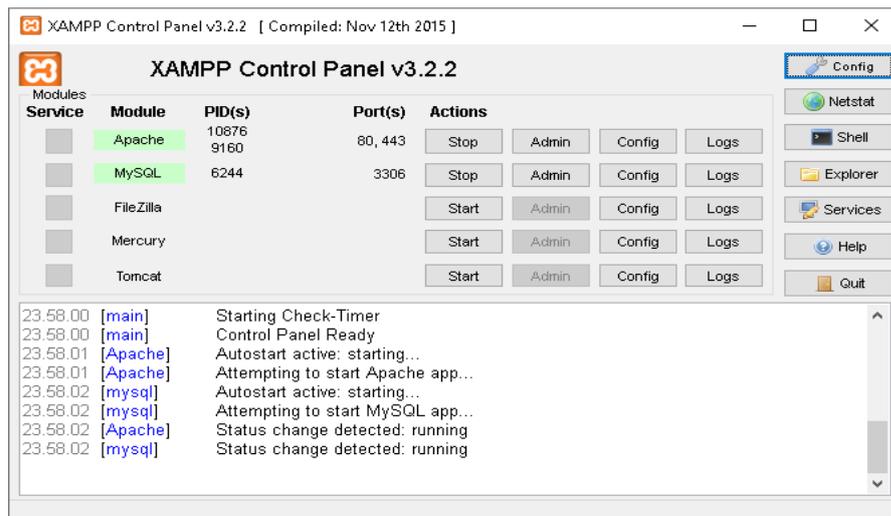


CARA MENJALANKAN PROGRAM

SISTEM PENCARIAN BARANG HILANG

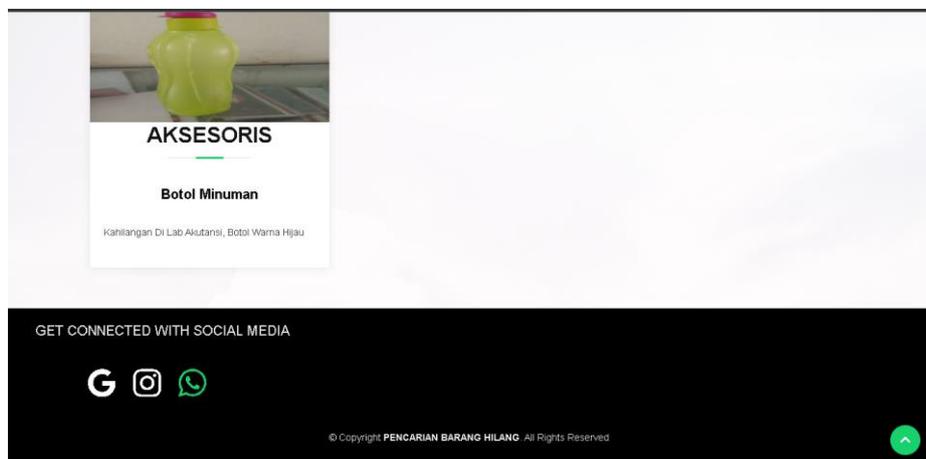
1. XAMPP



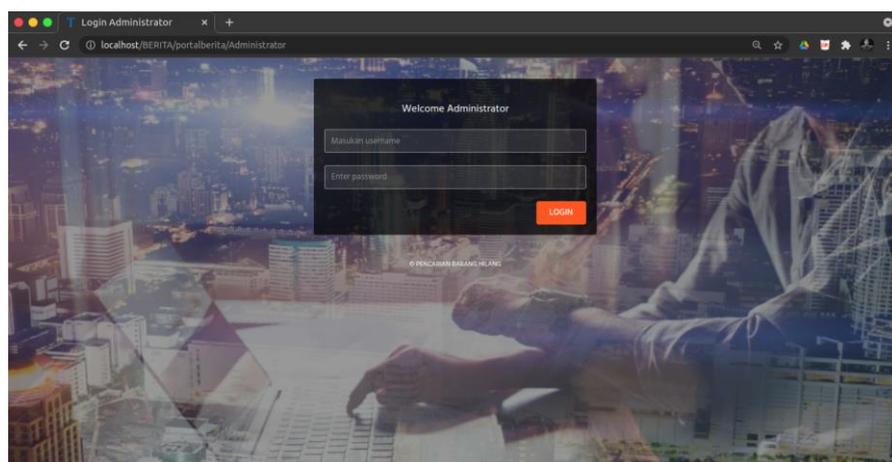
- Menjalankan aplikasi xampp dengan mengaktifkan Module Apache dan MySQL.
- Selanjutnya menjalankan aplikasi browser yaitu Google Chrome sebagai test program yang sudah dibuat.
- Menggunakan alamat Url <http://localhost/BERITA/portalberita/>, sebagai alamat utama

2. Tampilan Halaman Utama Program Sistem Pencarian Barang Hilang



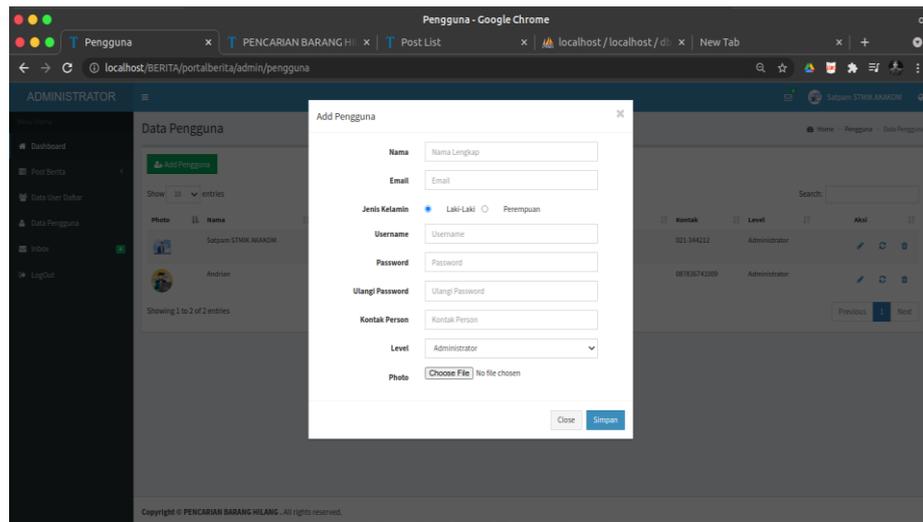


3. Tampilan Login Admin



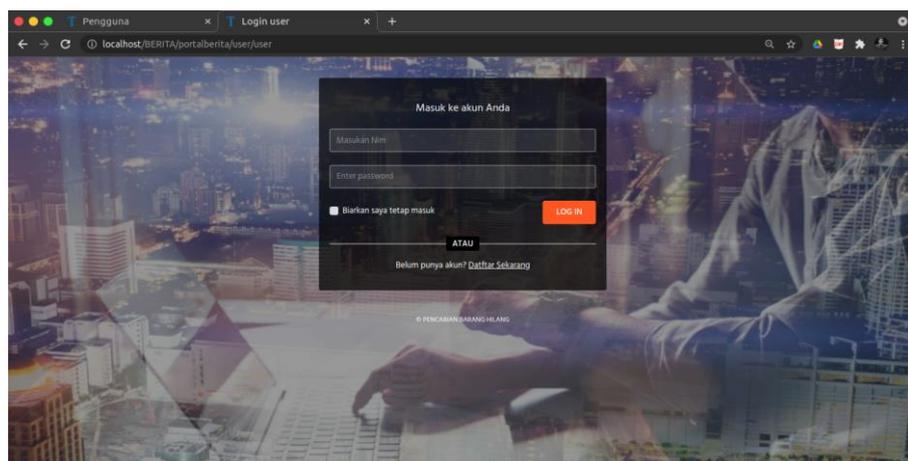
Tampilan yang digunakan untuk login admin menuju dashboard, yang dimana memberikan izin hak akses kepada satpam yang sudah terdaftar di dalam database untuk masuk ke dalam aplikasi

4. Tampilan Register Pengguna (Admin)



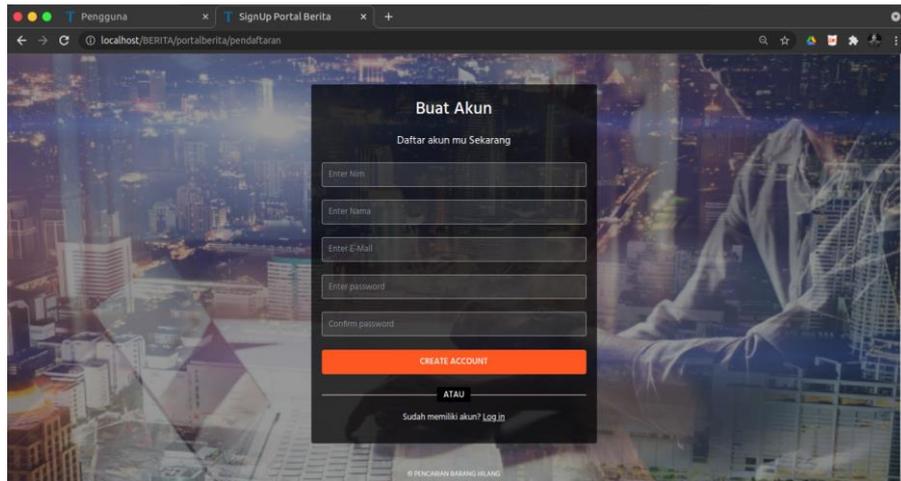
Tampilan yang digunakan untuk melakukan *input* data pengguna akan digunakan sebagai data *login* admin.

5. Tampilan Login User



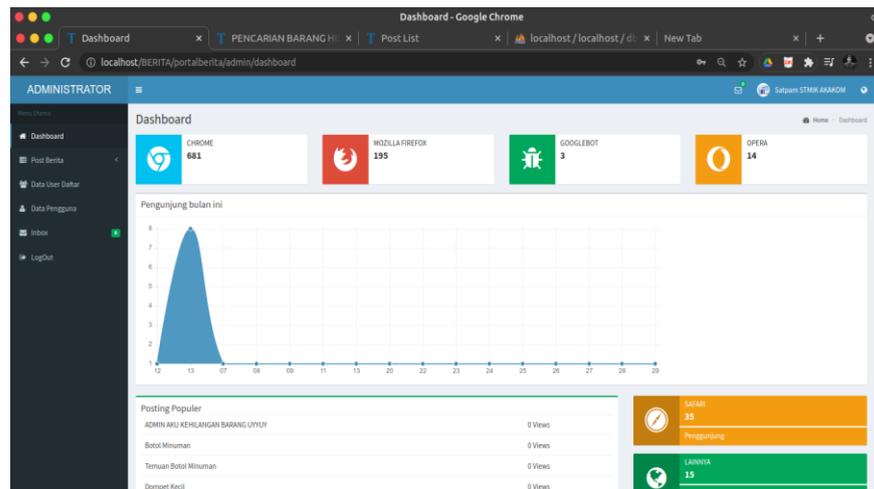
Tampilan yang digunakan untuk login *user* menuju dashboard, yang dimana memberikan izin hak akses kepada mahasiswa yang sudah terdaftar di dalam database untuk masuk ke dalam aplikasi.

6. Tampilan Register *User* (Mahasiswa)



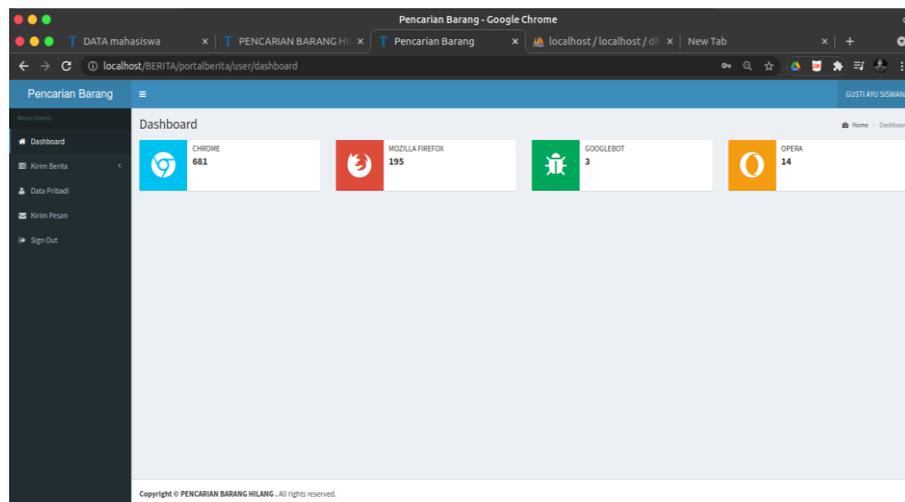
Tampilan yang digunakan untuk melakukan input data *user* untuk disimpan kedalam database yang nantinya akan digunakan sebagai data login *user*.

7. Tampilan Dashboard Pengguna (Admin)



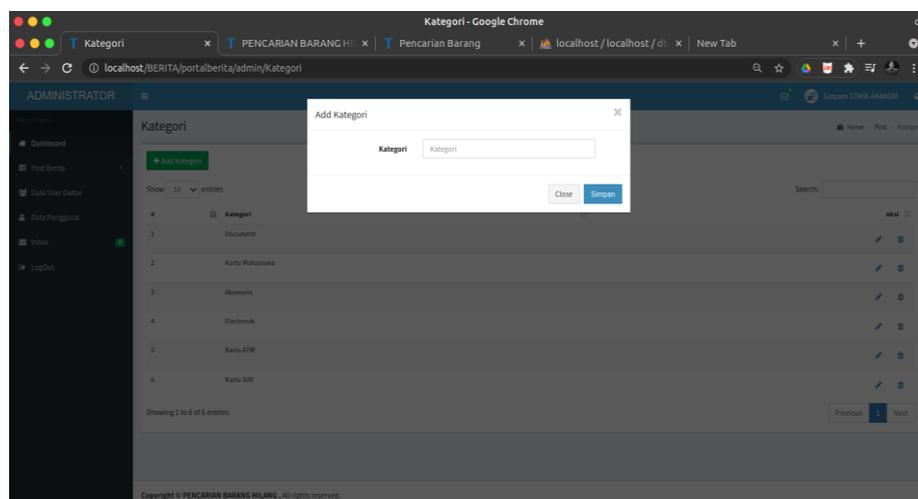
Tampilan awal setelah satpam sebagai admin *login* dimana aplikasi ini menggunakan SPA (*Single Page Applications*), yang dimana admin bisa melakukan *input*, *edit* dan hapus data tulisan berita barang hilang di dalam satu halaman.

8. Tampilan Dashboard *User* (Mahasiswa)



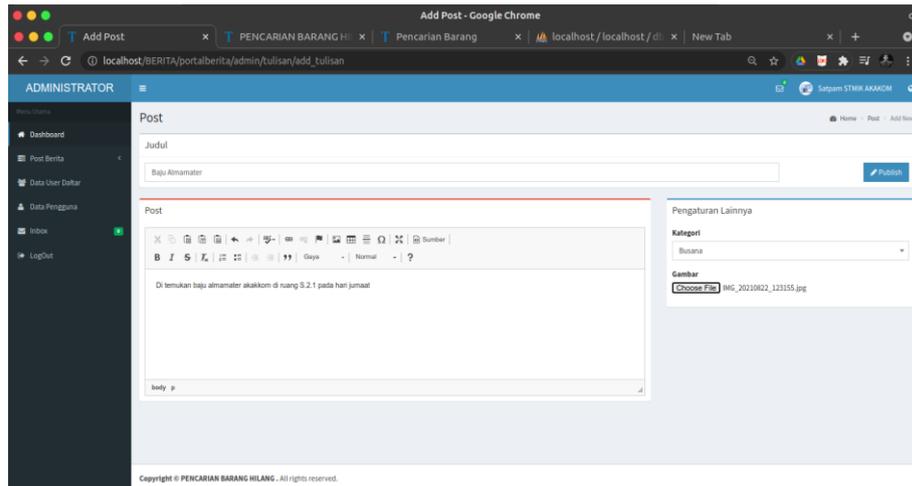
Tampilan awal setelah *user login*, yang dimana *user* bisa melakukan *input* dan *edit* data tulisan berita barang hilang dan *input* data inbox untuk melakukan pengiriman pesan.

9. Tampilan Tambah Kategori



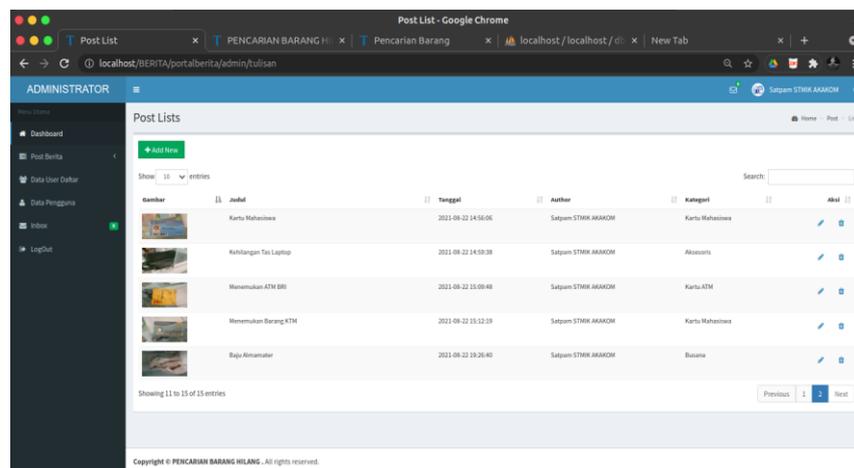
Tampilan yang digunakan untuk *input* data kategori baru yang akan menjadi sebuah fitur pada proses penginputan data tulisan barang hilang yang dilakukan oleh *user* dan pengguna.

10. Tampilan Tambah Tulisan Pengguna (Admin)



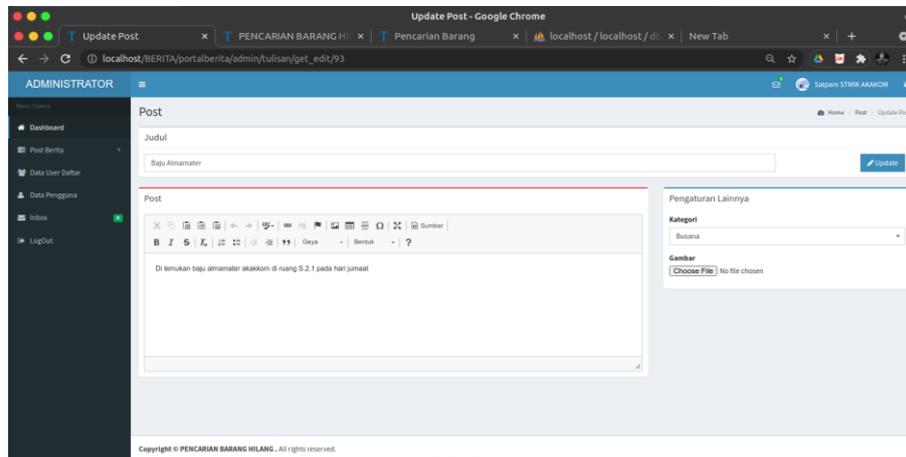
Tampilan yang digunakan untuk *input* data tulisan berita kehilangan.

11. Tampilan List Berita



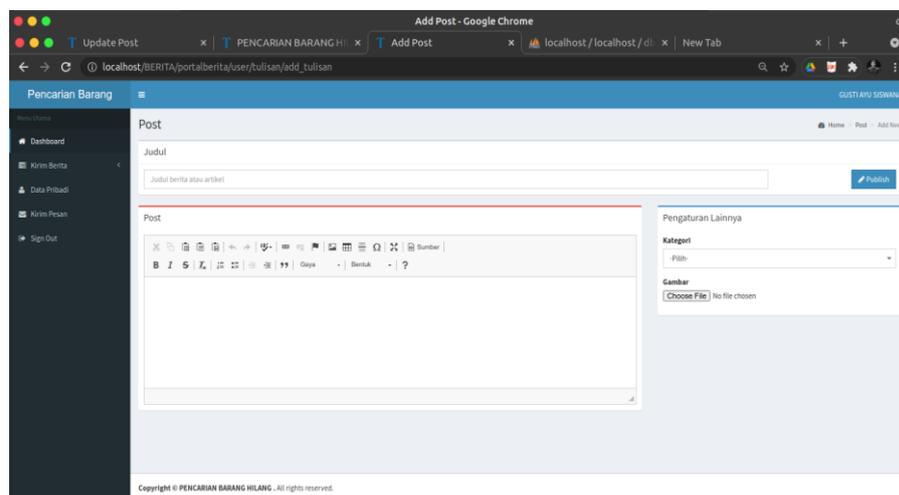
Tampilan data tulisan barang hilang yang diinputkan oleh admin, dan digunakan sebagai penyampain informasi seputar barang hilang, sehingga seluruh warga STMIK AKAKOM dapat mengetahui barang apa saja yang hilang.

12. Tampilan Edit Tulisan



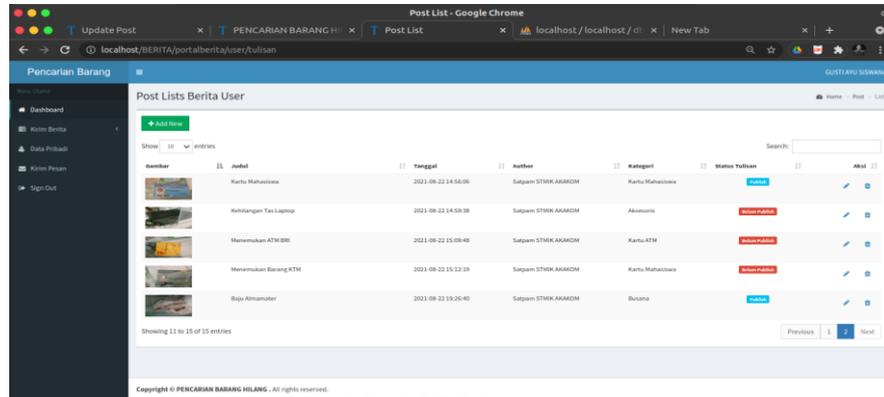
Merupakan proses untuk melakukan *edit* data tulisan berita barang hilang.

13. Tampilan Tambah Tulisan *User* (Mahasiswa)



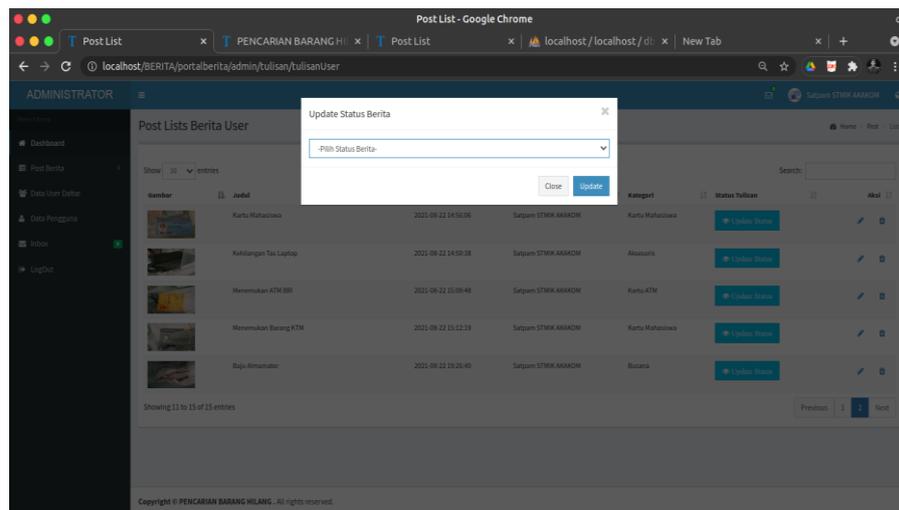
Merupakan proses *input* data tulisan yang dilakukan oleh *user* sebagai data pengajuan berita barang hilang.

14. Tampilan List Berita *User* (Mahasiswa)



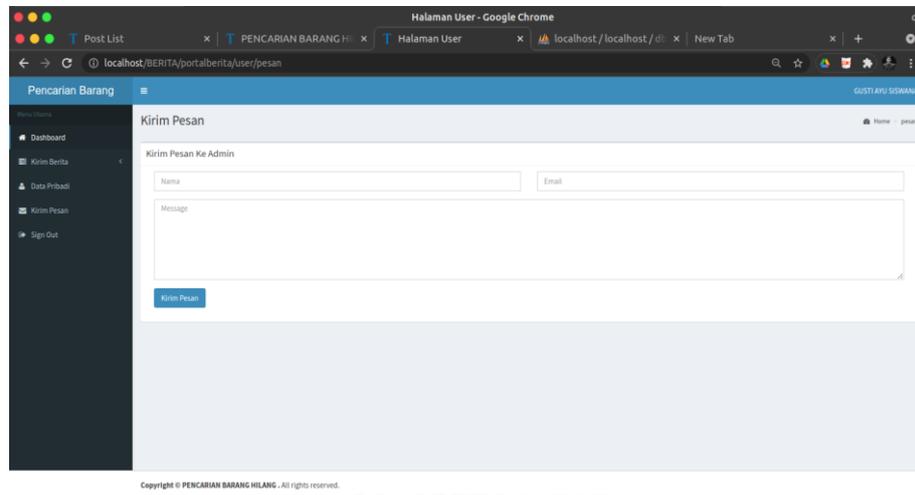
Merupakan tampilan hasil inputan data tulisan berita yang dilakukan oleh *user*.

15. Tampilan Pengajuan Tulisan Berita *User* (Mahasiswa)



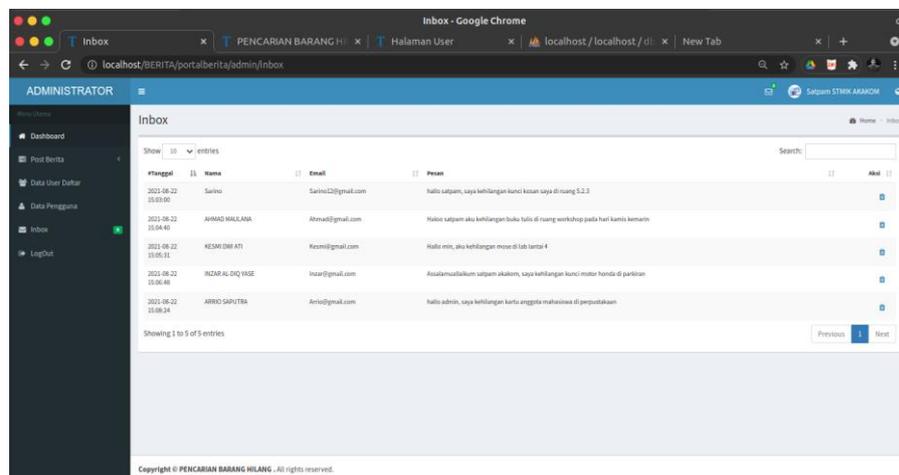
Merupakan tampilan data tulisan *user* untuk dilakukan proses validasi, dimana data tersebut akan dilakukan proses *update* status dan digunakan untuk melakukan publish data tulisan, kemudian data tulisan akan ditampilkan sebagai berita barang hilang.

16. Tampilan Tambah Inbox *User* (Mahasiswa)



Tampilan yang digunakan untuk menginputkan data inbox sebagai proses pengirim pesan seputar barang hilang ke admin, kemudian data inbox akan disimpan ke database.

17. Tampilan Inbox



Tampilan data *inbox* yang dikirim oleh *user* sebagai penyampaian pesan atau admin dan dilakukan proses pengecekan apakah pesan yang dikirim tersebut masih dalam seputar barang hilang atau tidak, jika isi pesan tersebut valid maka akan dilakukan proses penanganan.

18. POSTMAN

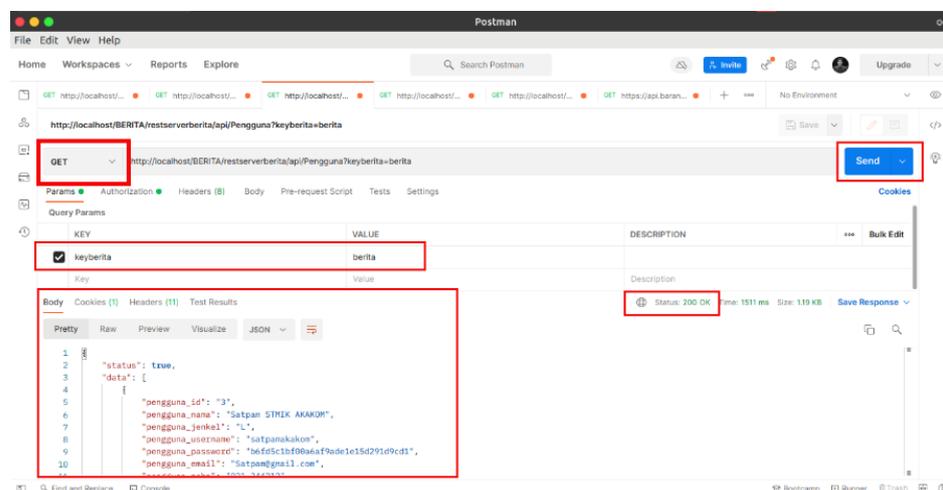


- Mengaktifkan aplikasi Postman sebagai test tool API
- Selanjutnya menggunakan alamat URL API

<http://localhost/BERITA/restserverberita/...../> dengan memanggil

service-service yang ingin digunakan sebagai test API

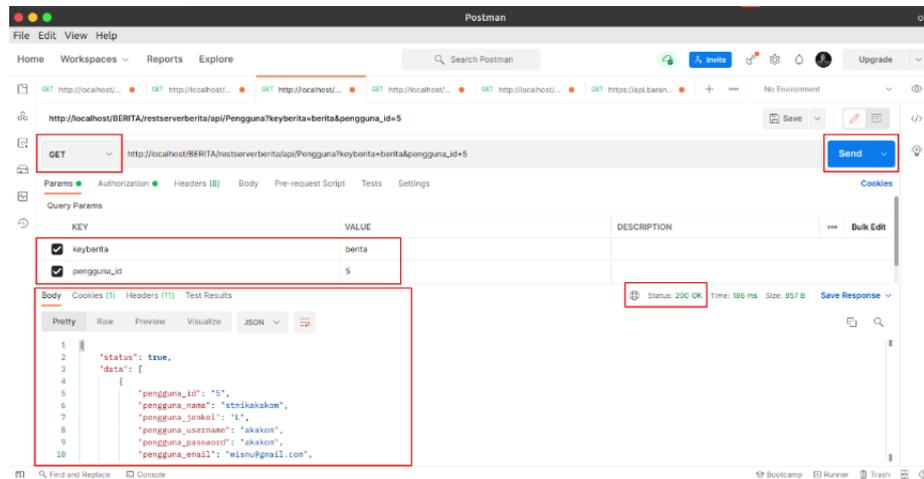
19. Tampilan GET Testing



Merupakan proses *request* data pengguna dengan menggunakan metod

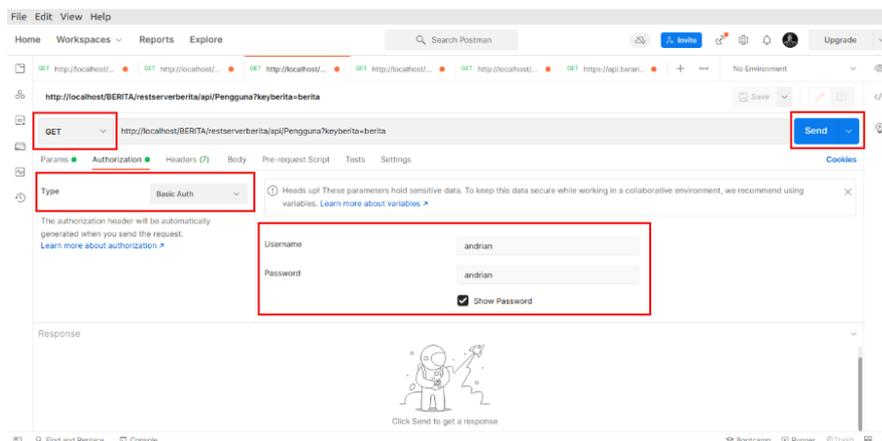
get.

20. Tampilan GET Testing Berdasarkan *Id*



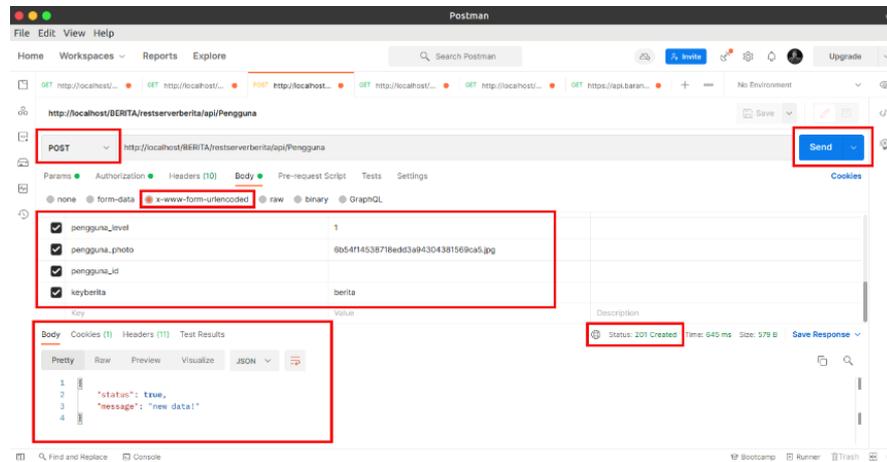
Merupakan proses *request* data pengguna dengan menggunakan metode *get* untuk menampilkan data pengguna berdasarkan *Id*.

21. Tampilan *Authorization* Rest API



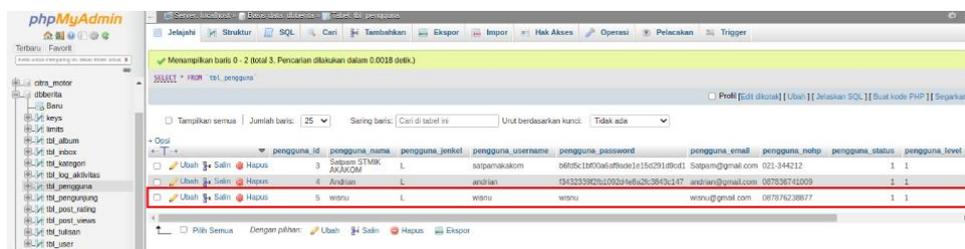
Merupakan teknik otentikasi *http request* dengan menggunakan *type basic auth*, metode ini membutuhkan informasi *username* dan *password* untuk disisipkan dalam *header request*, *authentication* ini diperlukan untuk membatasi akses pada API, karena API bersifat *stateless* atau tidak menggunakan *session* untuk menyimpan *auth*.

22. Tampilan POST Testing



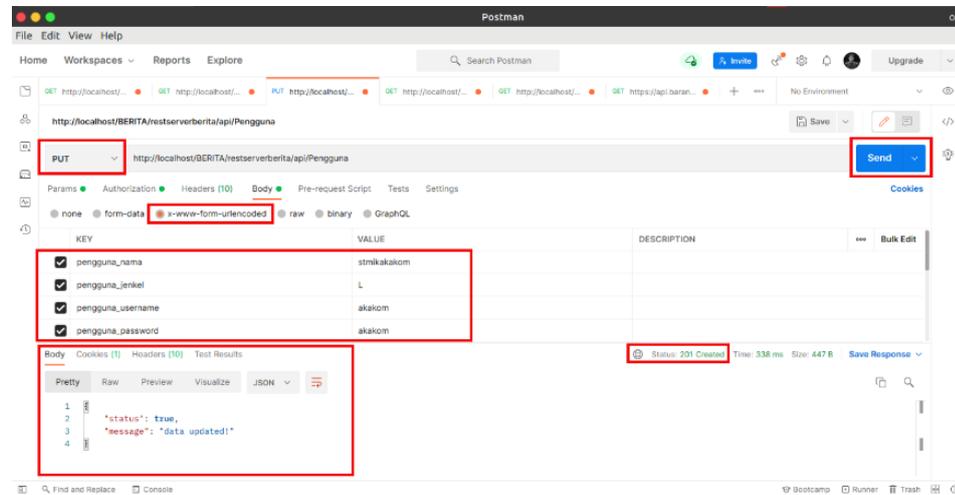
Untuk mengirimkan data baru dari *client* ke *server* REST API, sebagai contohnya digunakan untuk menambahkan data pengguna baru dengan menyertakan API *key*.

23. Tampilan Hasil POST Testing



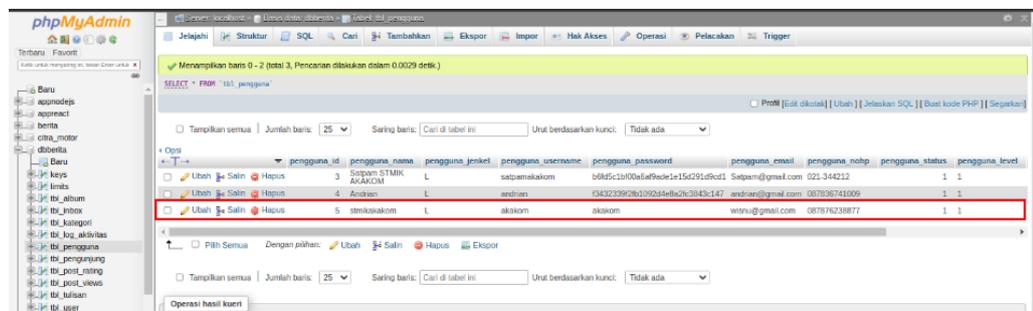
Merupakan hasil proses *request* data pengguna dengan menggunakan metode *post*, terlihat bahwa data yang sudah di *post* akan tersimpan pada tabel pengguna *database* dbberita di MySQL.

24. Tampilan PUT Testing



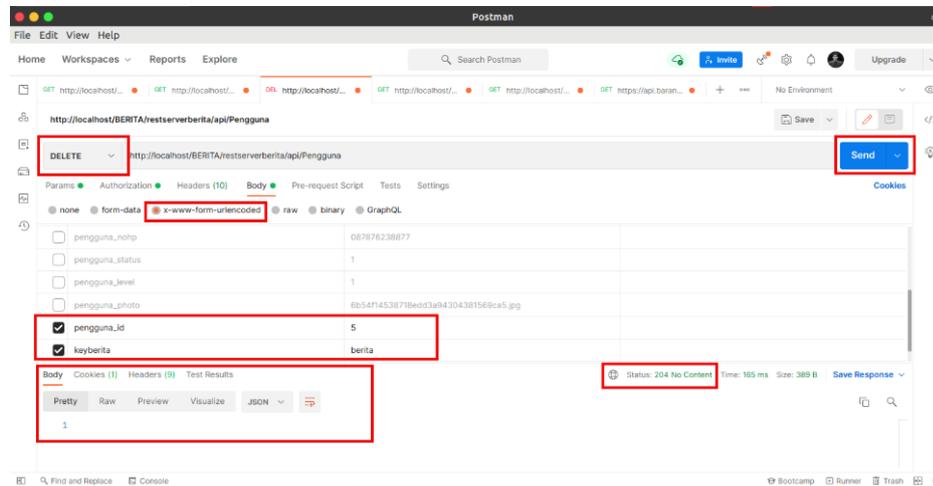
Merupakan proses *request* data pengguna dengan menggunakan metod *put* untuk memperbarui data pengguna, cukup dengan memanggil *controller url* pengguna.

25. Tampilan Hasil PUT Testing



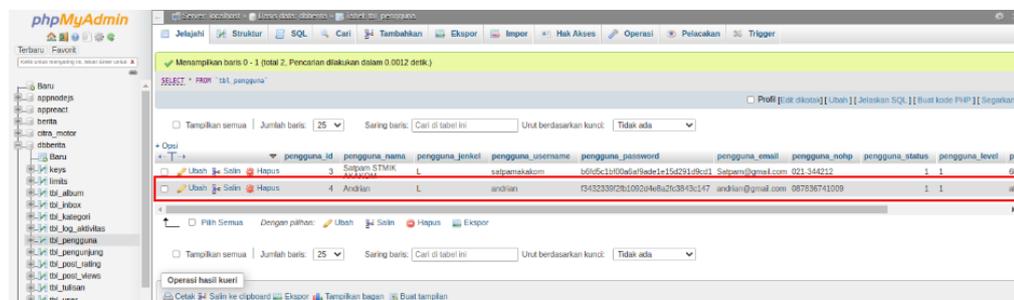
Merupakan hasil proses *request* data pengguna dengan menggunakan metod *put*, terlihat bahwa data yang sudah di *put* akan tersimpan pada tabel pengguna *database* dbberita di MySQL.

26. Tampilan DELETE Testing



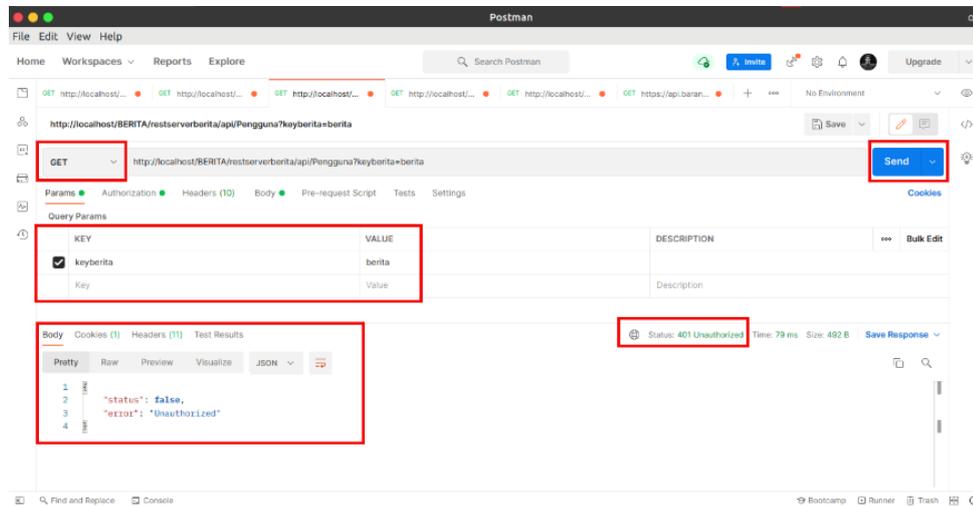
Merupakan proses *request* data pengguna dengan menggunakan metode *delete* untuk menghapus data pengguna, cukup dengan memanggil *controller url* pengguna.

27. Tampilan Hasil DELETE Testing



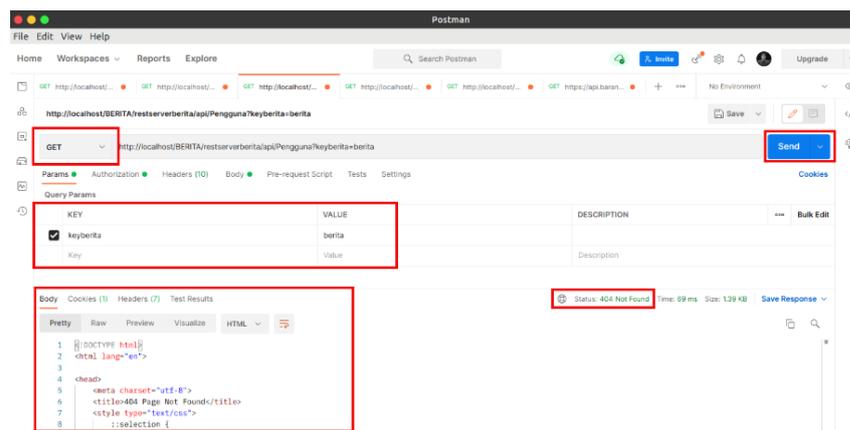
Merupakan hasil proses *request* data pengguna dengan menggunakan metode *delete*, terlihat bahwa data yang sudah di *delete* akan menghilang pada tabel pengguna *database* dbberita di MySQL.

28. Tampilan *Status Server Hak Akses API*



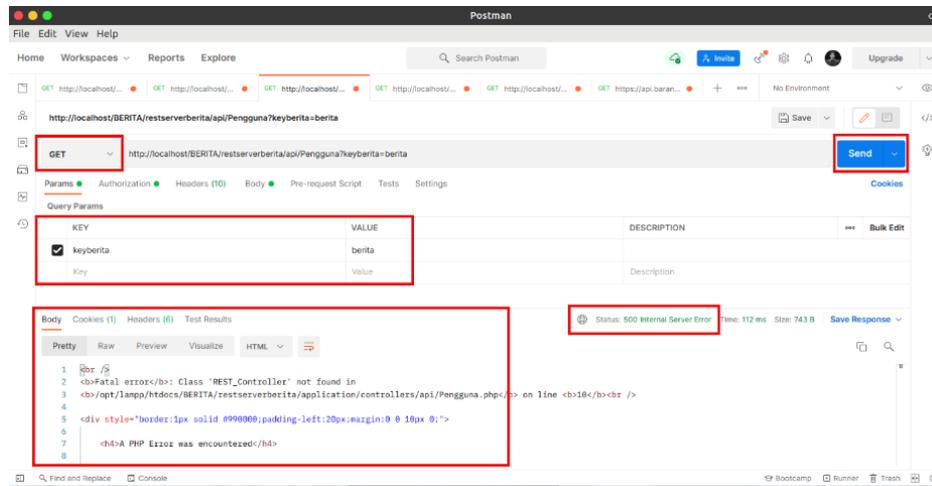
Merupakan proses *request* data yang membutuhkan otentikasi, status *response* 401 *unauthorized error* ini muncul karena permintaan *request* tak bisa memberikan bukti kewenangan yang sah saat diminta *server*, sehingga memerlukan proses menginputkan *username* dan *password* pada bagian *authorization* dengan *type basic auth*.

29. Tampilan *Maintenance Server API*



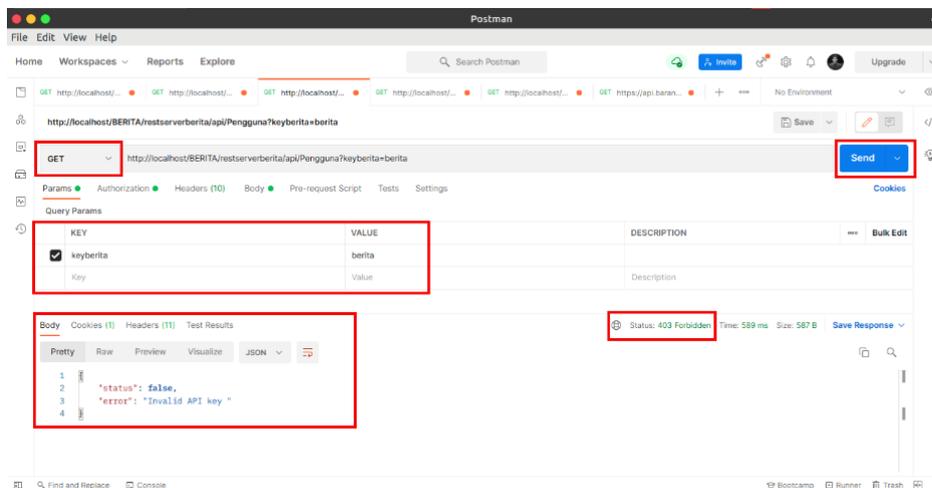
Merupakan proses *request* data yang tidak ditemukan, status *response* 404 *not found* inilah kode status tiga digit paling terkenal di jagat maya, *error* ini menunjukkan data yang diminta itu tidak ada di *server*.

30. Tampilan GET Server Error



Merupakan proses *request* data tidak dapat memenuhi permintaan, status *response* 500 adanya kesalahan di *server* atau *internal server error*.

31. Tampilan Invalid API Key



Merupakan proses *request* data terlarang, status *response* 403 *forbidden* ini dikarenakan menonaktifkan *key* pada sisi *backend* di *restserver*.