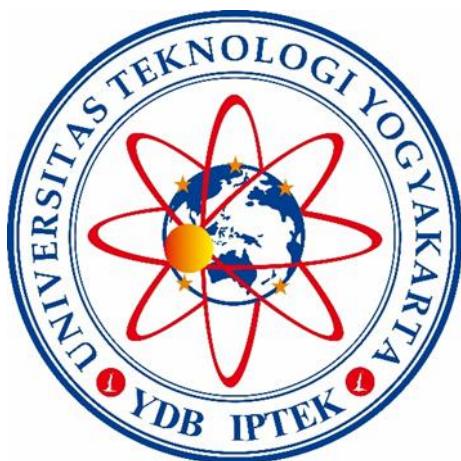


## **RESPONSI 2**

### **PEMROGRAMAN MOBILE PRAKTIKUM**



Disusun Oleh:

Nama : Yusuf Bhaskara Adjı

NIM : 5190411129

Kelas : Pemrograman Mobile Praktikum VII

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS SAINS & TEKNOLOGI**  
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**  
**2021/2022**

## Responsi 2

### Source Code :

- Github

<https://github.com/yusufadji/Responsi-2>

- Google Drive

[https://drive.google.com/drive/folders/1V6fMFl6lgOdTe6JqYupQK8xY\\_x1gznrS?usp=sharing](https://drive.google.com/drive/folders/1V6fMFl6lgOdTe6JqYupQK8xY_x1gznrS?usp=sharing)

### Note :

Untuk testing aplikasi ini dapat menggunakan identitas berikut:

Admin UserID: 08987654321

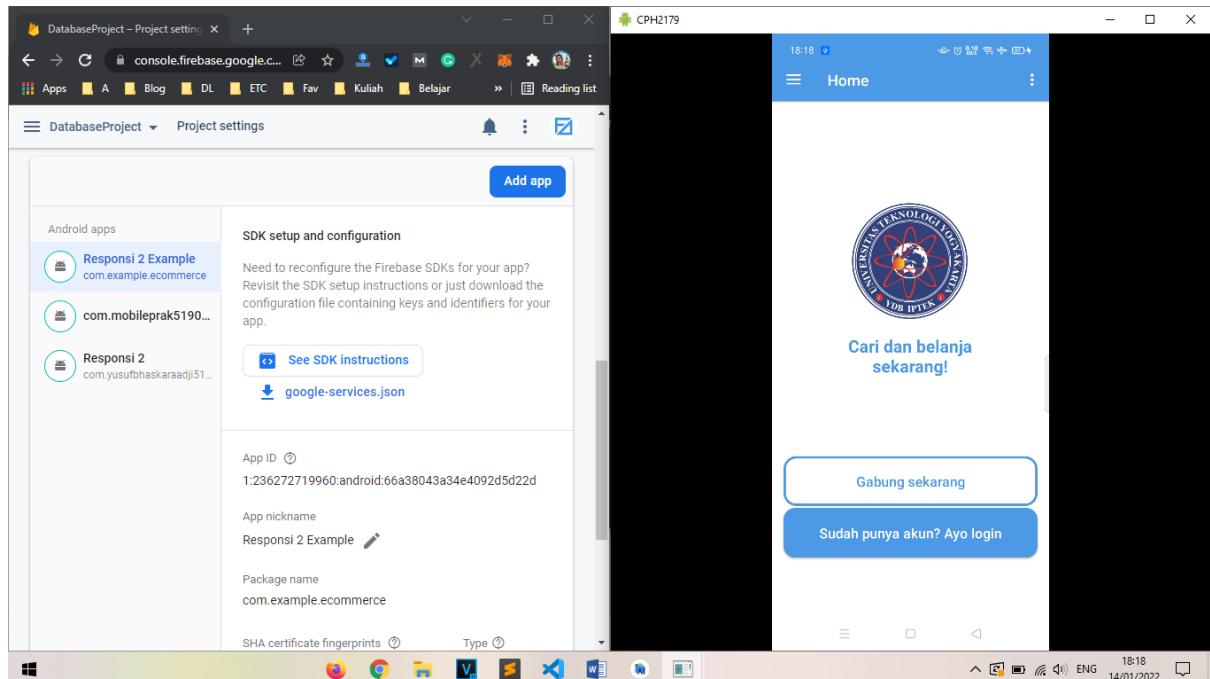
Admin Password: 123

User UserID: 08123456789

User Password: 123

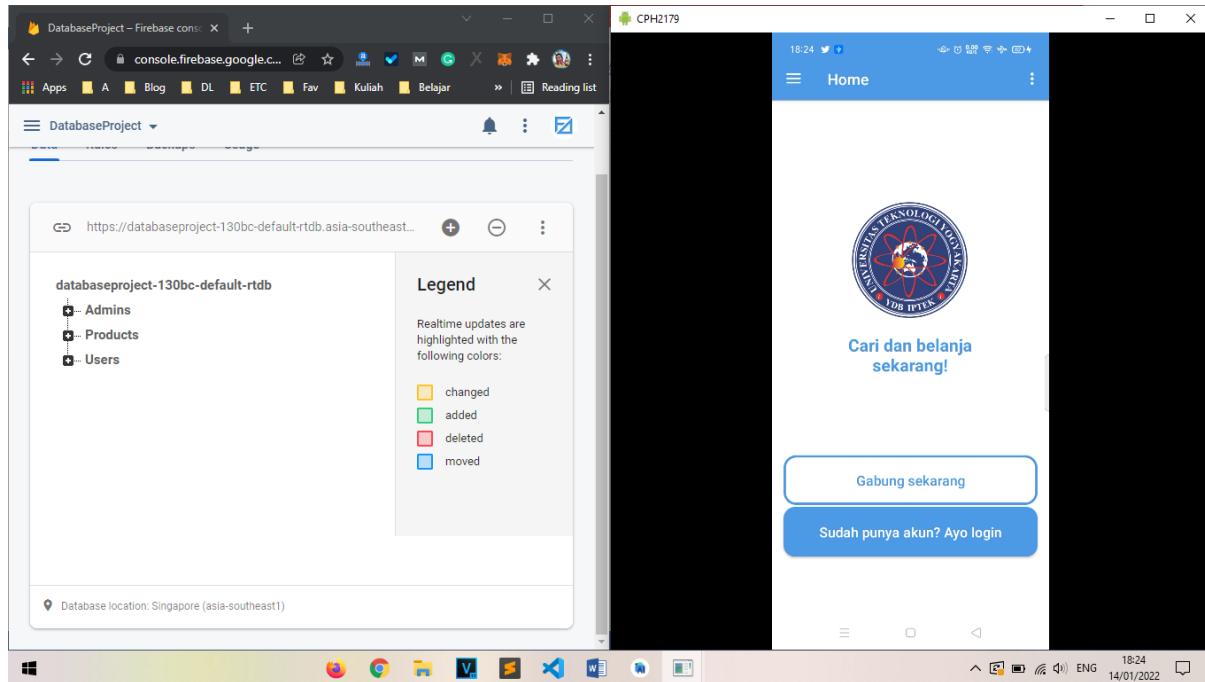
### Screenshot Running Aplikasi

Disini saya mengambil referensi contoh dari repository github yang Bapak berikan. Saya melakukan modifikasi dari segi tampilan dan koneksi databasenya.



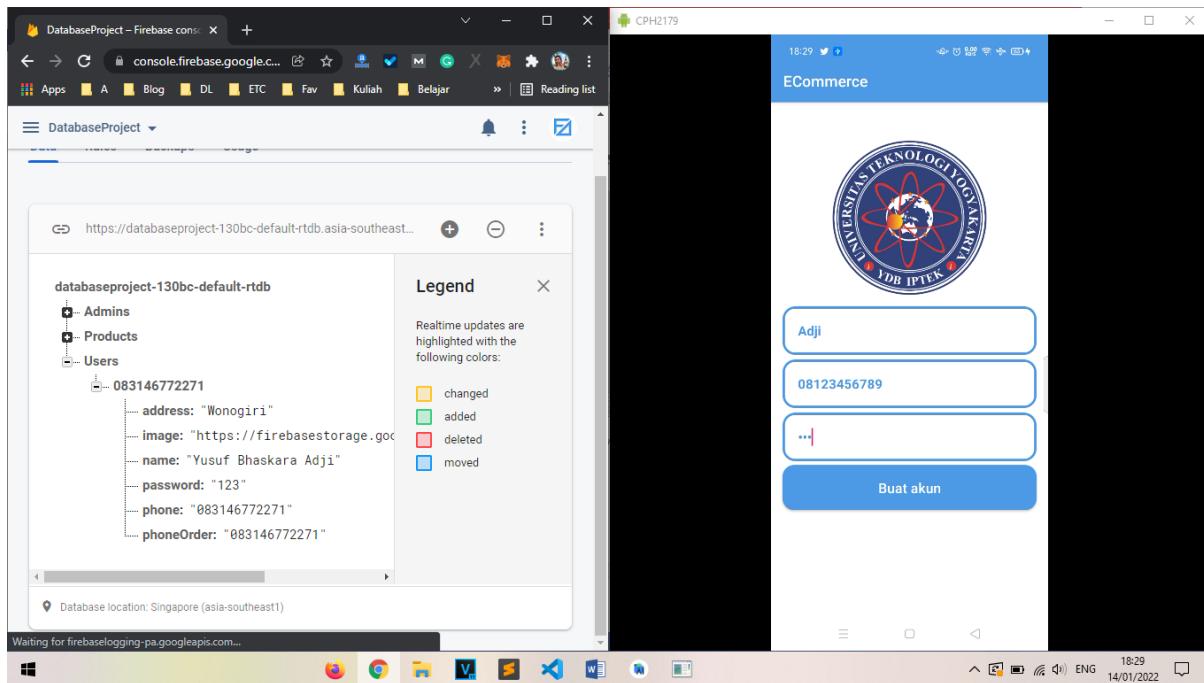
Gambar 1 - Tampilan aplikasi ketika pertama kali dijalankan

Pertama saya menambahkan aplikasi baru di project Firebase dengan nama package yang sama dengan repository tersebut. Hal ini perlu dilakukan agar aplikasi dapat terhubung dengan Firebase Realtime Database di project saya. Lalu saya mengunduh file google-services.json yang disediakan oleh Firebase dan mereplacenya dengan yang ada di project. Untuk key-value pairs nya saya samakan dengan value yang ada di kode project tersebut sehingga Firebase Realtime Databasenya kurang lebih akan terlihat seperti ini:



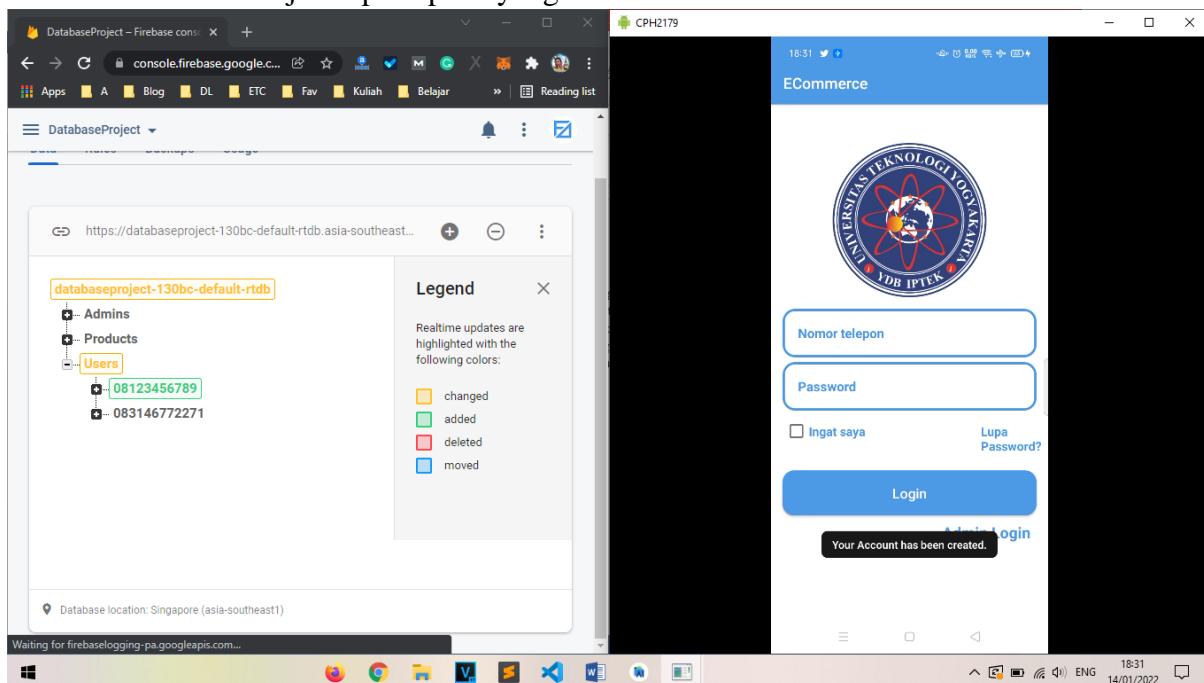
Gambar 2 - Tampilan Firebase Realtime Database

Selanjutnya kita masuk ke tampilan Register akun. Activity ini dapat diakses ketika user menekan tombol Gabung sekarang di tampilan sebelumnya pada Gambar 2. Berikut ini adalah contoh tampilan Register dan isi databasenya:



Gambar 3 - Isi key Users dan tampilan Register

Tahapan selanjutnya adalah demo register akun. Ketika user telah mengisi semua komponen inputan yang ada di form Register tersebut dan menekan tombol Buat akun seperti yang terlampir di Gambar 3, maka data akan dikirimkan ke Database sehingga muncul notifikasi berwarna hijau seperti pada yang tertera di Gambar 4 di bawah ini:

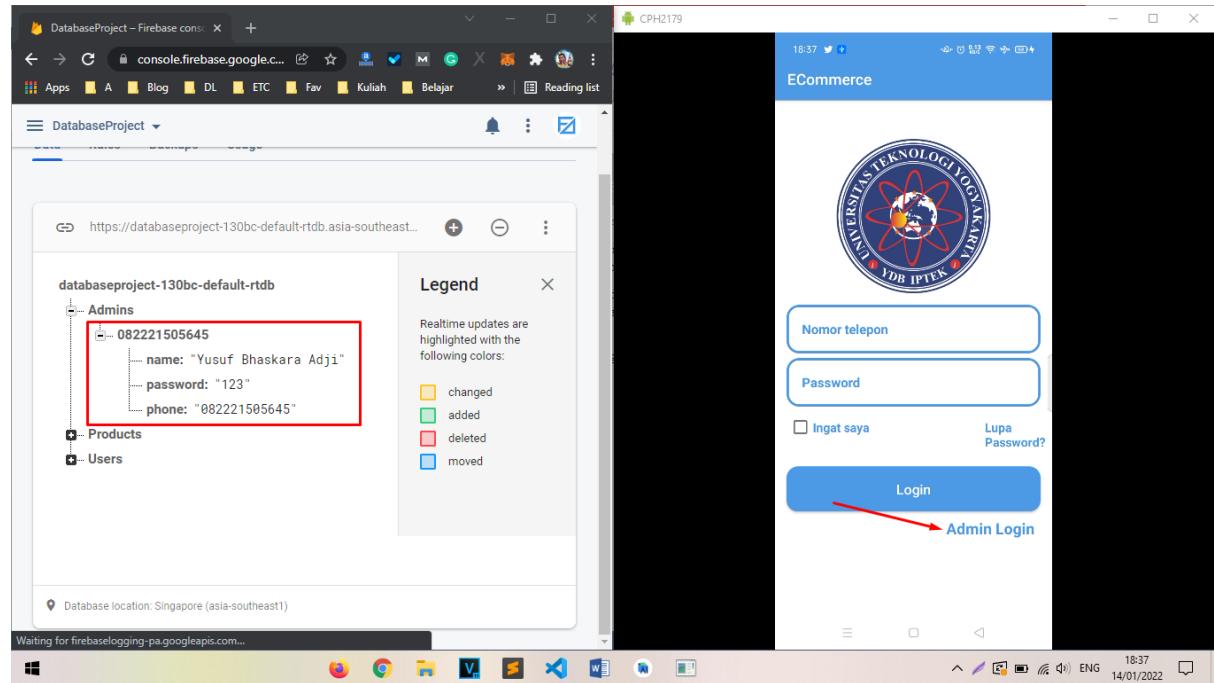


Gambar 4 -Demo akun berhasil dibuat

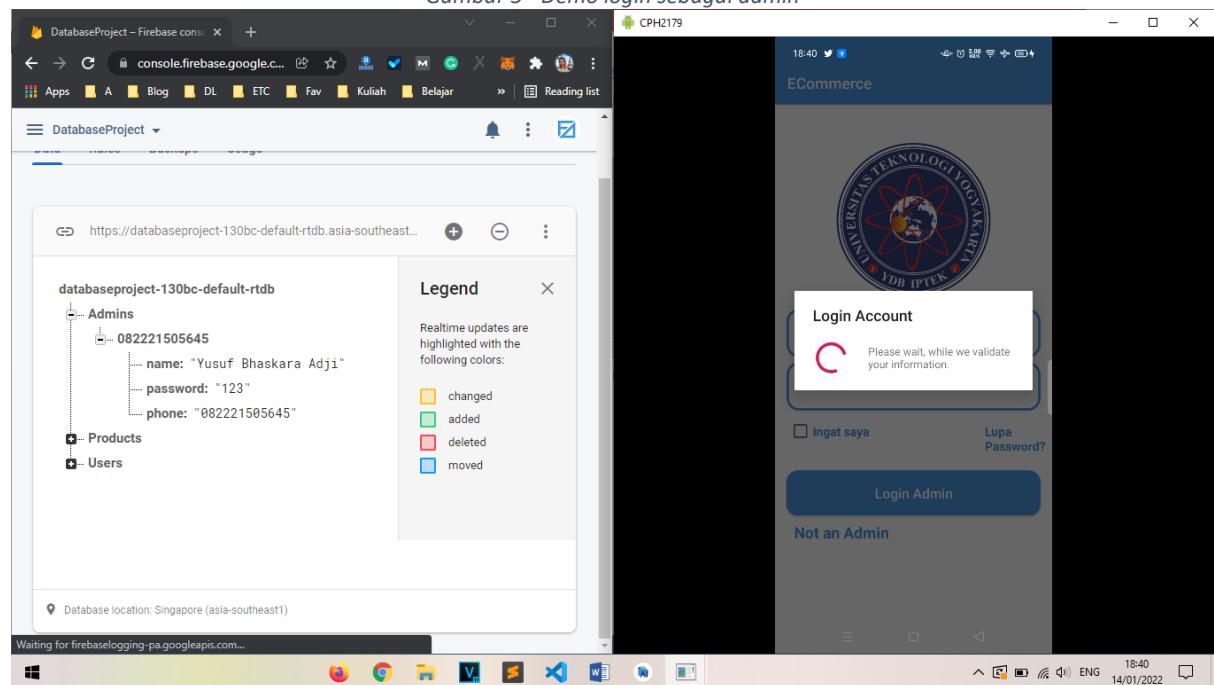
Pada gambar diatas, key yang diberi highlight warna kuning berarti terjadi perubahan dan key yang diberi highlight berwarna hijau berarti terdapat penambahan value baru.

Sebagai catatan, konteks register akun disini hanya berlaku untuk User saja, tidak untuk admin. Kredensial admin dapat ditambahkan secara manual melalui Firebase Realtime Database.

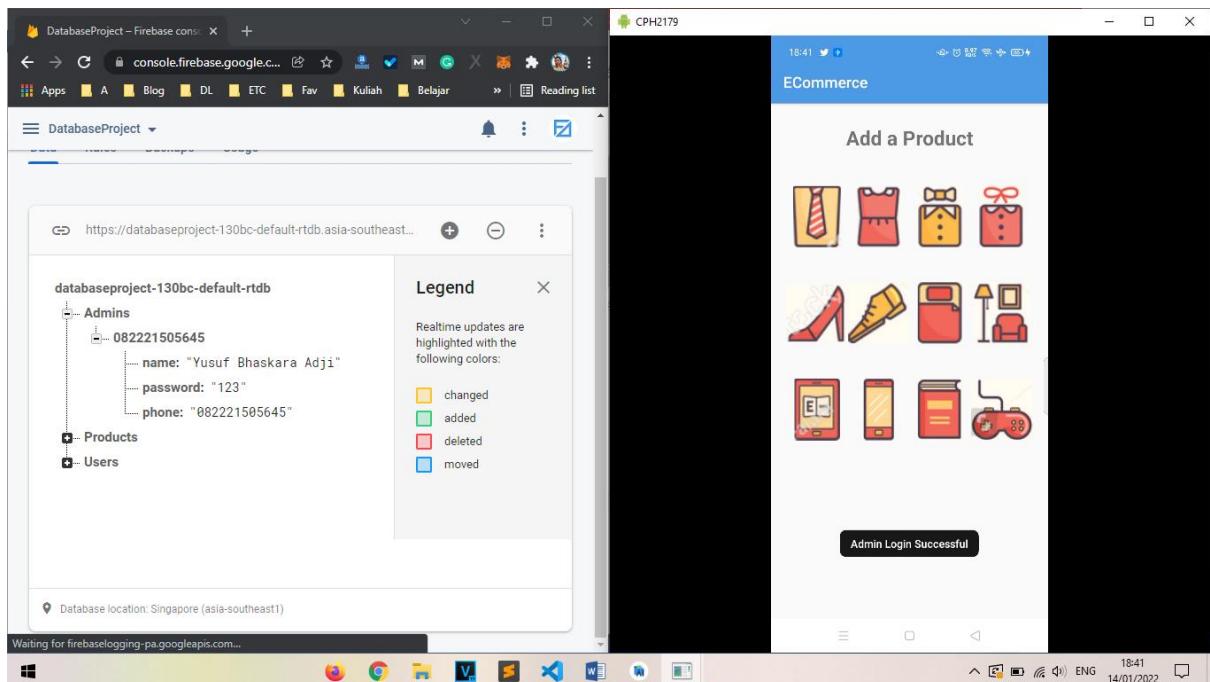
Selanjutnya kita beralih ke activity Login. Pertama kita akan login sebagai admin untuk menambah produk. Di activity login tekan textView Admin Login lalu gunakan kredensial yang terdapat di Firebase.



Gambar 5 - Demo login sebagai admin

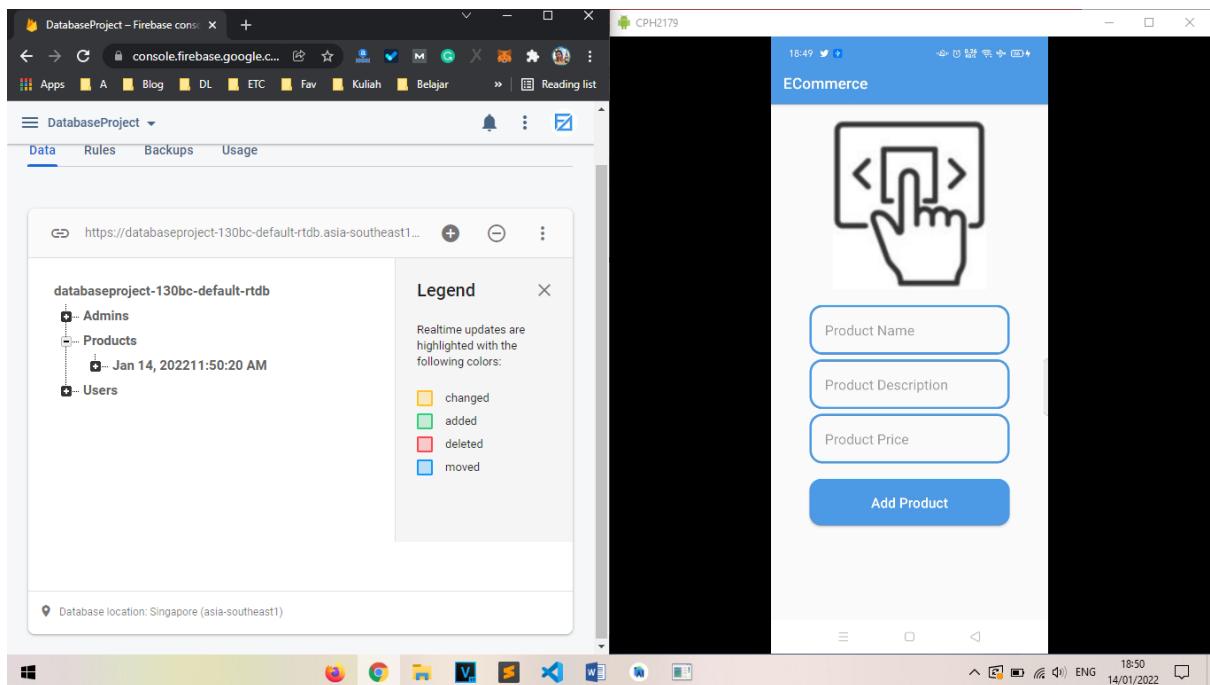


Gambar 6 - Tahap proses cek autentikasi akun



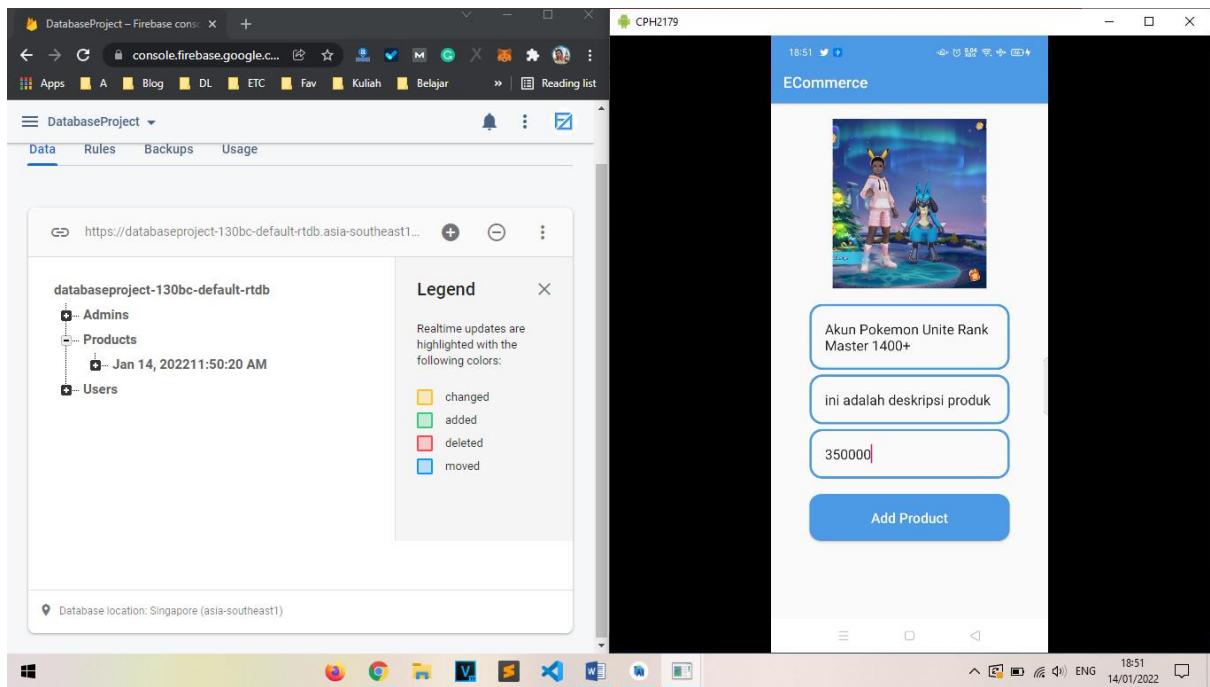
Gambar 7 - Berhasil login ke panel penambahan produk sebagai admin

Selanjutnya pilih salah satu kategori yang ada di panel tersebut, lalu akan diarahkan ke activity tambah produk seperti ini:



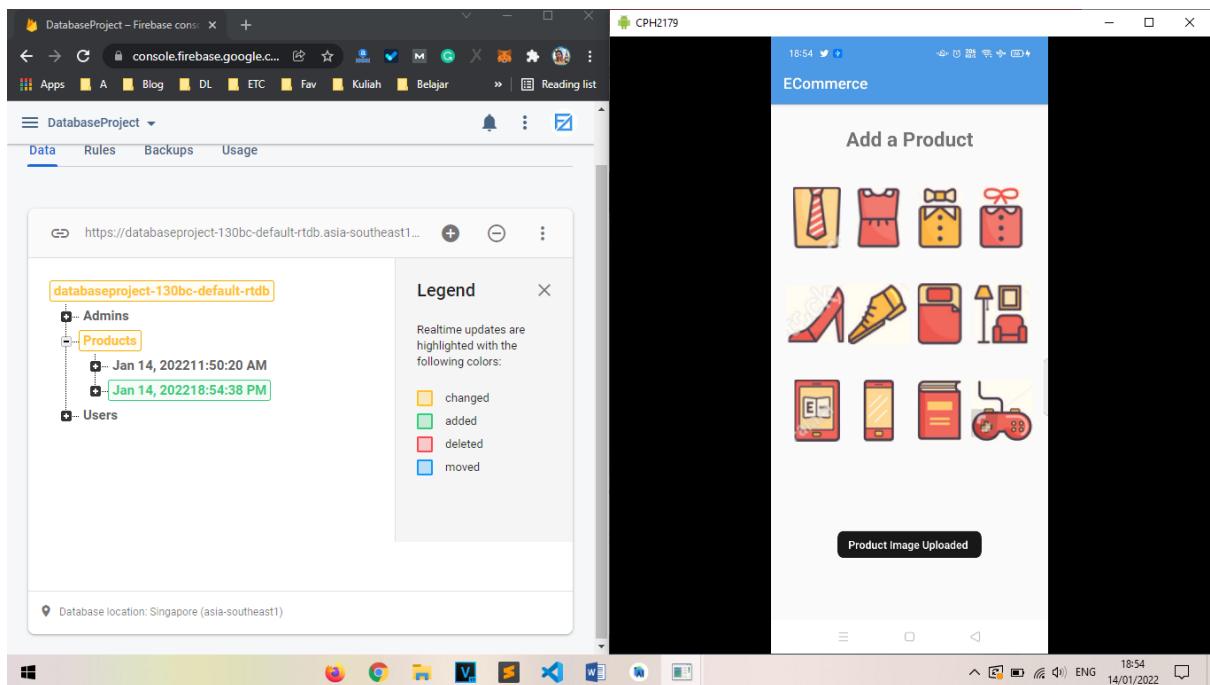
Gambar 8 - Tampilan tambah produk

Pada gambar 8, ketika tombol tap gambar diatas komponen inputan Product Name ditekan, akan muncul opsi untuk memilih gambar.



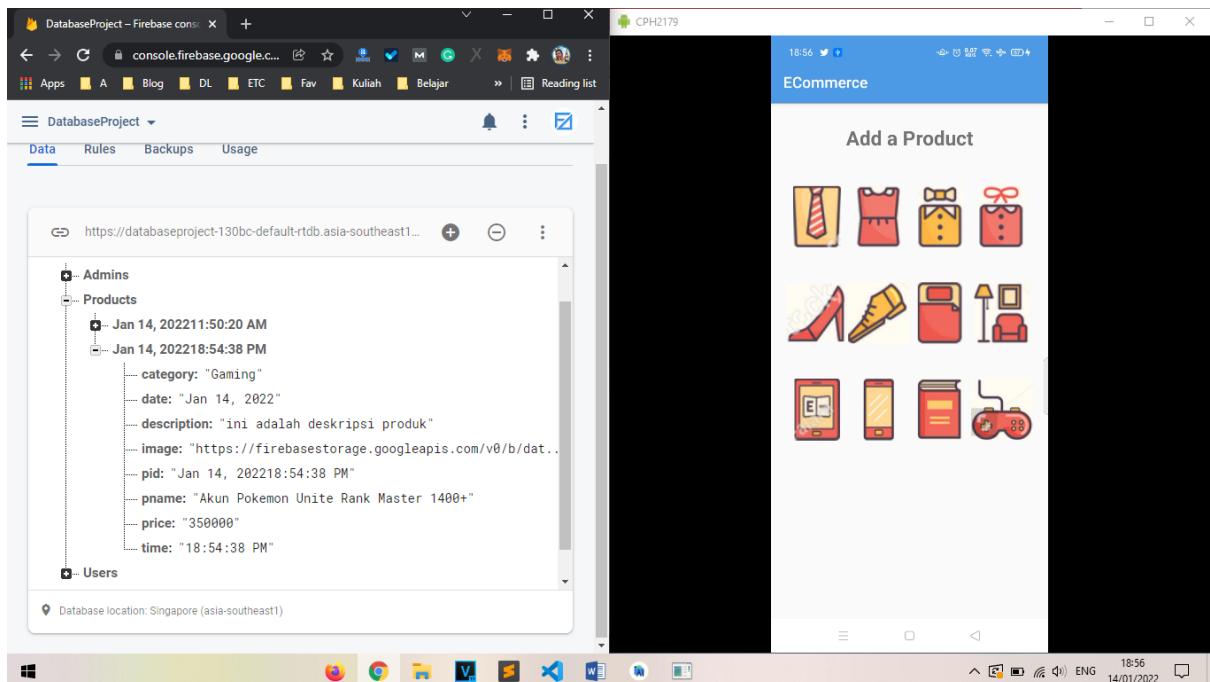
Gambar 9 - Contoh detail produk yang akan ditambahkan

Setelah komponen diisi seperti ini, lanjut tekan tombol Add Product untuk menambah produk ke Firebase Realtime Database. Hasilnya akan seperti berikut:



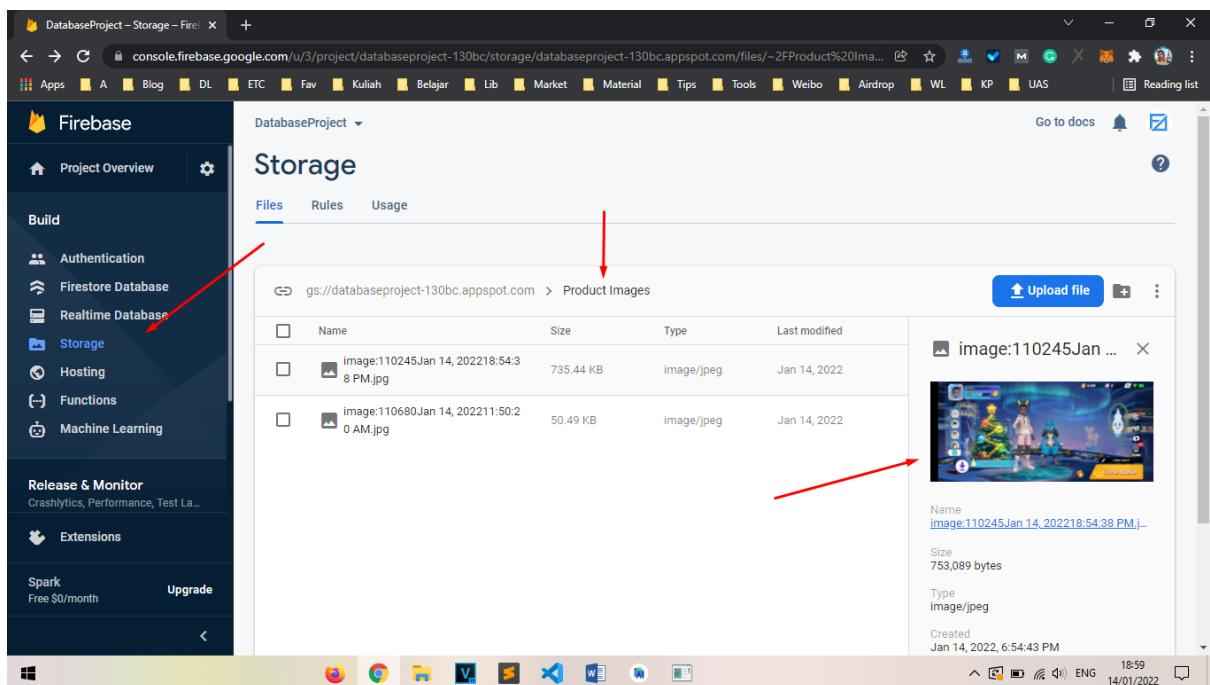
Gambar 10 - Produk berhasil ditambahkan

Catatan : Untuk bisa mengupload foto ke Storage di Firebase, perlu dilakukan penyesuaian rules yang ada. Jika tidak terautentikasi akan muncul error 403 Access Denied dan foto gagal diupload. Bisa diatasi dengan cara simple yaitu mengubah autentikasinya menjadi null, tapi rules ini membuat siapa saja dapat mengunggah foto ke Storage.



Gambar 11 - Tampilan isi key products

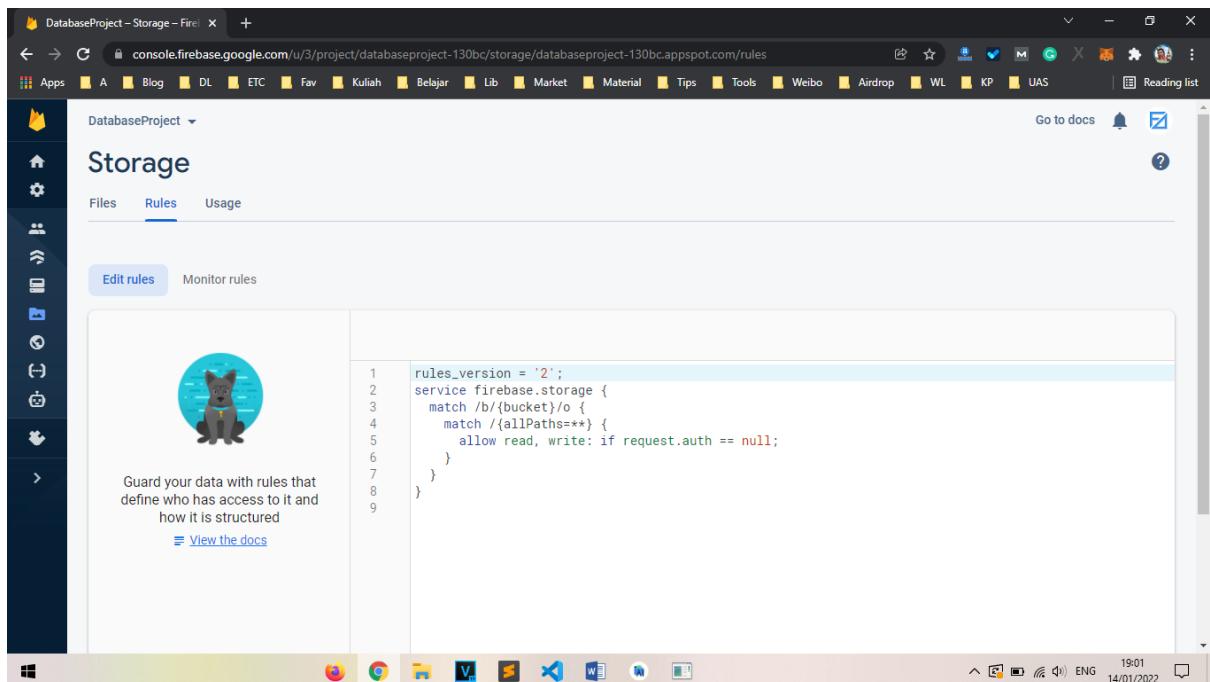
Produk ditambahkan dengan key yang diatur sesuai dengan waktu penambahannya. Khusus untuk kasus upload gambar tadi, harus diintegrasikan dengan fitur Storage dari Firebase. Untuk mengaksesnya masuk ke menu Firebase → Storage seperti ini:



Gambar 12 - Gambar berhasil diupload ke Firebase Storage

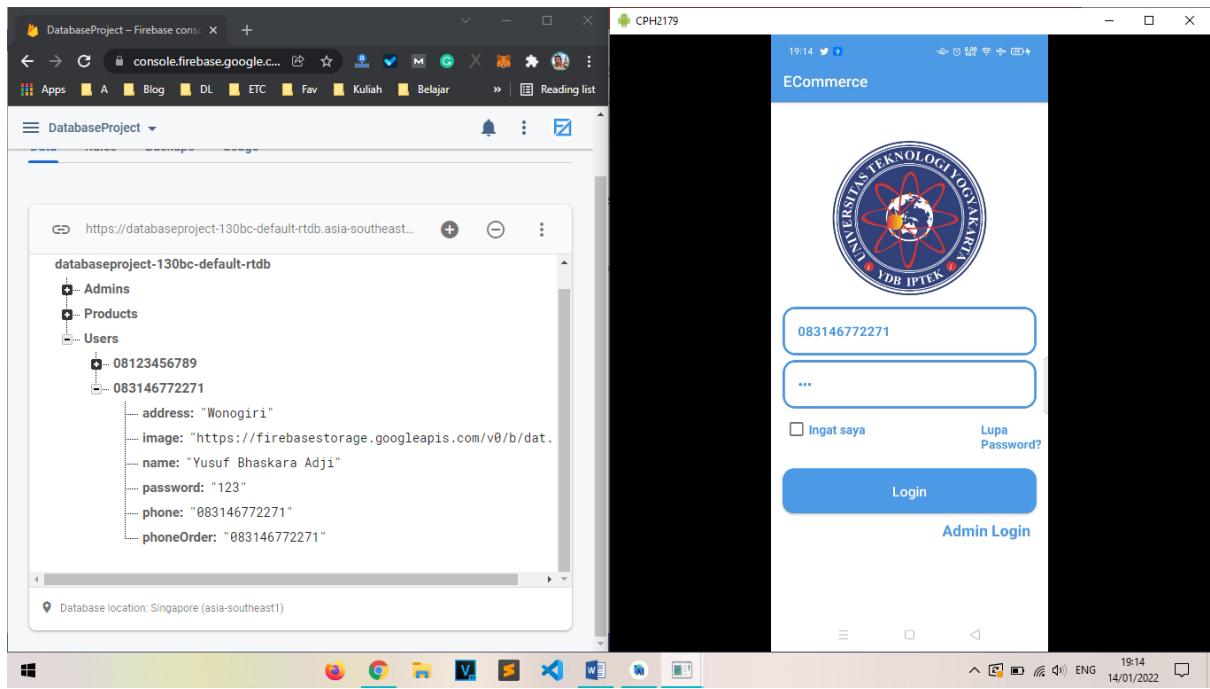
Gambar produk yang sudah ditambahkan akan masuk ke direktori Product images seperti yang tertera di gambar 12.

Catatan : Rules dari Firebase Storage harus diubah dari nilai defaultnya agar dapat mengunggah gambar dari aplikasi ke Storage. Disini saya menggunakan rules seperti berikut:

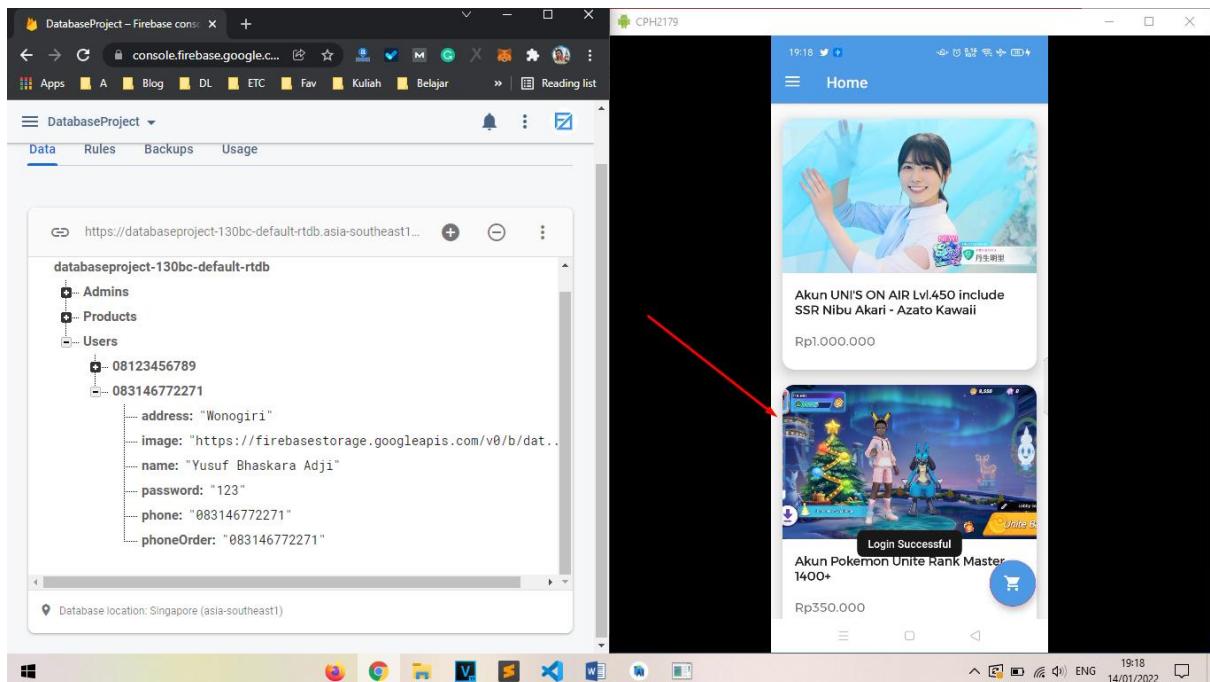


Gambar 13 - Rules Firebase Storage

Selanjutnya kita akan login sebagai user menggunakan kredensial yang sudah dibuat. Demonya seperti berikut:



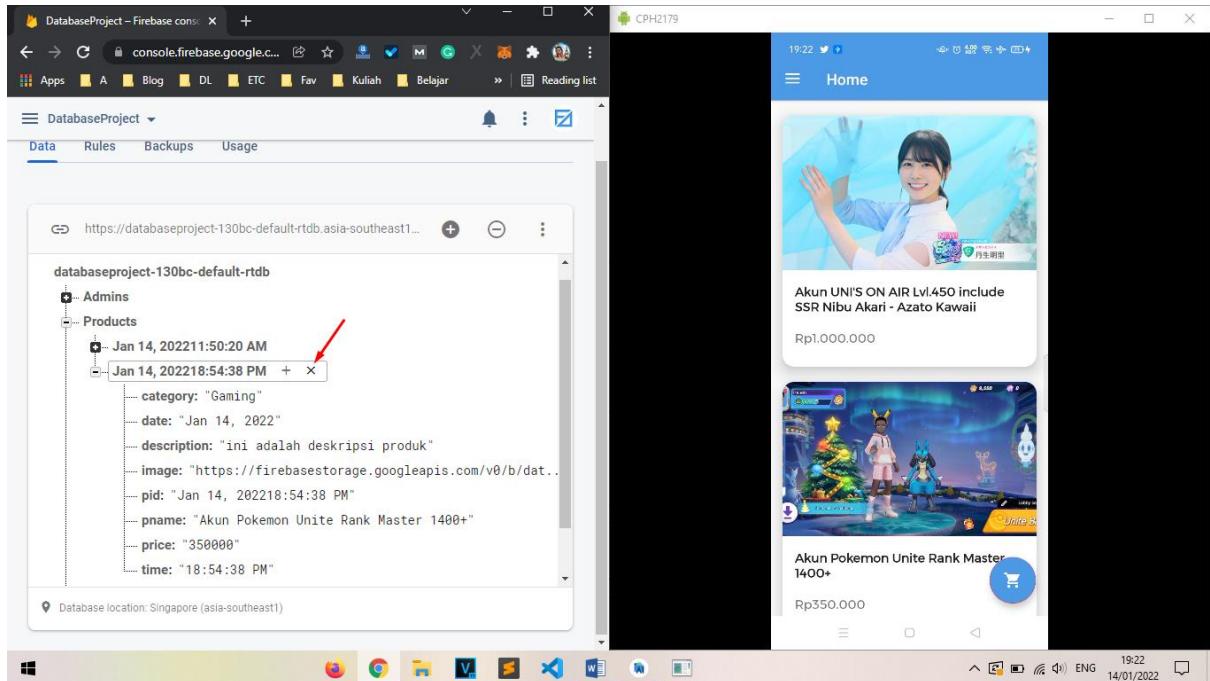
Gambar 14 - Login menggunakan akun yang sudah dibuat



Gambar 15 – Tampilan ketika login berhasil

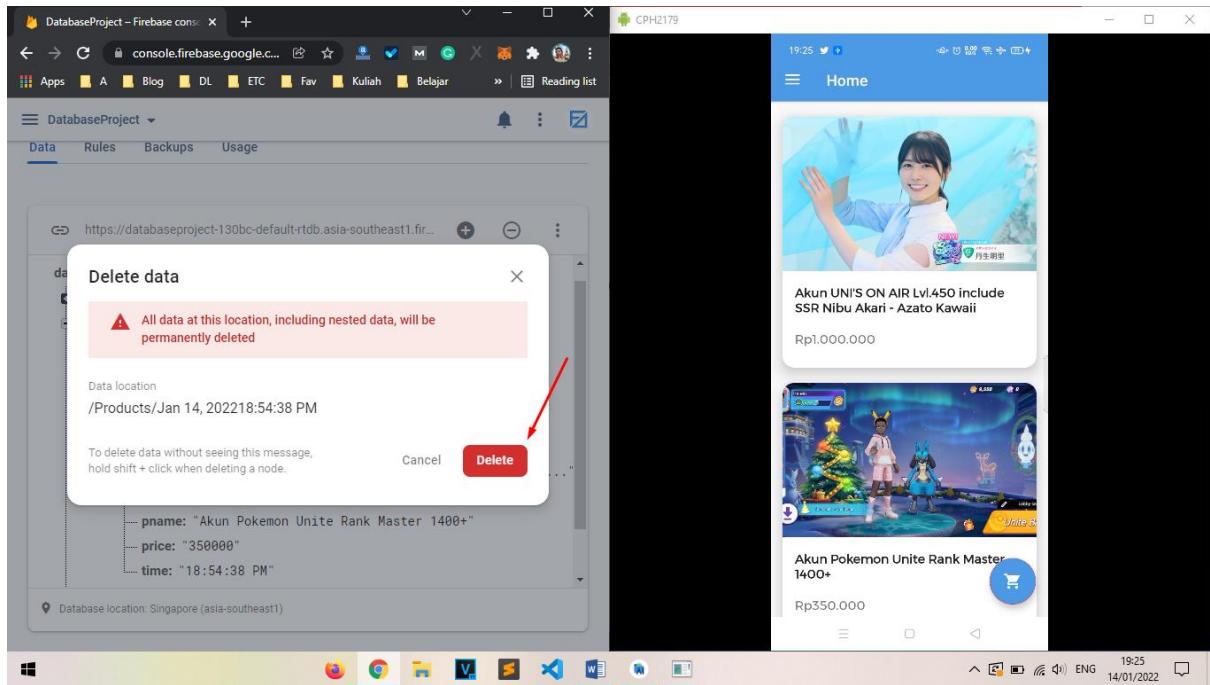
Ketika user berhasil login, daftar produk akan langsung ditampilkan seperti yang tertera pada gambar 15. Produk yang berhasil ditambahkan pada gambar 9 hingga 11 juga langsung ditampilkan di RecyclerView tersebut.

Selanjutnya saya akan melakukan demo hapus produk melalui Firebase Realtime Database. Disini saya akan mencoba menghapus produk yang sudah berhasil ditambahkan pada gambar 9 hingga 11. Pertama hover cursor ke produk seperti ini dan tekan tombol silang:



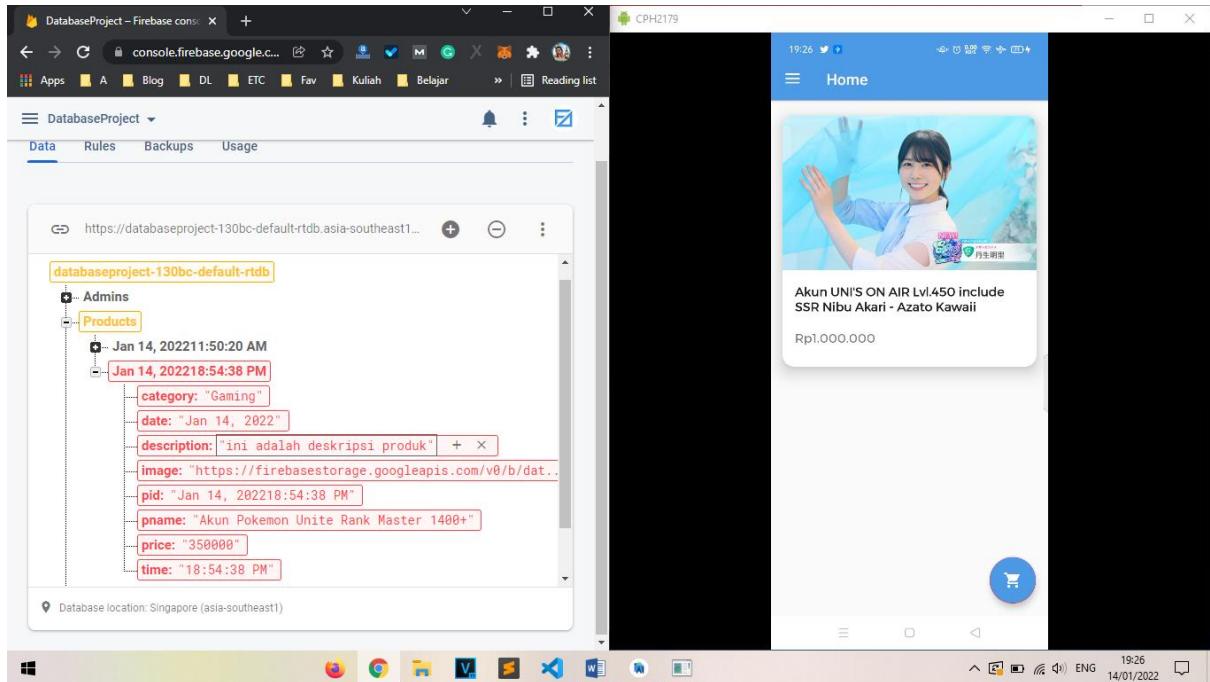
Gambar 16 - Demo menghapus data di Firebase Realtime Database

Akan muncul konfirmasi seperti berikut, lanjutkan tekan tombol Delete.



Gambar 17 - Konfirmasi penghapusan data dari database

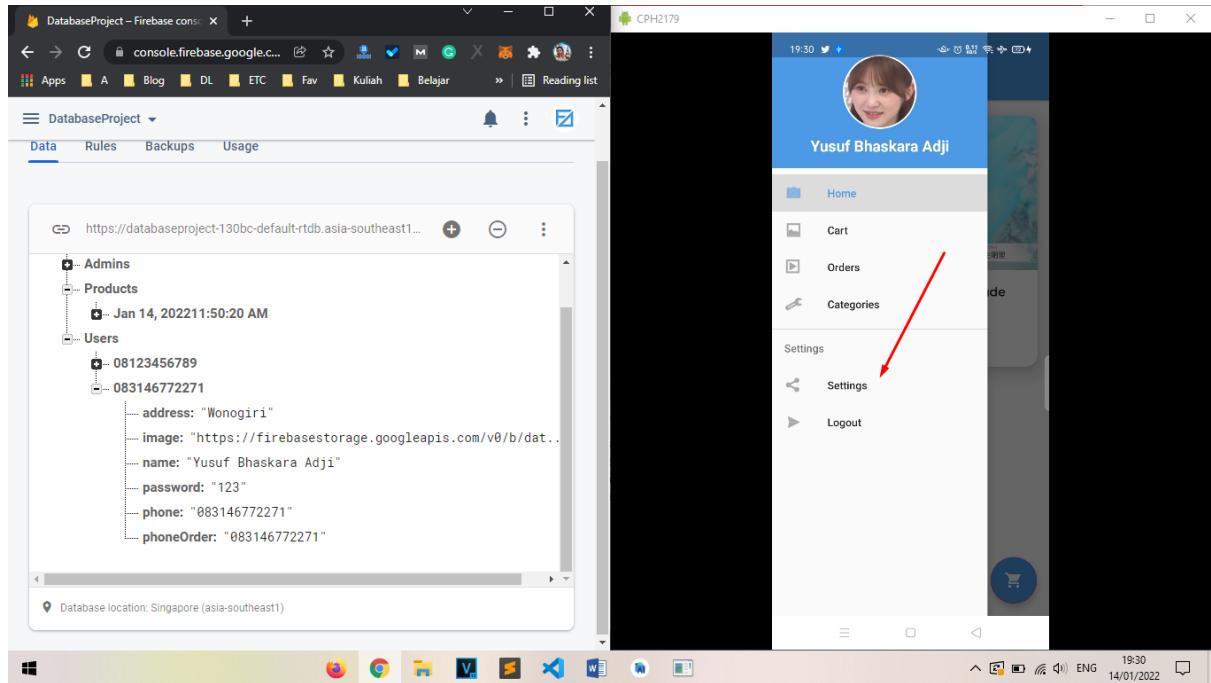
Ketika proses penghapusan selesai, tampilannya akan seperti berikut:



Gambar 18 - Penghapusan data berhasil

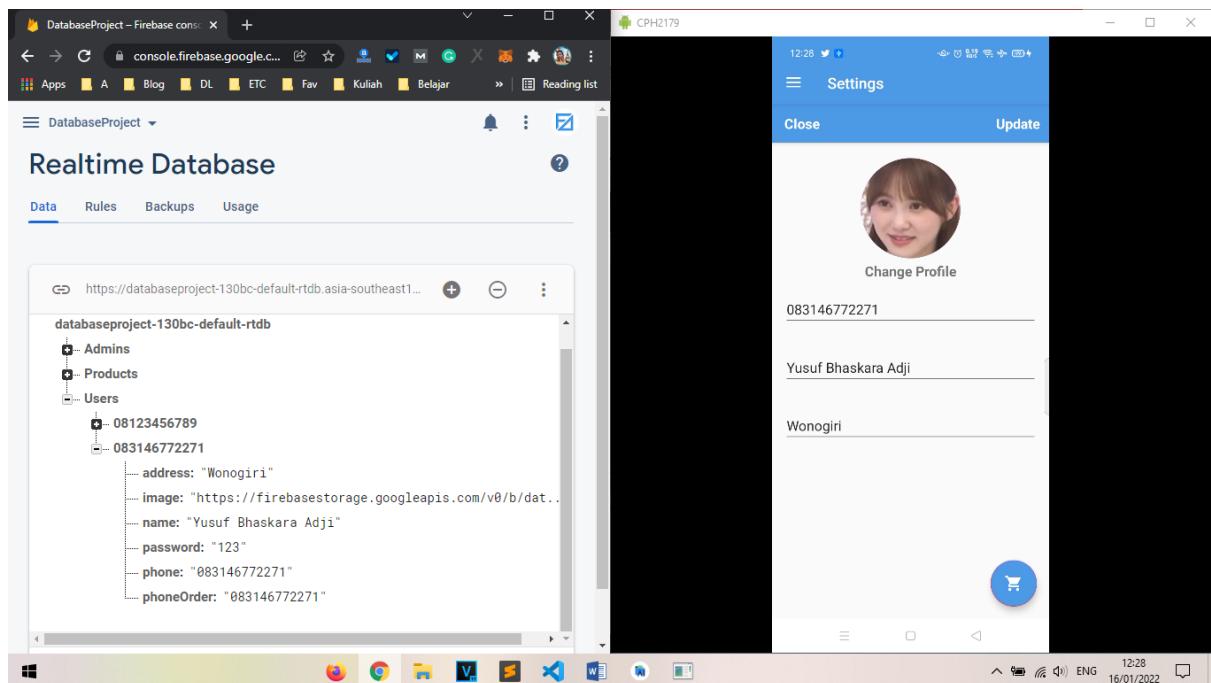
Penghapusan data pada database berhasil. Terlihat pada gambar 18 bahwa highlight berwarna kuning menandakan adanya perubahan data dan highlight merah menandakan adanya penghapusan data. Penghapusan data yang dilakukan langsung berefek ke aplikasi. Produk yang berhasil ditambahkan di gambar 9 hingga 11 hilang dari RecyclerView secara realtime.

Selanjutnya adalah demo update data user. Untuk melakukan update data user, tekan tombol navigasi menu di pojok kiri atas aplikasi yang tertera di gambar 18 untuk memunculkan navigasi barunya lalu pilih menu Setting seperti berikut:



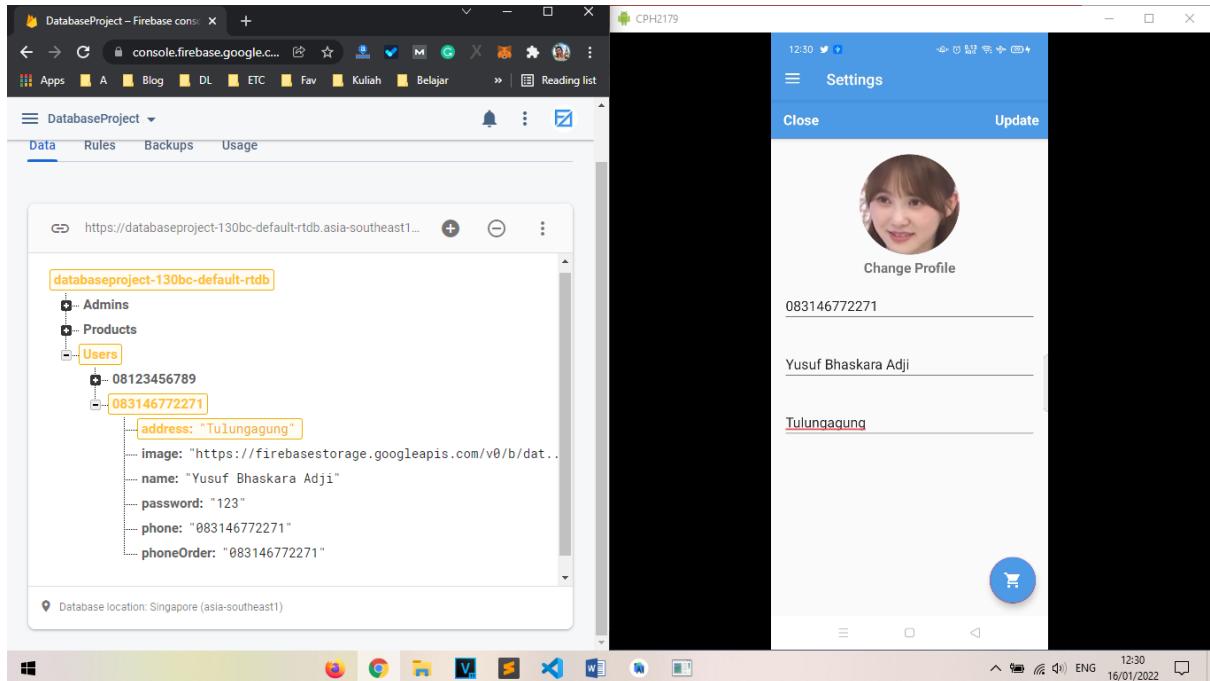
Gambar 19 - Akses ke activity update detail user

Berikut ini adalah tampilan detail usernya:



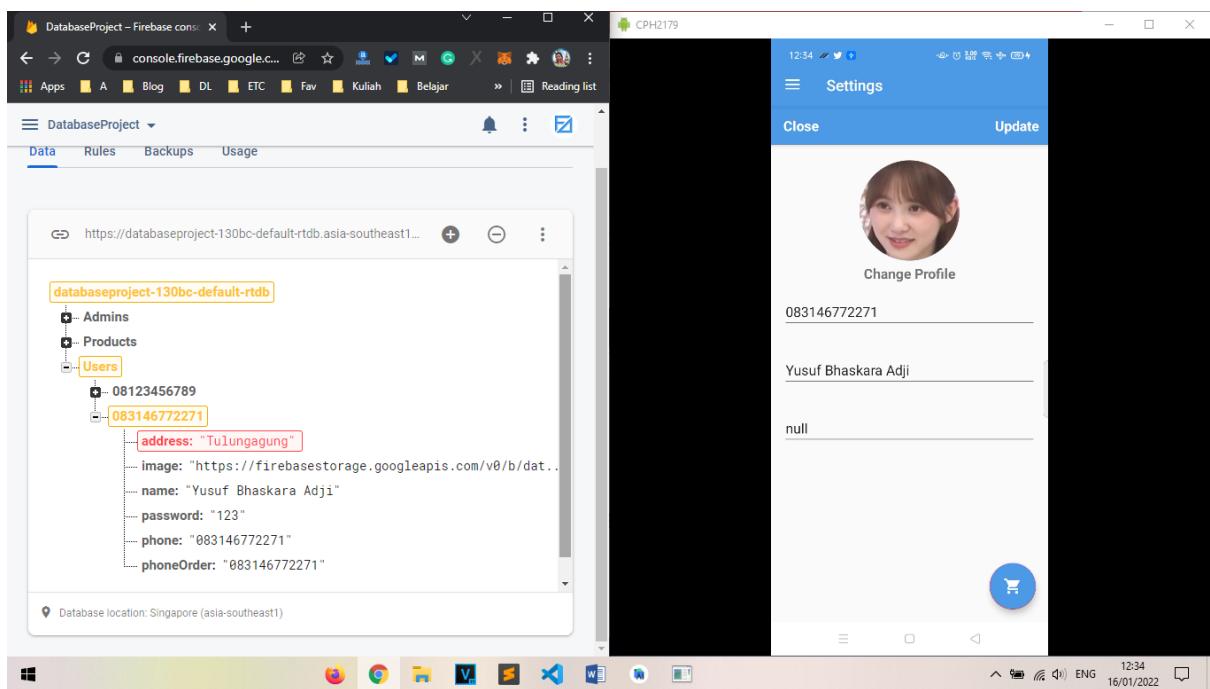
Gambar 20 - Tampilan detail user

Apabila data pada Firebase Realtime Database diubah, maka data di aplikasi ikut berubah secara realtime seperti berikut:



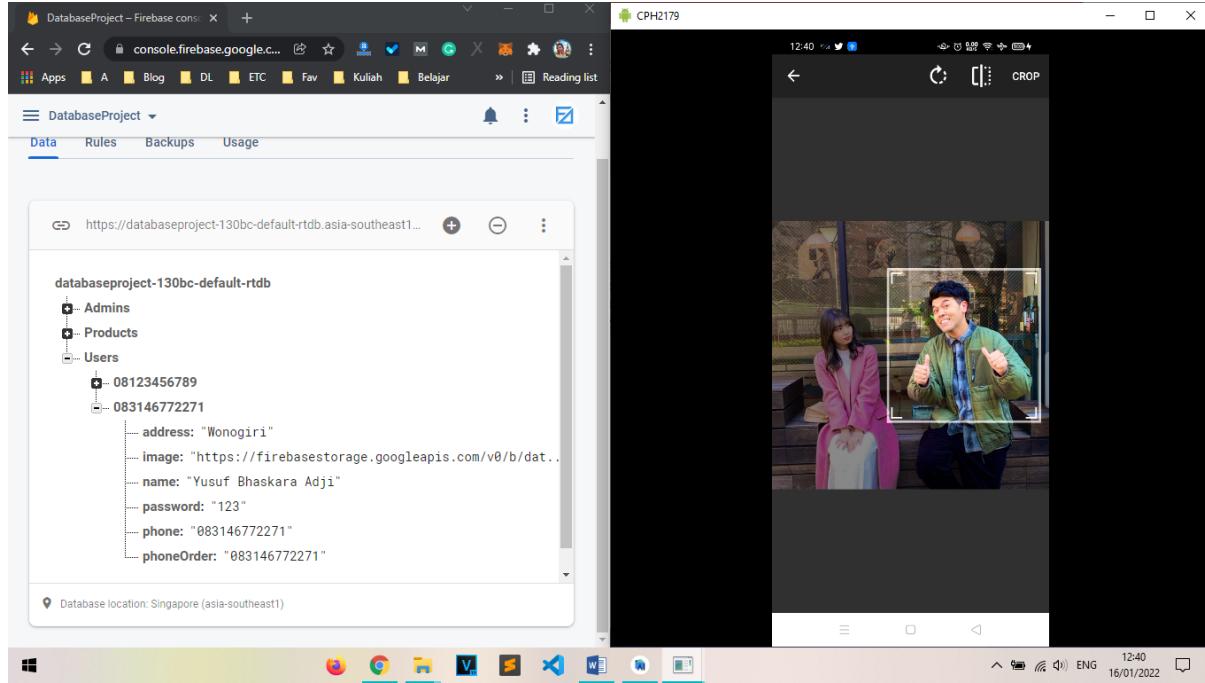
Gambar 21 - value "Wonogiri" pada key Address diubah menjadi "Tulungagung" melalui database, dan data di aplikasi turut berubah

Lalu apabila salah satu key dihapus, disini diambil contoh ke Address lagi, maka hasilnya akan seperti berikut:



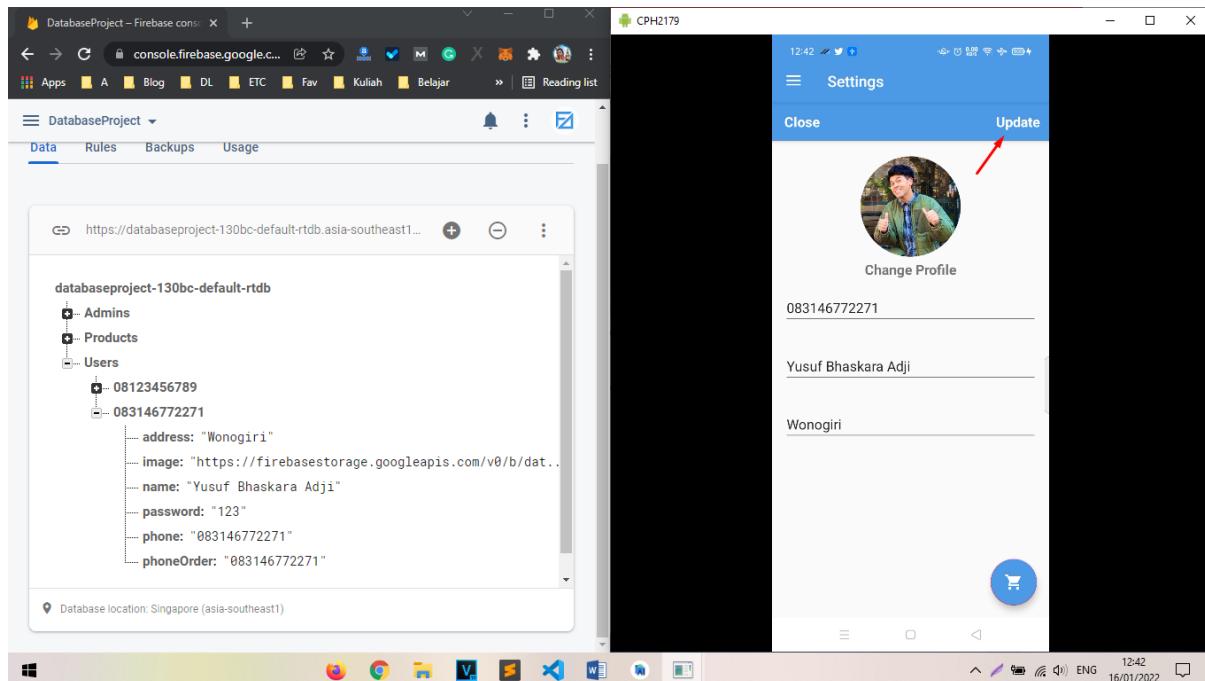
Gambar 22 - Data pada database dihapus membuat aplikasi menampilkan nilai null

Selanjutnya adalah demo mengubah foto profil user. Pada gambar 22, tekan tombol Change Profile untuk memunculkan imagepicker lalu pilih foto yang ingin digunakan. User dapat menyesuaikan foto yang dipilih dengan memangkasnya dengan fitur crop. Fitur ini memanfaatkan library dari repository GitHub [ArthurHub](#). Tampilannya akan seperti berikut:

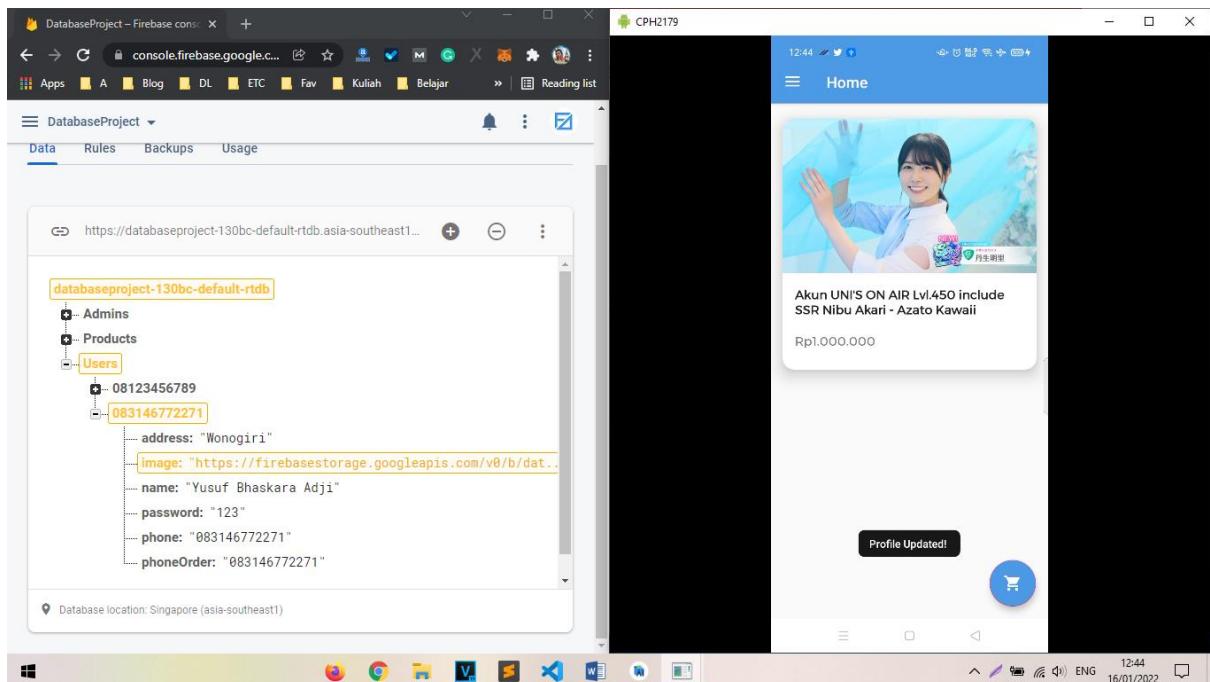


Gambar 23 Fitur crop dari Repository ArthurHub

Jika sudah seperti ini, tekan tombol Update di pojok kanan atas untuk melakukan proses pengunggahan foto ke Storage dan mengubah value dari key image di User menjadi link foto yang sudah diunggah tadi.

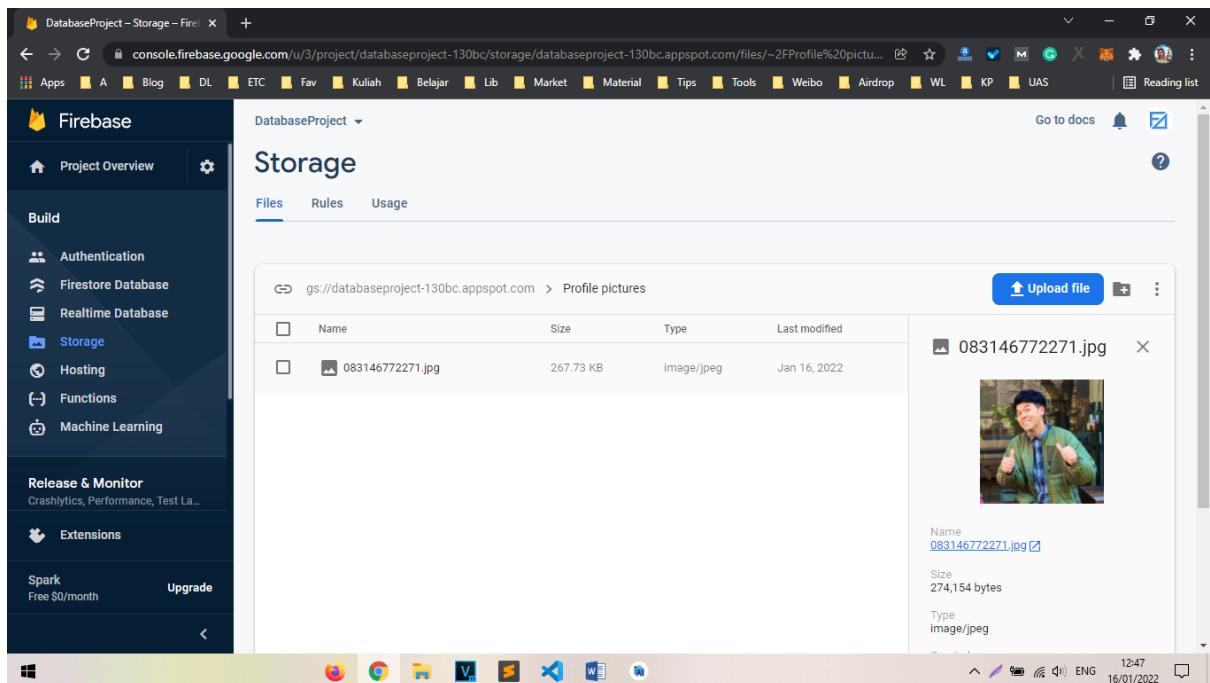


Gambar 24 - Tekan tombol Update untuk mengupload memperbarui data user



Gambar 25 - Profil berhasil diperbarui

Profil user berhasil diperbarui, data di database berhasil diupdate, dan user dikembalikan ke Home. Foto yang disimpan di Storage akan berada di direktori Profile pictures dengan nama file sama dengan id user / nomor telepon user seperti berikut:



Gambar 26 - Foto profil yang berhasil diunggah ke database

## DAFTAR PUSTAKA

<https://github.com/DarrenRegan/Final-Year-Project>