

Pengembangan Frontend dan Backend Aplikasi Deteksi Stunting

Arfiq Rimeldo
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
rimeldoarfiq@student.telkomuniversity
.ac.id

Casi Setianingsih
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
setiacasie@telkomuniversity.ac.id

Ashri Dinimiharwati
Fakultas Teknik Elektro
Telkom University
Bandung, Indonesia
ashridini@telkomuniversity.ac.id

Abstrak—Stunting merupakan masalah serius dalam kesehatan masyarakat Indonesia, khususnya di daerah-daerah yang masih sulit mendapatkan akses terhadap makanan-makanan bergizi, daerah dengan sanitasi yang kurang memadai, dan daerah dengan tingkat kesadaran yang rendah mengenai pentingnya gizi dan kesehatan anak sehingga diperlukan aplikasi yang dapat melakukan *early prediction* terhadap status anak agar orang tua mengetahui anak tersebut normal atau kekurangan nutrisi. Dalam membuat aplikasi tersebut tentu saja diperlukan *frontend* untuk membuat aplikasi tersebut dapat berinteraksi dengan pengguna dan diperlukan juga sistem *backend* untuk menyimpan data menggunakan Firebase dan server menggunakan FastAPI. Dilakukan juga beberapa pengujian untuk menguji apakah aplikasi dapat diterima dan digunakan oleh pengguna nantinya.

Kata Kunci—stunting, aplikasi android, *frontend*, *backend*.

I. PENDAHULUAN

Stunting adalah kondisi kekurangan gizi kronis pada anak yang mengakibatkan pertumbuhan tinggi badan tidak optimal sesuai dengan usianya. Kondisi ini biasanya terjadi selama periode 1.000 hari pertama kehidupan, mulai dari masa kehamilan hingga anak berusia dua tahun. Stunting tidak hanya mempengaruhi pertumbuhan fisik anak, tetapi juga perkembangan kognitif dan kesehatan jangka panjang. Anak-anak yang mengalami stunting berisiko lebih tinggi mengalami masalah kesehatan seperti penurunan daya tahan tubuh, keterlambatan perkembangan mental, serta berkurangnya produktivitas di masa dewasa. Pencegahan stunting memerlukan intervensi gizi yang komprehensif dan tepat waktu, serta perbaikan dalam pola asuh, akses ke layanan kesehatan, dan kondisi sanitasi [1].

Untuk mendeteksi stunting pada balita, pengukuran tinggi atau panjang badan merupakan metode yang umum digunakan. Pengukuran ini dilakukan dengan membandingkan tinggi badan anak sesuai dengan usianya menggunakan standar yang telah ditetapkan oleh Organisasi Kesehatan Dunia (WHO). Anak dinyatakan stunting jika tinggi atau panjang badannya lebih dari dua standar deviasi di bawah median kurva pertumbuhan WHO. Penting untuk melakukan pengukuran secara berkala dan konsisten, serta menggunakan alat ukur yang akurat. Deteksi dini sangat penting karena stunting sering kali tidak tampak secara kasat mata dalam tahap awal, tetapi memiliki dampak yang signifikan terhadap perkembangan anak [2].

Dengan adanya permasalahan ini, aplikasi ini dibuat untuk melakukan deteksi dini terhadap anak, pengukuran yang dilakukan pada aplikasi ini dilakukan secara otomatis tanpa perlu menggunakan alat ukur lagi sehingga dapat

memudahkan pengguna kecuali untuk pengukuran berat badan yang harus ditimbang dan dimasukkan ke dalam aplikasi secara manual. Selain itu, dalam penentuan status balita pada aplikasi ini sudah berdasarkan dengan standar WHO jadi untuk hasil pengukuran bisa dikatakan dapat dipercaya untuk deteksi dini pada anak.

Agar semua hal tersebut dapat berjalan dengan baik diperlukan pembuatan *frontend* aplikasi menggunakan React Native agar aplikasi dapat digunakan oleh pengguna serta diperlukan sistem *backend* dengan menyimpan data pengguna nantinya pada Firebase serta menjalankan server menggunakan *framework* FastAPI. Dengan dibuatnya *frontend* dan *backend* tadi maka diharapkan aplikasi deteksi stunting dapat berjalan dengan baik dan sesuai dengan harapan.

II. KAJIAN TEORI

Dalam pembuatan *frontend* aplikasi deteksi stunting ini digunakan *framework* React Native. React Native adalah *framework open-source* yang dikembangkan oleh Facebook untuk membangun aplikasi mobile lintas platform menggunakan JavaScript dan React. *Framework* ini memungkinkan pengembang untuk menulis satu set kode yang dapat dijalankan pada berbagai platform, seperti iOS dan Android, sehingga mempercepat proses pengembangan dan mengurangi biaya. Dengan menggunakan komponen UI asli dari platform, React Native mampu memberikan performa dan pengalaman pengguna yang hampir setara dengan aplikasi native. Selain itu, React Native juga mendukung penggunaan modul native yang memungkinkan integrasi dengan fitur-fitur khusus dari masing-masing platform. Kemampuannya dalam menggabungkan efisiensi pengembangan dengan pengalaman pengguna yang kaya menjadikannya pilihan yang populer di kalangan pengembang aplikasi mobile modern [3].

Untuk menyimpan *database* pengguna digunakan firebase, firebase yang dimiliki oleh Google merupakan platform pengembangan aplikasi berbasis cloud yang menyediakan berbagai layanan, termasuk layanan database. Firebase menyediakan layanan database NoSQL yang dapat diakses secara real-time, yang dikenal sebagai Firebase Realtime Database. Database ini menggunakan format JSON untuk menyimpan dan menyinkronkan data antara berbagai klien aplikasi secara otomatis [4]. Selain itu, Firebase juga menyediakan layanan database Firestore, yang merupakan basis data NoSQL yang dirancang untuk menangani volume data yang lebih besar dan menyediakan lebih banyak fitur pencarian dan penyaringan [4].

FastAPI adalah *framework* Python yang digunakan untuk membangun API secara cepat dan mudah. Framework ini

sangat cepat dan mendukung *asynchronous programming*, yang berarti aplikasi bisa menangani banyak permintaan secara efisien. FastAPI juga otomatis menghasilkan dokumentasi API yang mudah digunakan, seperti Swagger UI. Setelah itu digunakan metode *tunneling* yaitu layanan yang memungkinkan pengguna untuk mengekspos server lokal mereka ke internet melalui terowongan aman. Pengguna dapat membuat URL publik yang mengarahkan lalu lintas internet ke server local menggunakan Ngrok.

III. METODE

Dalam pembuatan *frontend* dan sistem *backend* dari aplikasi deteksi stunting ini digunakan beberapa metode yang dimulai dari:

A. Pembuatan *Frontend* Aplikasi

Hal pertama yang diperlukan dalam membuat dan mengemabngkan aplikasi deteksi stunting ini adalah React Native CLI, buka terminal atau command prompt kemudian jalankan perintah `npm install -g react-native-cli` untuk menginstal React Native CLI secara global dilanjutkan dengan perintah `npx react-native init NamaProyek` untuk membuat folder yang digunakan untuk pengembangan aplikasi nantinya pada Visual Studio Code. Setelah semua persiapan selesai maka aplikasi sudah dapat dibuat melalui Visual Studio Code dengan React Native.

B. Pembuatan dan Integrasi *Database*

Database yang digunakan untuk menyimpan data pada aplikasi NutriScan ini adalah Firebase. Penggunaan Firebase bisa dibilang cukup mudah dibandingkan dengan *database* lainnya. Firebase digunakan untuk menyimpan data pengguna yang mendaftar pada aplikasi dan data anak yang ditambahkan oleh pengguna pada aplikasi untuk diperiksa statusnya.

C. Pembuatan Sistem API

API yang digunakan dalam pembuatan aplikasi ini adalah FastAPI. FastAPI adalah kerangka kerja web modern berbasis Python yang dirancang untuk membangun API RESTful dengan cepat dan efisien. FastAPI memanfaatkan tipe anotasi Python untuk validasi data otomatis dan dokumentasi API dinamis melalui OpenAPI dan JSON [5]. FastAPI digunakan untuk integrasi antara algoritma *image processing* dengan aplikasi deteksi stunting yang sedang dibuat.

Tunneling Server dan Integrasi pada Aplikasi

Ngrok adalah alat yang memungkinkan untuk mengekspos server lokal ke internet melalui *tunneling*. Ngrok dapat membuat URL publik yang dapat diakses dari mana saja di dunia, yang mengarah ke server lokal. Ini sangat berguna untuk pengembangan aplikasi, terutama untuk menguji layanan API dari perangkat atau lokasi yang berbeda tanpa harus meng-host-nya di server publik.

IV. HASIL DAN PEMBAHASAN

A. Pembuatan *Frontend* Aplikasi

Pembuatan proyek React Native pada cmd dilakukan terlebih dahulu sebelum memulai pengembangan aplikasi deteksi stunting. Setelah semua selesai pembuatan dan pengembangan aplikasi deteksi stunting dapat dilakukan di Visual Studio Code yang dapat dimuali dengan langkah awal yaitu:

a. Mengatur App.js Sebagai Fondasi Utama Aplikasi

App.js adalah *file JavaScript* yang memainkan peranan yang sangat penting dalam pembuatan dan pengembangan aplikasi yang menggunakan *framework JavaScript* seperti React Native ini salah satunya. App.js digunakan sebagai *entry point* aplikasi, pengelola navigasi antar halaman aplikasi, dan untuk inisialisi dan konfigurasi yang diperlukan untuk aplikasi.

b. Pembuatan Halaman Hero

Halaman hero dibuat dengan menampilkan logo dari aplikasi yang dibuat yaitu Nutriscan dilanjutkan dengan pesan-pesan yang dapat menarik perhatian pengguna seperti tagline dan ucapan selamat datang dan yang terakhir adalah tombol "Get Started" yang dapat menavigasikan pengguna ke dalam aplikasi/halaman berikutnya.

c. Pembuatan Halaman Login

Halaman login berisikan logo Nutriscan di bagian paling atas diikuti dengan pengisian *email* dan *password* yang telah dibuat pengguna pada bagian pendaftaran. Untuk mendaftar pengguna dapat menekan tombol "Register Now" pada bagian bawah. Jika pengguna lupa *password* yang mereka gunakan ketika mendaftar pada aplikasi Nutriscan mereka dapat menekan tombol "Forgot Password".

d. Pembuatan Halaman Pendaftaran

Pada halaman pendaftaran pengguna diharuskan mengisi data-data yang diperlukan terlebih dahulu seperti nama, email, password, dan konfirmasi password. Setelah semua data diisi maka pengguna dapat menekan tombol "Regist" untuk mendaftarkan akun dan data-data tersebut akan tersimpan ke dalam database sehingga pengguna tidak dapat menggunakan email yang sama ketika mendaftar lagi. Konfigurasi dari databasenya itu telah disimpan ke dalam bagian app.js tadi.

e. Pembuatan Halaman Lupa Password

Pengguna dapat memasukkan email mereka terlebih dahulu kemudian email untuk melakukan konfirmasi dan reset password akan dikirimkan ke email tersebut menggunakan fitur dari Firebase Authentication. Setelah itu pengguna dapat memasukkan password baru mereka dan mereka dapat masuk kembali ke dalam aplikasi Nutriscan.

f. Pembuatan Halaman Home

Pada bagian kiri atas halaman terdapat tulisan nama pengguna yang mendaftar pada aplikasi, yang mana nantinya tulisan tersebut akan berubah sesuai dengan username yang di daftar pengguna dengan cara mengambil data username tersebut dari database. Pengguna dapat memilih 3 fitur utama dari aplikasi yaitu tambah data anak, periksa anak, dan data anak dimana setiap fitur tersebut saling berhubungan satu sama lain. Untuk bagian "News" masih disajikan dengan gambar biasa untuk saat ini dan belum disambungkan dengan URL resmi dari berita tersebut.

g. Pembuatan Halaman Tambah Data Anak

Pertama pengguna harus mengisi nama dari balita yang akan diperiksa dilanjutkan dengan mengisi tanggal lahir, dalam pengisian tanggal lahir ini akan muncul kalender untuk memudahkan pengguna dalam memilih tanggal lahir. Kalender ini dapat dimunculkan menggunakan library `DateTimePicker` dari React Native. Terakhir, pengguna harus memasukkan jenis kelamin balita dimana dalam memilih jenis kelamin pengguna dapat langsung memilih jenis kelaminnya

laki-laki atau perempuan menggunakan library dropdown-select-list.

Setelah semua data dari balita diisi maka pengguna dapat menekan tombol “Done” yang nantinya akan menyimpan data tersebut dalam database serta data tersebut akan tersimpan dan dapat dilihat pada bagian “Data Anak”.

h. Pembuatan Halaman Periksa Anak

Pada bagian ini juga dapat dilihat daftar anak-anak yang sudah ditambahkan datanya tadi terdapat nama dan umur serta gambar disebelah kanan yang menyesuaikan dengan jenis kelamin balita yang didaftarkan. Pengguna diminta untuk memilih balita mana yang ingin diperiksa terlebih dahulu dengan menekan kotak sesuai dengan yang mereka mau.

i. Pembuatan Halaman Periksa Anak2

Terdapat 4 data yang harus dilengkapi oleh pengguna untuk menentukan apakah balita tersebut terkena stunting atau tidak. Pertama tinggi badan, ketika pengguna menekan tombol tinggi badan maka aplikasi akan meminta izin untuk mengakses kamera HP terlebih dahulu, setelah diizinkan maka akan muncul kamera untuk memotret balita yang nantinya pada kamera tersebut akan muncul tinggi badan pengguna yang sudah disambungkan dengan algoritma *image processing* tadi jadi tidak perlu diukur secara manual. Sama halnya dengan bagian pengukuran lingkaran kepala dan lingkaran lengan.

Kamera ini dapat dimunculkan menggunakan *library* 'react-native-image-picker' dari React Native yang kemudian dikonfigurasi ke tombol tinggi badan, lingkaran lengan, dan lingkaran kepala. Setelah mengambil gambar balita, maka gambar tersebut akan diproses dan akan dimunculkan ukurannya berdasarkan perhitungan algoritma yang sudah ditetapkan.

Pengukuran terakhir yang diperlukan yaitu berat badan yang hanya dapat dihitung secara manual kemudian datanya dimasukkan ke dalam kotak berat badan tersebut. Setelah semua data terisi maka pengguna dapat menekan tombol “Next” yang dapat memperlihatkan apakah balita tersebut stunting atau tidak.

j. Pembuatan Halaman Data Anak

Terdapat data-data dari balita yang sudah ditambahkan mulai dari nama, tanggal lahir, jenis kelamin, dan umur. Data-data tersebut diambil dari database kemudian ditampilkan dalam bentuk kolom. Dalam perhitungan umur dari balita digunakan library moment dengan cara menghitung selisih antara tanggal lahir dari balita tersebut dengan tanggal saat ini.

B. Pembuatan dan Integrasi Database

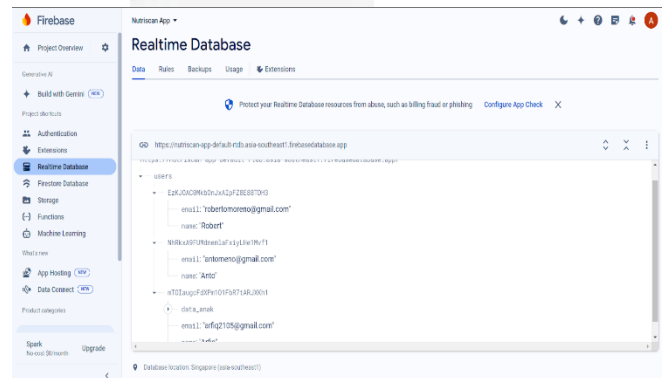
Langkah pertama yaitu dengan membuka website Firebase lalu pilih untuk membuat proyek baru dilanjutkan dengan memilih ingin menyambungkan databasenya dengan menggunakan aplikasi android. Setelah semua sudah dipilih maka akan diberikan konfigurasi databasenya yang akan dimasukkan ke dalam Visual Studio Code untuk menyambungkannya dengan aplikasi.

```
const App = () => {
  useEffect(() => {
    if (!firebase.apps.length) {
      firebase.initializeApp({
        apiKey: "AIza yAKLScdh8x5yBdCCm dpQTCeCJiUK9R igQ",
        authDomain: "nutriscan-app.firebaseio.com",
        projectId: "nutriscan-app",
        messagingSenderId: "238088799437",
        appId: "1:238088799437:android:2a343467da823d8d0bd584",
        storageBucket: "nutriscan-app.appspot.com",
        databaseURL:
          "https://nutriscan-app-default-rtdb.asia-southeast1.firebaseio.com/app/"
      });
    }
  }, []);
};
```

Gambar 1 Codingan Integrasi Firebase

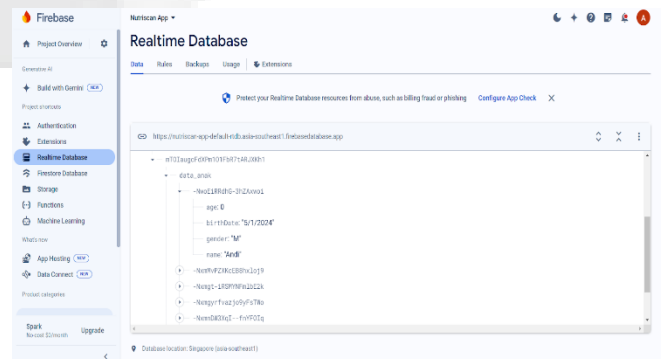
Kodingan diatas merupakan konfigurasi yang digunakan untuk menyambungkan Firebase dengan aplikasi, mulai dari apiKey, authDomain sampai databaseURL . Konfigurasi ini dapat ditemukan pada proyek Firebase yang sudah dibuat di *website* tadi. Sebelum konfigurasi ini dimasukkan haruslah mengimpor Firebase tersebut terlebih dahulu dengan menggunakan kodingan import firebase from '@react-native-firebase/app'.

Setelah semua konfigurasi selesai maka Firebase sudah dapat disambungkan dengan aplikasi dan siap untuk digunakan untuk berbagai keperluan seperti proses autentikasi, penyimpanan data, dll.



Gambar 2 Tampilan Database

Gambar di atas memperlihatkan apa saja yang dimasukkan ke dalam database ini dimulai dari *email* dan *username* yang didaftarkan pengguna serta data anak yang ditambahkan pengguna ketika masuk ke dalam aplikasi.

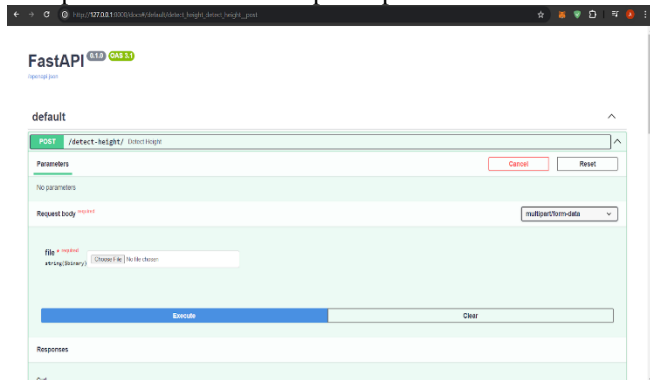


Gambar 3 Tampilan Database2

Gambar di atas menampilkan data anak yang disimpan ke dalam *database* antara lain umur, tanggal lahir, jenis kelamin, dan nama. Jadi semua data yang dimasukkan pengguna ketika menggunakan aplikasi sudah aman dalam *database*.

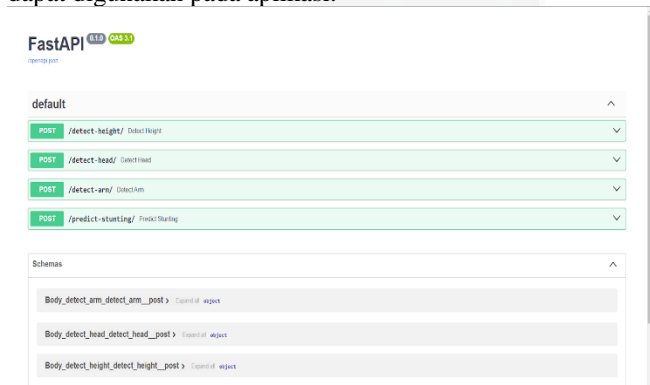
C. Pembuatan Sistem API

Penggunaan FastAPI pada aplikasi ini adalah untuk menyambungkan machine learning dengan aplikasi Nutriscan. Contohnya untuk menyambungkan algoritma pengukuran tinggi badan dengan mengirimkan foto ke server FastAPI kemudian foto tersebut diproses di sana kemudian hasil pemrosesan akan menampilkan tinggi dari orang yang terdapat dalam foto tersebut pada aplikasi.



Gambar 4 Tampilan FastAPI Deteksi Tinggi

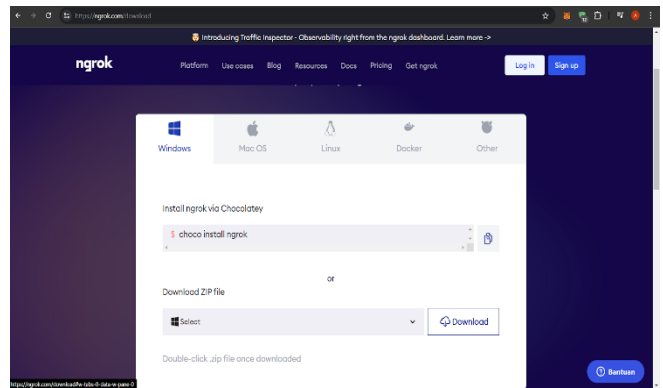
Selain itu semua *endpoint* yang diperlukan untuk menyambungkan algoritma *machine learning* dengan aplikasi Nutriscan dimasukkan ke dalam FastAPI supaya dapat digunakan pada aplikasi.



Gambar 5 Tampilan Keseluruhan FastAPI

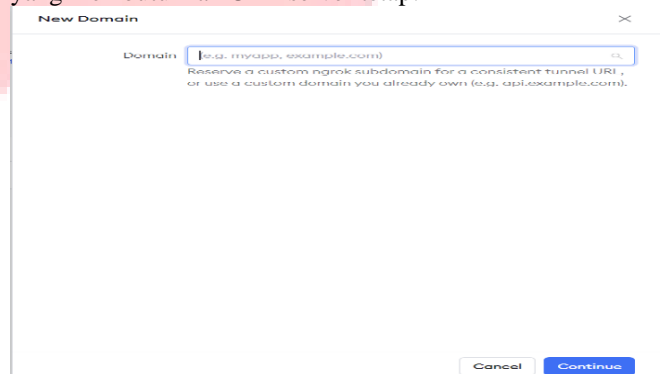
Dari gambar di atas dapat dilihat keseluruhan *endpoint* dari FastAPI yang digunakan, mulai dari pengukuran tinggi, pengukuran lingkaran kepala, pengukuran lingkaran lengan, dan penentuan status balita apakah stunting, normal, atau yang lainnya.

Setelah FastAPI berhasil diselesaikan semua dan sudah bisa digunakan, langkah berikutnya yang dilakukan adalah melakukan *tunneling* server agar server yang sebelumnya lokal menjadi online dan dapat diakses dimanapun dan kapanpunselama masih ada koneksi internetnya menggunakan ngrok. Langkah pertama yang dilakukan adalah dengan mengunduh Ngrok pada website resminya.



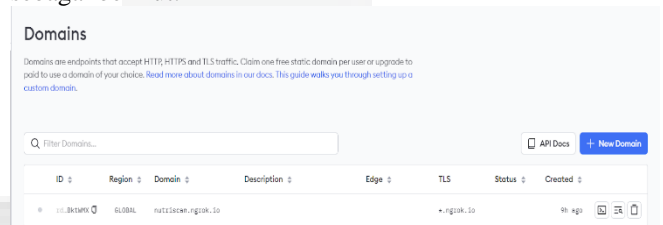
Gambar 6 Mengunduh Ngrok

Selanjutnya adalah membuat domain khusus pada ngrok untuk mendapatkan URL tetap yang tidak berubah setiap kali menjalankan ngrok. Hal ini sangat berguna untuk pengembangan dan pengujian, serta untuk distribusi aplikasi yang membutuhkan URL server tetap.



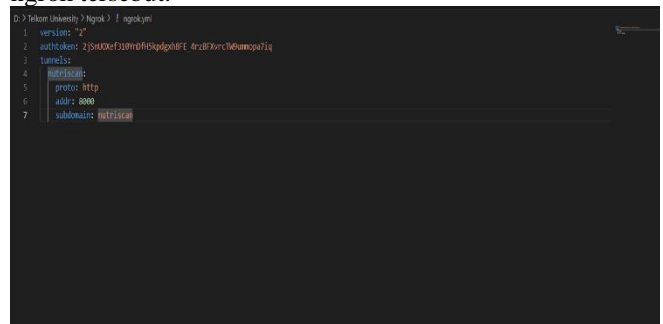
Gambar 7 Membuat Domain Khusus

Tampilan pada ngrok setelah membuat domain khusus adalah sebagai berikut:



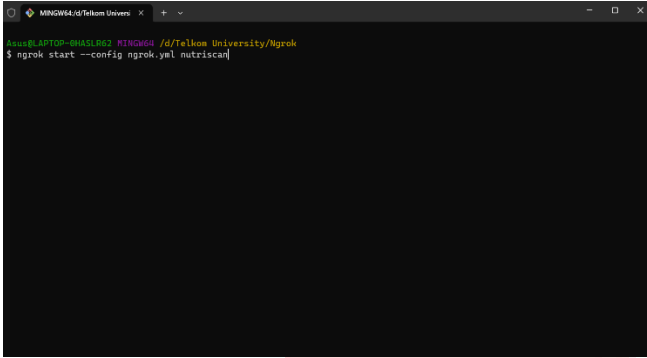
Gambar 8 Tampilan Domain Khusus Telah Berhasil Dibuat

Berikutnya yang perlu dilakukan adalah membuat file *ngrok.yml* yang dapat menyimpan semua pengaturan ngrok dalam satu file, Setelah semua pengaturan disimpan dalam file tersebut dilanjutkan dengan dengan menjalankan server *ngrok* tersebut.



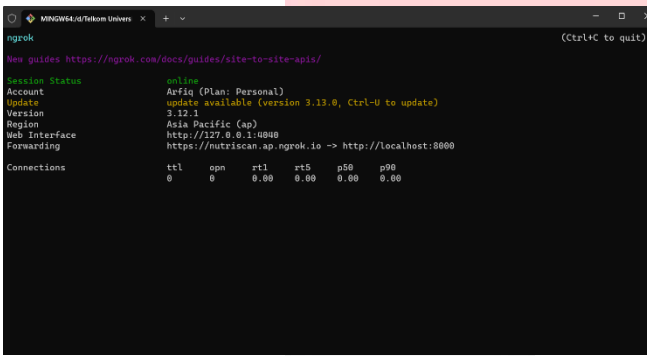
Gambar 9 Tampilan File ngrok.yml

Setelah file ngrok berhasil dibuat maka dapat dilanjutkan dengan menjalankan server ngrok.



Gambar 10 Menjalankan Server Ngrok

Setelah server dijalankan maka akan muncul tampilan seperti ini:



Gambar 11 Tampilan Ngrok Berhasil Dijalankan

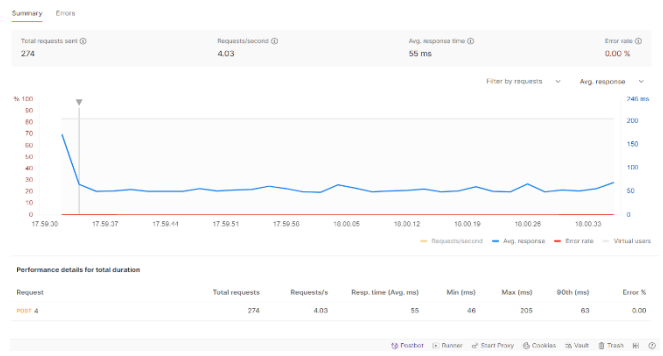
Gambar di atas memperlihatkan adanya URL publik yang telah dibuat oleh Ngrok yaitu “https://nutriscan.ap.ngrok.io”, sehingga server yang sebelumnya hanya berjalan secara lokal sekarang dapat diakses secara publik dengan cara memasukkan URL tadi ke dalam kodingan aplikasi Nutriscan.

```
try {
    const response = await axios.post('https://nutriscan.ap.ngrok.io/detect-height/', formData, {
        headers: {
            'Content-Type': 'multipart/form-data',
        },
        responseType: 'json',
    });
}
```

Gambar 12 Codingan Integrasi Ngrok dengan Aplikasi

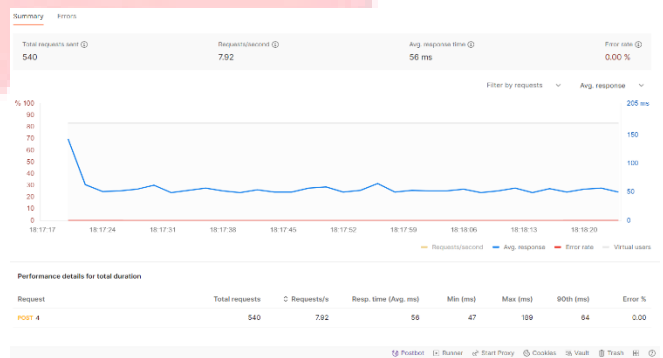
Contohnya pada kodingan di atas yaitu pada bagian pengukuran tinggi, setelah URL dimasukkan ke dalam bagian *response* maka setelah mengambil foto anak maka foto tersebut akan diproses di server dan akan dikembalikan ke aplikasi dengan menampilkan tingginya setelah dihitung menggunakan algoritma *image processing* pada server.

Setelah server berhasil diintegrasikan dengan aplikasi, dapat dilanjutkan ke langkah berikutnya untuk memeriksa dan melakukan pengujian terhadap server. Dalam pengujian server digunakan *tools* tambahan yaitu Postman. Postman adalah alat pengujian beban dan performa server yang bertujuan untuk melihat kinerja dan ketahanan dari server tersebut.



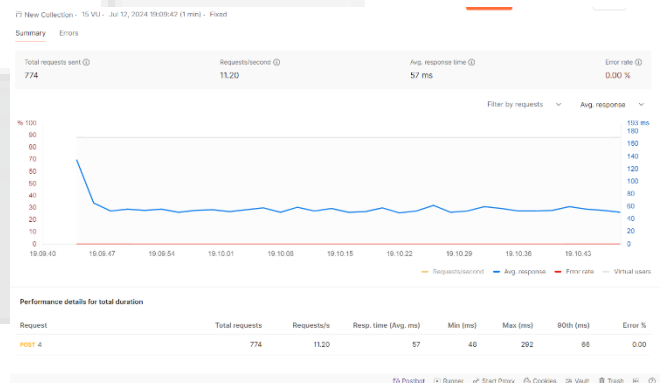
Gambar 13 Grafik Pengujian Server Pertama

Grafik di atas menunjukkan hasil pengujian beban pada server prediksi stunting yang diuji dengan 5 *Virtual Users*. Pengujian dilakukan selama 1 menit dan didapatkan *requests* sebanyak 274 kali. Tidak ada *error* dalam percobaan tersebut dengan rata-rata waktu respons selama 55 ms Dengan waktu respons terlama selama 205 ms dan tercepat selama 46 ms.



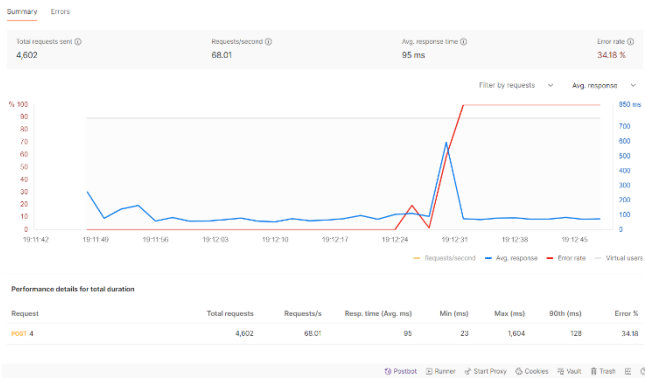
Gambar 14 Grafik Pengujian Server Kedua

Grafik di atas menunjukkan hasil pengujian beban pada server prediksi stunting yang diuji dengan 10 *Virtual Users*. Pengujian dilakukan selama 1 menit dan didapatkan *requests* sebanyak 540 kali. Tidak ada *error* dalam percobaan tersebut dengan rata-rata waktu respons selama 56 ms Dengan waktu respons terlama selama 189 ms dan tercepat selama 47 ms.



Gambar 15 Grafik Pengujian Server Ketiga

Grafik di atas menunjukkan hasil pengujian beban pada server prediksi stunting yang diuji dengan 15 *Virtual Users*. Pengujian dilakukan selama 1 menit dan didapatkan *requests* sebanyak 774 kali. Tidak ada *error* dalam percobaan tersebut dengan rata-rata waktu respons selama 57 ms Dengan waktu respons terlama selama 292 ms dan tercepat selama 48 ms.



Gambar 16 Grafik Pengujian Server Keempat

Grafik di atas menunjukkan hasil pengujian beban pada server prediksi stunting yang diuji langsung dengan 100 *Virtual Users*. Pengujian dilakukan selama 1 menit dan didapatkan *requests* sebanyak 4602 kali. Terjadi *error* dengan persentasae sebesar 34.18% dengan rata-rata waktu respons selama 95 ms Dengan waktu respons terlama selama 1604 ms dan tercepat selama 23 ms. *Error* terjadi ketika memasuki respons ke 1500, hal ini dikarenakan server Ngrok hanya mampu memproses *request* sebanyak lebih kurang 1400 dalam waktu 1 menit. Untuk lebih lengkap dapat dilihat pada tabel berikut:

Tabel 1 Hasil Pengujian Server

No	VU	Requ est	Waktu Respon s Minim um	Waktu Rspns Maksimu m	Rata-Rata Waktu Respons	Persent ase Error
1.	5	274	46 ms	205 ms	205 ms	0%
2.	10	540	47 ms	189 ms	189 ms	0%
3.	15	774	48 ms	292 ms	292 ms	0%
4.	100	4602	23 ms	1604 ms	1604 ms	34,18%

D. Pengujian Aplikasi

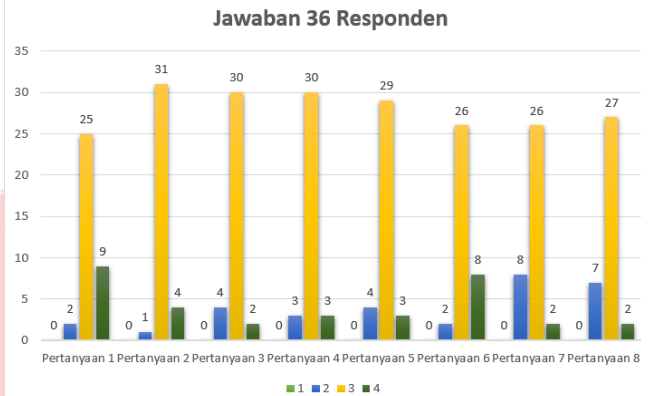
Dalam pengujian aplikasi deteksi stunting yang sudah dikemabangkan digunakan alfa dan beta testing untuk menentukan apakah aplikasi dapat diterima dan digunakan oleh pengguna.

Pengujian alfa dilakukan sebelum aplikasi yang dibuat diuji langsung oleh pengguna supaya nantinya aplikasi tersebut dapat berfungsi dengan baik di tangan pengguna. Pengujian alfa dilakukan dengan pengujian 32 tombol pada aplikasi dan melihat apakah tombol tersebut dapat berfungsi sesuai dengan napa yang diharapkan. Setelah dilakukan pengujian 32 tombol tersebut, didapatkan semua tombol berfungsi sesuai dengan napa yang diharapkan dan artinya pengujian alfa berhasil.

Pengujian beta dilakukan pada tanggal 16 Juli 2024 sampai dengan tanggal 22 Juli 2024. Pengujian dilakukan secara daring dengan memberikan kuesioner. Untuk memberi penjelasan bagaimana aplikasi Nutriscan bekerja, diberikan *link* video yang berisikan cara menggunakan aplikasi Nutriscan tersebut dari awal registrasi hingga pendeteksian stunting pada balita tersebut.

Terdapat sebanyak 49 responden telah mengisi kuesioner ini yang terdiri dari 38 orang berjenis kelamin perempuan dan 11 orang berjenis kelamin laki-laki. Responden terbanyak

berusia 20-39 tahun sebanyak 29 orang diikuti dengan usia 40-59 tahun sebanyak 18 orang dan yang terakhir berusia di bawah 20 tahun sebanyak 2 orang. Sebanyak 36 responden memiliki balita baik itu anak sendiri, saudara ataupun sepupu sedangkan sisanya tidak memiliki balita. Dikarenakan aplikasi ini ditujukan kepada responden yang memiliki balita berarti hanya 36 jawaban yang dapat digunakan untuk pengujian ini.



Gambar 17 Hasil Jawaban 36 Responden

V. KESIMPULAN

Dari penelitian, pengembangan, dan pengujian yang dilakukan dapat diambil kesimpulan bahwa aplikasi deteksi stunting dibuat dan dikembangkan menggunakan *framework* React Native dimana untuk sistem *backendnya* yang dimulai dari *database*, digunakan Firebase karena penggunaanya yang lebih mudah.

Untuk server digunakan *framework* FastAPI untuk melakukan integrasi antara algoritma *image processing* dengan aplikasi dilanjutkan dengan melakukan *tunneling* server yang sebelumnya lokal menjadi dapat diakses melalui internet menggunakan ngrok. Diakhiri dengan pengujian untuk menguji server dan aplikasi apakah keduanya dapat digunakan dan diterima pengguna.

REFERENSI

- [1] Prendergast, A. J., & Humphrey, J. H. (2014). The stunting syndrome in developing countries. *Paediatrics and International Child Health*, 34(4), 250-265.
- [2] de Onis, M., & Branca, F. (2016). Childhood stunting: A global perspective. *Maternal & Child Nutrition*, 12(S1), 12-26.
- [3] Holmes, M. (2022). *Creating Apps with React Native: Deliver Cross-Platform 0 Crash, 5 Star Apps*. Apress.
- [4] Kumar, A. (2018). *Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase*. Packt Publishing.
- [5] Parelta, JH. (2023). *Microservice APIs: Using Python, Flask, FastAPI, OpenAPI and More*.