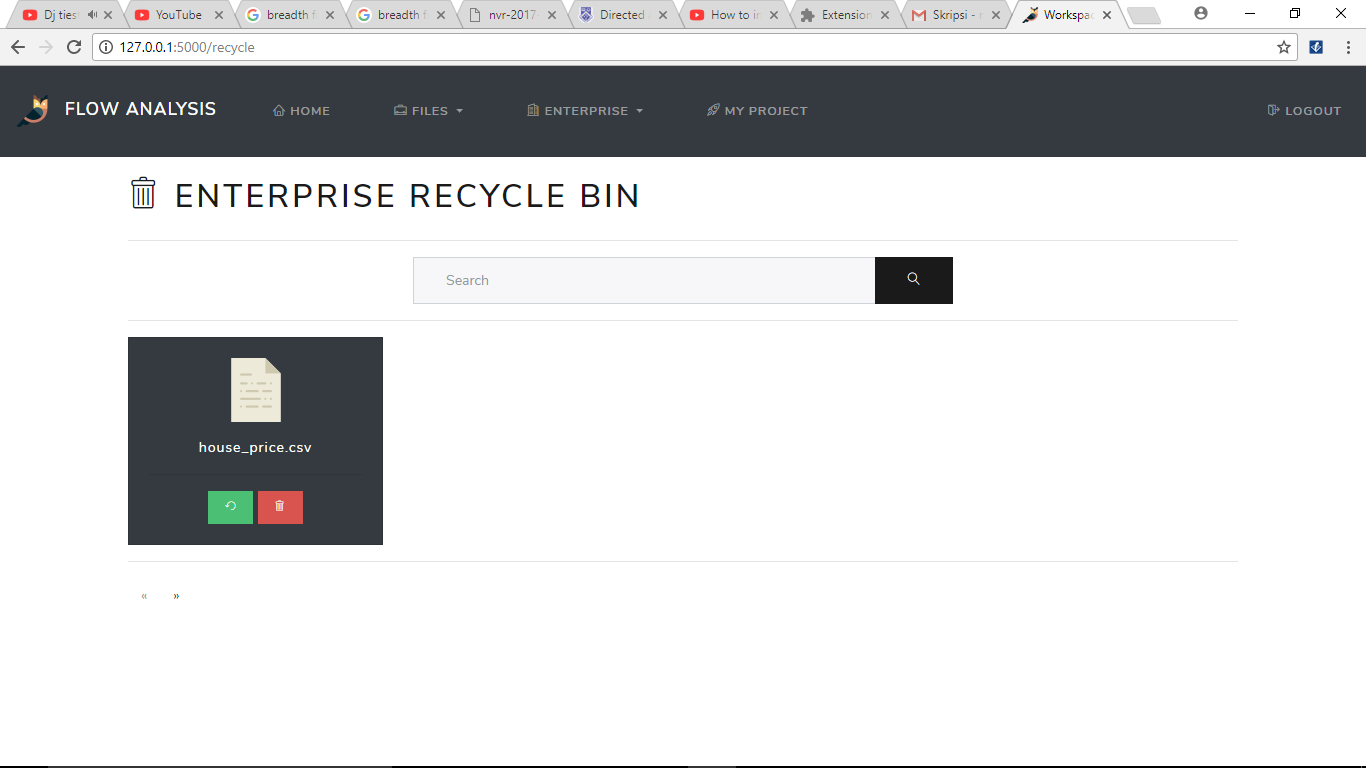
Jika pengguna menekan tombol *share* maka sistem akan menampilkan data user – user yang sebelumnya telah mendapatkan akses kepada file tersebut. Kemudian terdapat juga sebuah *form* untuk menambahkan user yang dapat melakukan akses ke file tersebut. Pilihan user ketika ingin *share* file tersebut adalah user dalam 1 perusahaan yang sama tetapi tidak memiliki akses kepada file tersebut.

### Halaman *Shared File*

Halaman *Shared File* merupakan halaman yang digunakan untuk melihat *file – file* yang bukan milik *user* *(*pemilik sesi*)* tetapi di-*share* *user* lainya. Pada halaman ini pengguna dapat melakukan *download* atau *view* jika pengguna memiliki *permission* untuk *read.* *User* juga dapat menggunakan *file* tersebut untuk membuat *project* baru jika *user* memiliki *permission* *write*.

### Halaman *Enterprise Recycle Bin*

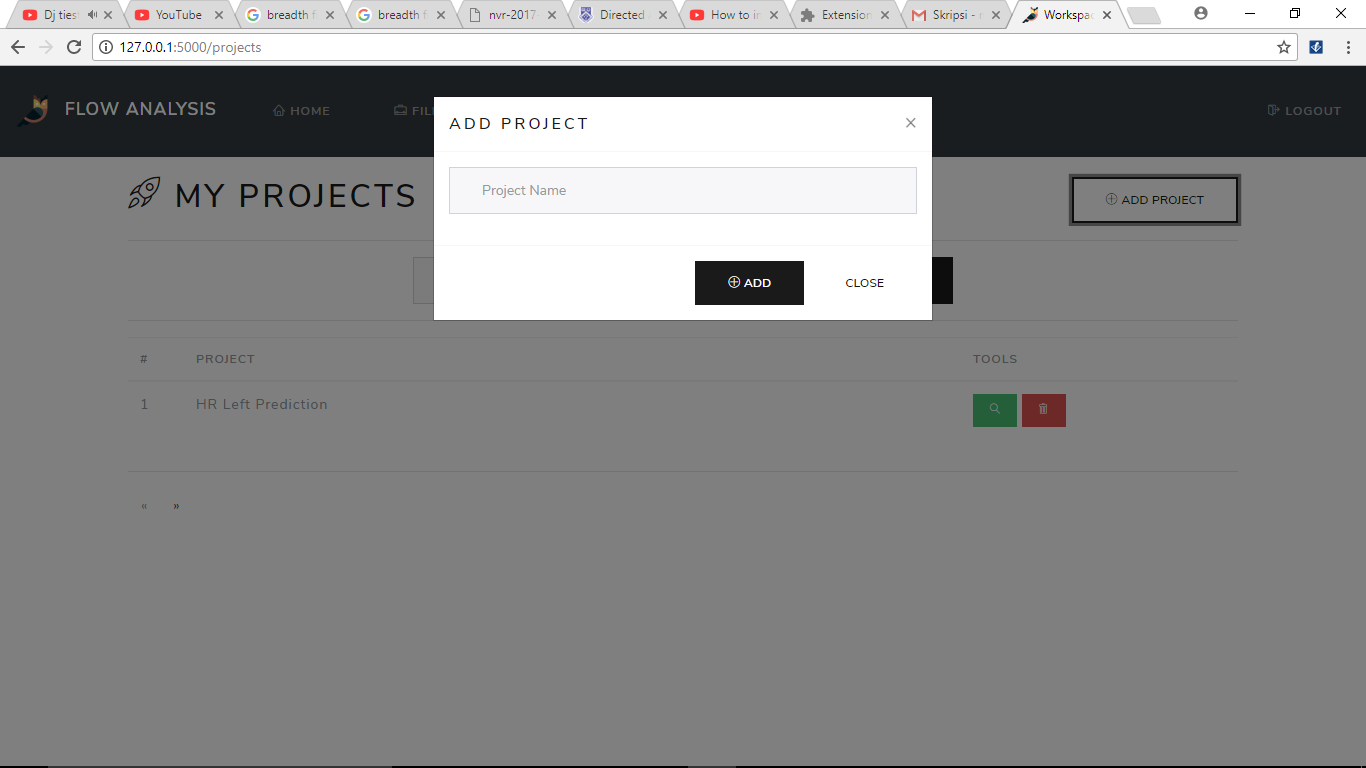
Halaman *Enterprise Recycle Bin* merupakan halaman yang dapat diakses oleh E*nterprise Administrator*. Halaman ini bertujuan untuk melihat *file – file* yang telah dihapus oleh *Enterprise User* pada 1 perusahaan / *Enterprise* yang sama. Pada halaman ini *Enterprise Administrator* dapat melakukan *purge file* atau *recover file*. *Purge* *file* merupakan tindakan yang dilakukan untuk menghapus *file* pada *database* dan *local file system*. *Recover file* dilakukan untuk mengembalikan *file* kepada pemilik aslinya yang dulunya telah menghapus *file* tersebut.



1. Halaman Enterprise Recycle Bin

### Halaman *Create Project*

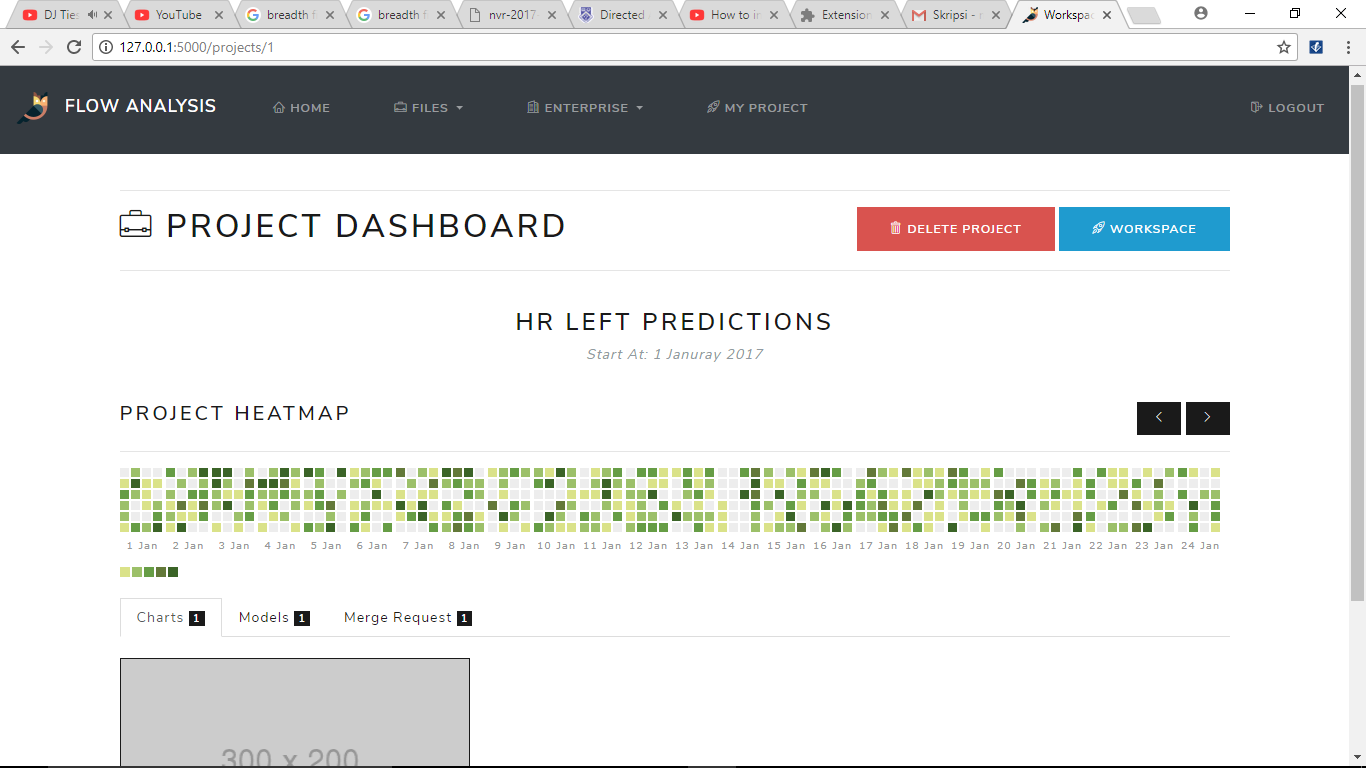
Halaman *Create Project* merupakan halaman yang digunakan untuk membuat sebuah *project.* Halaman ini akan meminta *user* untuk memasukkan data mengenai *project* tersebut seperti nama *project*, dan pengaturan *project.* Setelah user memasukkan data yang diinginkan maka sistem akan membuat sebuah *repository* baru yang merupakan tempat penampungan *project* tersebut. *Repository* akan dibuat dan diatur oleh Gitlab.



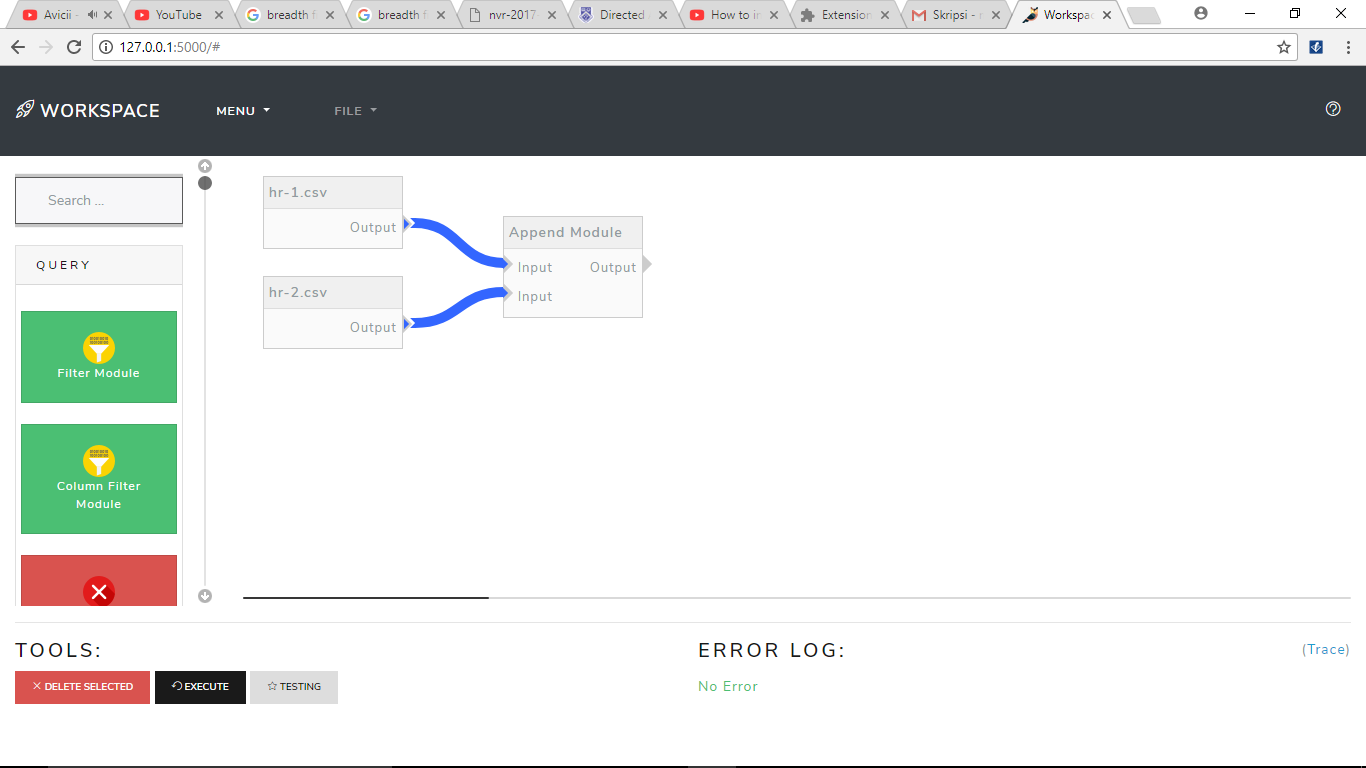
1. Halaman Create Project

### Halaman *Project*

Halaman *Project* merupakan halaman yang digunakan untuk mengatur dan melakukan analisa pada data yang diinginkan. Halaman *project* memiliki banyak fitur yang dapat digunakan untuk melakukan analisa, *query* data, *cleaning* data, membuat chart dan melakukan *recommend and analyze*.



1. Desain Halaman *Project*



1. Halaman *Workspace*

Pada halaman *project* *user* mendapatkan beberapa fitur yaitu:

1. *Insert Data*

*User* dapat menambahkan sebuah *record* atau baris pada data yang sudah ada, dan data yang diinputkan oleh *user* harus memiliki struktur yang sama dengan struktur data yang sedang dianalisa.

1. *Update Data*

*User* dapat mengubah data sebuah kolom tertentu yang dipilih *user* dengan data baru yang juga diinputkan oleh *user.* Data yang diubah dapat seluruh data atau *user* dapat memberikan filter data mana saja yang akan diubah.

1. *Delete Data*

*User* dapat menghapus baris tertentu yang dipilih *user.User* dapat memberikan filter data mana saja yang akan diubah.

1. *Filter Data*

*User* dapat melakukan filter data dan menampilkan data yang telah difilter oleh *user.* Filter yang dilakukan adalah sesuai dengan aturan yang diberikan oleh *user.*

1. *Describe*

*User* dapat menggunakan menu ini untuk menampilkan *summary* data yang sudah ada. Summary dari data terbagi menjadi 2, *Summary Numerical* dan *Summary Categorical.*

1. *Correlation Table (Paerson)*

*User* dapat menggunakan menu ini untuk menampilkan tabel korelasi antar kolom yang ada pada data. Analisa korelasi ini hanya dapat menggunakan kolom pada data yang memiliki tipe data *numeric*

1. *Create Chart*

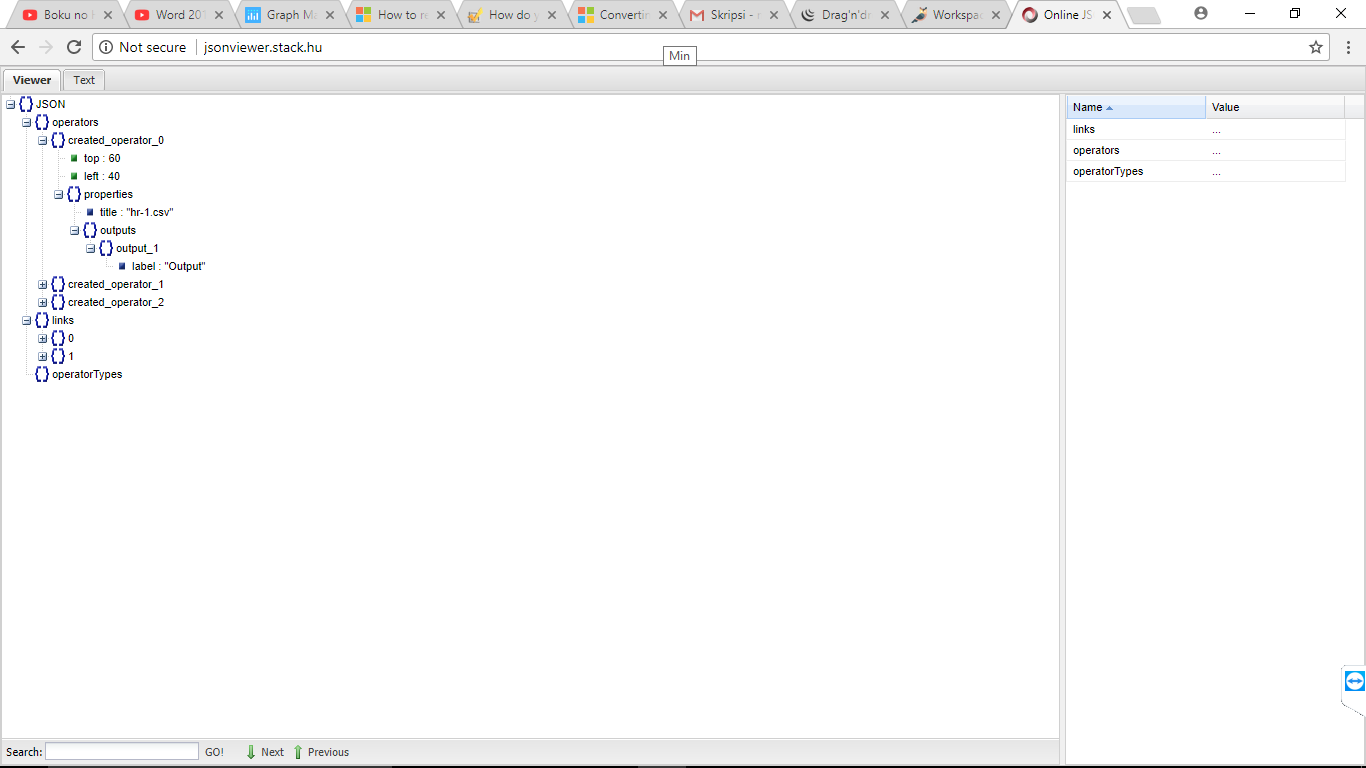
*User* dapat membuat beberapa grafik/c*hart*  yang dapat menggambarkan data tersebut. Grafik yang telah terbuat dapat di-*save* kedalam *home directory user.* Ketika *user* ingin menambahkan grafik, maka pengguna akan memilih grafik mana yang ingin digunakan, setelah itu *user* akan memasukkan parameter yang diperlukan untuk membuat grafik tersebut.

1. *Recommend & Analyze*

*User* bisa mendapatkan rekomendasi metode prediksi dari fitur ini. Ketika *user* menekan tombol *Recommed & Analyze* maka user akan mendapatkan *summary* dari *sampling* yang dilakukan sisten dalam bentuk urutan metode beserta hasil validasi yang sudah dilakukan oleh sistem. Rekomendasi dari sistem diharapkan dapat membantu *user* dalam pengambilan keputusan selanjutnya.

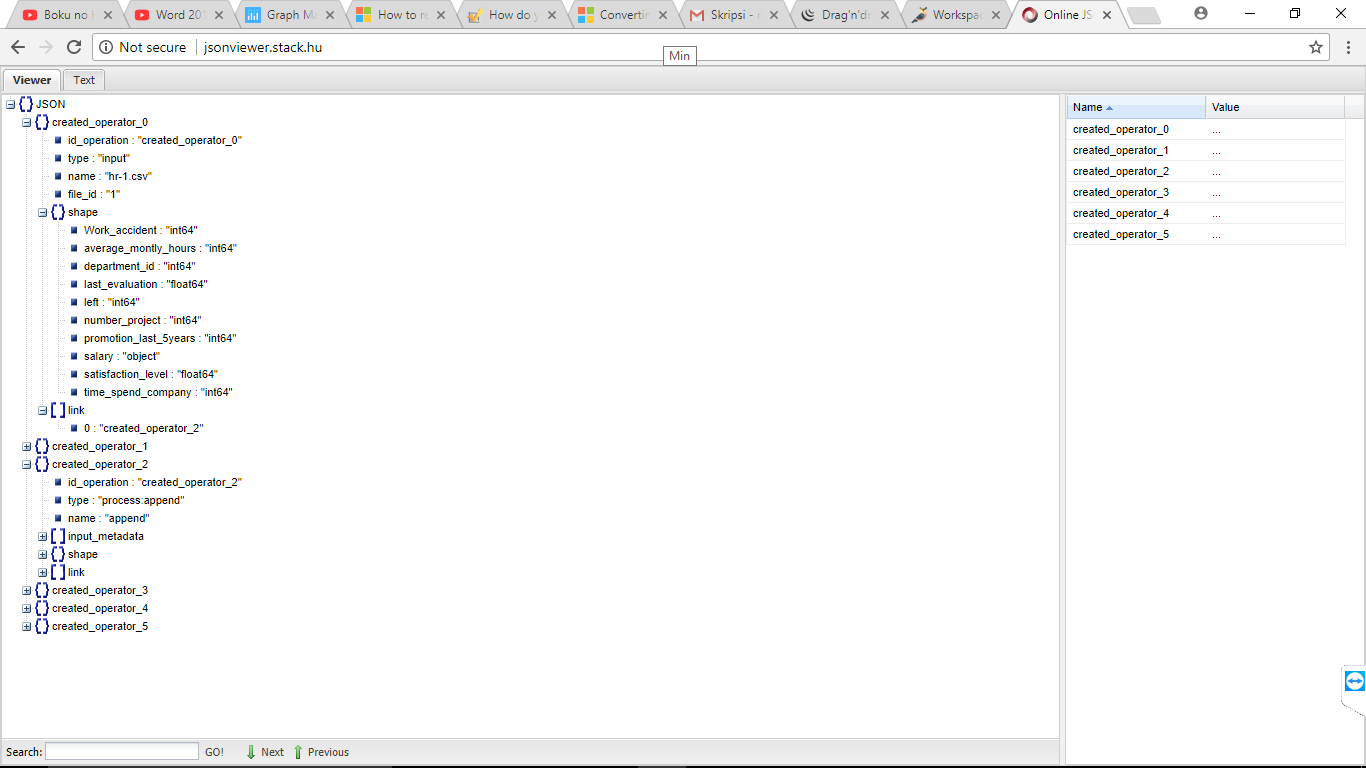
## Desain Metadata

*Metadata* merupakan data dalam format JSONyang dibentuk untuk memberikan penjelasan mengenai *schema* dan *parameters* pada halaman *workspace. Schema* merupakan data dengan format JSON yang dibentuk untuk memberikan informasi mengenai bentuk *Directed Acyclic Graph* yang telah dibentuk *user* pada halaman *workspace.*



1. Bentuk Metadata Schema

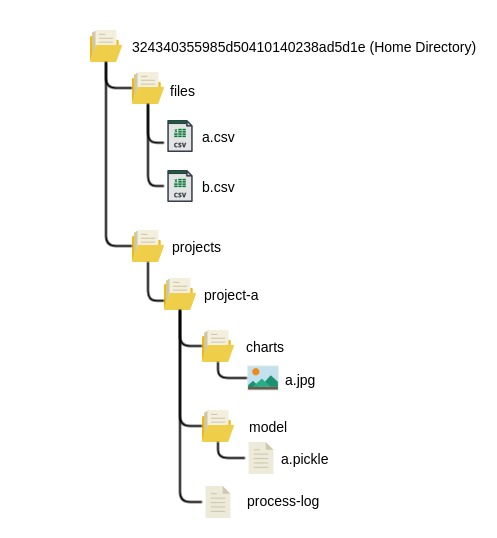
Parametersi merupakan metadata yang dibentuk untuk membantu schema memberikan data parameter yang terjadi ketika user menambahkan input pada module.



1. Bentuk Metadata Parameters

## Desain Struktur *Home Directory User*

*Home Directory User* merupakan tempat *user* menyimpan *project*, *file* dan gambar *chart*. *Home Directory* ini tersimpan dalam penyimpanan lokal yang memiliki struktur pernyimpanan sebagai berikut:



1. Struktur Home Directory User