



KERJA PRAKTIK - EF234603

Perancangan Web Virtual Try-On Berbasis Next.js

Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember

Kampus ITS Sukolilo Surabaya 60111

Periode: 15 Juli 2024 – 15 Oktober 2024

Oleh:

Widian Sasi Disertasiani 5025211024

Hana Maheswari 5025211182

Pembimbing Departemen

Dr. Kelly Rossa Sungkono, S.Kom., M.Kom.

Pembimbing Lapangan

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024



KERJA PRAKTIK - IF184801

Perancangan Web Virtual Try-On Berbasis Next.js

Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo Surabaya 60111

Periode: 15 Juli 2024 – 15 Oktober 2024

Oleh:

Widian Sasi Disertasiani 5025211024

Hana Maheswari 5025211182

Pembimbing Departemen

Dr. Kelly Rossa Sungkono, S.Kom., M.Kom.

Pembimbing Lapangan

Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2024

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI	i
DAFTAR GAMBAR	v
DAFTAR TABEL	vii
LEMBAR PENGESAHAN	ix
KATA PENGANTAR	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Tujuan	2
1.3. Manfaat.....	2
1.4. Rumusan Masalah	2
1.5. Lokasi dan Waktu Kerja Praktik.....	2
1.6. Metodologi Kerja Praktik	3
1.6.1. Perumusan Masalah	3
1.6.2. Studi Literatur.....	3
1.6.3. Analisis dan Perancangan	3
1.6.4. Implementasi	3
1.6.5. Hasil dan Pembahasan	3
1.6.6. Kesimpulan dan Saran	4
1.7. Sistematika Laporan	4
1.7.1. Bab I Pendahuluan.....	4
1.7.2. Bab II Profil Instansi.....	4
1.7.3. Bab III Tinjauan Pustaka	4
1.7.4. Bab IV Analisis dan Perancangan.....	4
1.7.5. Bab V Implementasi	4

1.7.6. Bab VI Pengujian dan Evaluasi	4
1.7.7. Bab VII Kesimpulan dan Saran	5
BAB II PROFIL PERUSAHAAN	7
2.1. Sejarah Instansi.....	7
2.2. Visi Instansi.....	8
2.3. Misi Instansi	8
2.4. Struktur Instansi.....	8
2.5. Laboratorium	9
2.6. Lokasi	11
BAB III TINJAUAN PUSTAKA.....	13
3.1. Web Virtual Try-On	13
3.2. Frontend Next.js	13
3.3. Backend API Firebase	14
3.4. Hugging Face	15
3.5. Firebase Storage	15
3.6. Integrasi dan Arsitektur Sistem.....	16
BAB IV ANALISIS DAN PERANCANGAN	17
4.1. Analisis Sistem	17
4.1.1. Kebutuhan Sistem.....	17
4.2. Perancangan Infrastruktur.....	18
4.2.1. Desain Arsitektur Sistem	19
4.3. Perancangan Antarmuka Pengguna (UI).....	21
BAB V IMPLEMENTASI.....	25
5.1. Implementasi Next.js	25
5.2. Implementasi Firebase	29

5.3. Implementasi Hugging Face	34
BAB VI HASIL DAN PEMBAHASAN	37
6.1. Hasil Evaluasi.....	37
6.2. Pembahasan.....	41
BAB VII KESIMPULAN DAN SARAN.....	43
7.1. Kesimpulan.....	43
7.2. Saran.....	44
DAFTAR PUSTAKA	45
BIODATA PENULIS.....	47

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Struktur Instansi Departemen Teknik Informatika ITS	9
Gambar 4.2.1 Desain Arsitektur Sistem	19
Gambar 4.3.1 Desain UI pertama	21
Gambar 4.3.2 Desain UI Kedua.....	21
Gambar 5.1 Implementasi Next.js	25
Gambar 6.1.1 Tampilan Aplikasi Virtual Try-On: Pilih Gambar, Pakaian, dan Generate	37
Gambar 6.1.2 Tampilan Aplikasi Virtual Try-On: Hasil	38

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 6.1. Hasil Evaluasi Pengujian	38
---	----

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

Perancangan Web Virtual Try-On Berbasis Next.js

Oleh:

Widian Sasi Disertasiani	5025211024
Hana Maheswari	5025211182

Disetujui oleh Pembimbing Kerja Praktik:

1. Dr. Kelly Rossa Sungkono,
S.Kom., M.Kom.
NIP. 1994201912088


(Pembimbing Departemen)

2. Shintami Chusnul Hidayati,
S.Kom., M.Sc., Ph.D.
NIP. 1987202012004


(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Perancangan Web Virtual Try-On Berbasis Next.js

Nama Mahasiswa : Widian Sasi Disertasiani
NRP : 5025211024
Nama Mahasiswa : Hana Maheswari
NRP : 5025211182
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Dr. Kelly Rossa Sungkono, S.Kom.,
M.Kom.
Pembimbing Lapangan : Shintami Chusnul Hidayati, S.Kom.,
M.Sc., Ph.D.

ABSTRAK

Web virtual try-on adalah solusi inovatif untuk mencoba pakaian secara virtual. Penelitian ini mengembangkan aplikasi berbasis Next.js dengan integrasi Firebase untuk penyimpanan data dan Hugging Face untuk pemrosesan gambar, bertujuan meningkatkan pengalaman pengguna melalui proses yang cepat dan akurat. Metode pengembangan melibatkan arsitektur client-server dan pengujian dilakukan untuk memastikan integrasi yang efisien, akurasi hasil gambar, dan kemudahan penggunaan. Hasil menunjukkan aplikasi berfungsi dengan baik, menghasilkan visualisasi pakaian yang realistis dengan waktu respons cepat. Hal ini menjadikan aplikasi sebagai solusi potensial yang efektif untuk virtual try-on.

Kata Kunci : Firebase, Hugging Face, Next.js, Virtual try-on

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Perancangan Web Virtual Try-On Berbasis Next.js.

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Ibu Dr. Kelly Rossa Sungkono, S.Kom., M.Kom. selaku dosen pembimbing kerja praktik.
3. Ibu Shintami Chusnul Hidayati, S.Kom., M.Sc., Ph.D. selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 31 Desember 2024

Widian Sasi Disertasiani dan Hana Maheswari

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Industri mode terus mengalami transformasi seiring dengan perkembangan teknologi digital. Salah satu inovasi terkini adalah aplikasi virtual try-on yang memungkinkan pengguna mencoba pakaian secara virtual tanpa perlu hadir secara fisik di toko. Teknologi ini menawarkan solusi praktis untuk mengatasi kendala waktu, lokasi, dan keterbatasan fasilitas, serta memberikan pengalaman belanja yang lebih interaktif dan personal. Penggunaan virtual try-on menjadi semakin relevan di tengah tren belanja online yang terus meningkat. Konsumen menginginkan cara yang lebih efektif untuk memastikan kecocokan pakaian sebelum melakukan pembelian [1]. Di sisi lain, pelaku industri mode menghadapi tantangan dalam meningkatkan pengalaman pelanggan sekaligus mengurangi angka pengembalian produk akibat ketidaksesuaian [2]. Teknologi virtual try-on ada untuk memenuhi kebutuhan tersebut.

Dalam penelitian ini, aplikasi web virtual try-on dirancang menggunakan Next.js sebagai framework frontend untuk mendukung antarmuka interaktif, Firebase sebagai platform penyimpanan cloud, serta Hugging Face untuk pemrosesan gambar berbasis machine learning. Integrasi teknologi ini bertujuan menciptakan sistem yang tidak hanya efisien, tetapi juga memberikan hasil visual yang akurat. Pengembangan sistem berbasis web ini menerapkan arsitektur client-server dalam membangun aplikasi yang scalable dan mudah diakses. Melalui penelitian ini, diharapkan aplikasi virtual try-on yang dihasilkan dapat menjadi solusi inovatif yang mendukung transformasi digital di industri mode dan memberikan pengalaman belanja yang lebih baik bagi konsumen.

1.2. Tujuan

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai kerja praktik sebesar 2 sks dan membantu mengembangkan aplikasi web virtual try-on berbasis Next.js, guna meningkatkan pengalaman pengguna dalam mencoba pakaian secara virtual.

1.3. Manfaat

Manfaat yang diperoleh dengan adanya aplikasi web virtual try-on adalah mempermudah pengguna untuk mencoba pakaian secara virtual tanpa harus datang langsung ke toko fisik. Pengguna cukup mengunggah gambar melalui aplikasi dan dapat melihat hasil simulasi pakaian yang dikenakan secara akurat. Selain itu, aplikasi dapat membantu industri mode dalam meningkatkan pengalaman pelanggan dan mengurangi angka pengembalian barang akibat ketidaksesuaian ukuran atau model pakaian.

1.4. Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana merancang arsitektur aplikasi web virtual try-on berbasis Next.js yang dapat menyediakan layanan interaktif dan mudah digunakan?
2. Bagaimana cara mengintegrasikan Firebase untuk penyimpanan data dan Hugging Face untuk pemrosesan gambar?
3. Bagaimana hasil evaluasi aplikasi web virtual try-on?

1.5. Lokasi dan Waktu Kerja Praktik

Pengerjaan kerja praktik ini lakukan secara *remote*. Adapun kerja praktik dimulai pada tanggal 15 Juli 2024 hingga 15 Oktober 2024.

1.6. Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik:

1.6.1. Perumusan Masalah

Untuk mengetahui kebutuhan sistem aplikasi web virtual try-on, dilakukan diskusi awal dengan pembimbing lapangan, Bu Shintami. Pada pertemuan tersebut, dibahas mengenai kebutuhan pengguna, termasuk bagaimana sistem virtual try-on diharapkan bekerja. Fitur-fitur utama seperti integrasi Firebase untuk penyimpanan data dan Hugging Face untuk pemrosesan gambar dirumuskan berdasarkan kebutuhan fungsionalitas dan pengalaman pengguna.

1.6.2. Studi Literatur

Setelah mendapatkan gambaran awal mengenai sistem yang akan dikembangkan, dilakukan studi literatur terkait teknologi yang akan digunakan, seperti Next.js, Firebase, dan Hugging Face. Studi ini mencakup tinjauan dokumentasi teknis dan praktik terbaik dalam mengintegrasikan teknologi tersebut ke dalam aplikasi web.

1.6.3. Analisis dan Perancangan

Pada tahap ini, dilakukan analisis terhadap kebutuhan sistem dan perancangan arsitektur client-server. Desain mencakup diagram alur data, antarmuka pengguna (UI/UX), serta skema integrasi antara komponen frontend dan backend. Firebase dipilih untuk pengelolaan data yang mudah diakses dan Hugging Face dipilih untuk menghasilkan try-on yang akurat.

1.6.4. Implementasi

Implementasi dilakukan dengan membangun aplikasi menggunakan Next.js sebagai framework frontend. Firebase diintegrasikan untuk penyimpanan gambar, backend menggunakan layanan API Hugging Face untuk pemrosesan gambar. Tahap ini meliputi pengkodean, pengujian unit, serta pengaturan deployment.

1.6.5. Hasil dan Pembahasan

Setelah implementasi selesai, dilakukan pengujian menyeluruh untuk memastikan integrasi antara komponen

frontend, backend, dan penyimpanan berjalan sesuai harapan. Evaluasi melibatkan pengujian kecepatan proses, akurasi hasil virtual try-on, serta kemudahan penggunaan. Feedback dari pembimbing lapangan digunakan untuk melakukan perbaikan dan optimasi.

1.6.6. Kesimpulan dan Saran

Kesimpulan dibuat berdasarkan hasil pengujian dan evaluasi terhadap sistem yang dikembangkan. Saran diberikan untuk pengembangan lebih lanjut, seperti penambahan fitur yang mendukung pengalaman pengguna.

1.7. Sistematika Laporan

Sistematika laporan pada buku kerja praktik:

1.7.1. Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Instansi

Bab ini berisi sejarah instansi, visi instansi, misi instansi, struktur instansi, laboratorium, dan lokasi instansi.

1.7.3. Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4. Bab IV Analisis dan Perancangan

Bab ini berisi mengenai tahap analisis sistem aplikasi dalam menyelesaikan proyek kerja praktik.

1.7.5. Bab V Implementasi

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi web.

1.7.6. Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari web yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7. Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Sejarah Instansi

Perkembangan teknologi informasi telah mempengaruhi berbagai aspek kehidupan manusia dan menjadi faktor utama dalam pertumbuhan ekonomi suatu negara, baik pada masa kini maupun di masa depan. Menyadari pentingnya hal ini, pemerintah Indonesia mulai mengutamakan pendidikan tinggi di bidang komputer dan informatika sejak Rencana Pembangunan Lima Tahun (Repelita) V. Selain bidang komputer dan informatika, sektor lain seperti rekayasa, manajemen, akuntansi, serta seni juga menjadi fokus dalam kebijakan pendidikan tinggi. Tujuan dari kebijakan ini adalah untuk mempersiapkan Indonesia menghadapi era pembangunan industri dan teknologi informasi. Pada tahun 1985, Direktorat Jenderal Pendidikan Tinggi mengeluarkan kebijakan untuk membuka program studi Sarjana (S1) di bidang ilmu komputer dan teknologi informasi di beberapa universitas yang memiliki program Studi Teknik Informatika. Program ini awalnya dikenal dengan nama Program Studi Teknik Komputer, namun pada tahun 1993, namanya berubah menjadi Jurusan Teknik Komputer. Kemudian, pada tahun 1996, Surat Keputusan Direktur Jenderal Pendidikan Tinggi Nomor 224/DIKTI/Kep/1996 diterbitkan. Berdasarkan Surat Keputusan Nomor 003/BAN-PT/Ak-X/S1/V/2006 yang diterbitkan oleh Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) pada 18 Mei 2006, Jurusan Teknik Informatika mendapatkan akreditasi A. Selain program Sarjana (S1), Jurusan Teknik Informatika juga menawarkan program Pascasarjana (S2) yang dimulai pada tahun 1994 berdasarkan Surat Keputusan Direktur Jenderal Pendidikan Tinggi No. 2851/D/T/2001. Pada tahun 2011, Jurusan Teknik Informatika juga meluncurkan program Doktor (S3) untuk meningkatkan kualitas pendidikan dan penelitian di bidang ini. [3]

2.2. Visi Instansi

Seiring dengan visi ITS untuk menjadi perguruan tinggi terkemuka di dunia dalam bidang ilmu pengetahuan, teknologi, dan seni, dengan fokus pada pengembangan industri dan kelautan yang berkelanjutan, visi Departemen Informatika adalah menjadi pionir dalam bidang informatika yang diakui di tingkat nasional dan internasional, serta aktif berkontribusi pada kemajuan dan kesejahteraan masyarakat. [4]

2.3. Misi Instansi

Berikut adalah misi-misi yang dimiliki oleh Departemen Teknik Informatika, ITS [5].

1. Menyelenggarakan proses pembelajaran yang berkualitas, dan memenuhi standar nasional maupun internasional.
2. Melaksanakan penelitian yang inovatif, bermutu, dan bermanfaat.
3. Meningkatkan pemanfaatan teknologi informasi dan komunikasi untuk masyarakat.
4. Menjalin kemitraan dengan berbagai lembaga, baik di dalam maupun di luar negeri.

2.4. Struktur Instansi

Gambar 2.1 merupakan gambar struktur instansi pada Departemen Teknik Informatika ITS [5].

3. Laboratorium Grafika dan Interaksi

Laboratorium ini berfokus pada desain, pengembangan, dan pembuatan game serta aplikasi realitas virtual tiga dimensi, termasuk penggunaan game engine dan pemrograman grafis 3D.

4. Laboratorium Manajemen Informasi Cerdas

Bidang keahlian di laboratorium ini meliputi analisis dan evaluasi sistem informasi serta rekayasa pengetahuan untuk meningkatkan tata kelola teknologi informasi dalam konteks bisnis dan organisasi.

5. Laboratorium Komputasi Jaringan

Laboratorium ini memfokuskan pada pembangunan infrastruktur jaringan yang aman, sistem grid, serta aplikasi multimedia berbasis jaringan yang mendukung komunikasi data.

6. Laboratorium Keamanan Jaringan dan Teknologi Siber

Laboratorium ini memberikan pelatihan dalam pengamanan jaringan, dengan fokus pada penerapan arsitektur jaringan dan teknologi terbaru untuk memastikan keamanan sistem informasi.

7. Laboratorium Algoritma dan Pemrograman

Di laboratorium ini, mahasiswa dilatih untuk merancang dan menganalisa algoritma serta mengimplementasikannya dalam bahasa pemrograman yang sesuai, guna menyelesaikan berbagai permasalahan secara efisien.

8. Laboratorium Pemodelan dan Simulasi Terapan

Laboratorium ini mendukung penelitian dan kolaborasi industri di bidang pemodelan, simulasi, dan komputasi saintifik, serta penerapan dalam optimasi dan peramalan sains.

Selain laboratorium bidang minat, terdapat juga beberapa laboratorium workshop yang mendukung proses pembelajaran praktis mahasiswa:

1. Workshop Pemrograman 1

Workshop ini menyediakan fasilitas komputer dan jaringan yang mendukung pengajaran dan praktik pemrograman. Ruangannya dapat menampung hingga 54 orang.

2. Workshop Pemrograman 2

Laboratorium ini dilengkapi dengan fasilitas yang sama dengan Workshop Pemrograman 1, namun digunakan untuk kegiatan pemrograman tingkat lanjut.

3. Laboratorium Pascasarjana S2

Laboratorium ini dirancang khusus untuk mendukung kegiatan penelitian dan penulisan tesis mahasiswa program master, serta menyediakan akses ke fasilitas laboratorium bidang minat.

4. Laboratorium Pascasarjana S3

Laboratorium ini diperuntukkan bagi mahasiswa program doktor untuk menyelesaikan tugas kuliah, disertasi, dan penelitian. Laboratorium ini terbagi dalam beberapa lantai yang disesuaikan dengan status pendaftaran mahasiswa S3.

2.6. Lokasi

Departemen Teknik Informatika, Kampus ITS Sukolilo, Surabaya 60111.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Web Virtual Try-On

Virtual Try-On (VTO) adalah aplikasi berbasis web yang memungkinkan pengguna untuk mencoba pakaian atau aksesoris secara virtual melalui teknologi pemrosesan gambar dan pembelajaran mesin. Dalam aplikasi VTO, pengguna mengunggah gambar diri mereka, memilih pakaian yang ingin dicoba, dan sistem akan memadukan gambar pengguna dengan gambar model pakaian menggunakan teknik berbasis Generative Adversarial Networks (GANs) atau model pemrosesan gambar lainnya. Teknologi ini menggabungkan gambar pengguna dengan pakaian untuk menghasilkan gambar try-on yang realistis, memberi gambaran visual tentang bagaimana pakaian tersebut akan terlihat saat dikenakan. VTO umumnya menggunakan berbagai teknologi seperti pemodelan 3D, deteksi objek, dan pemrosesan citra untuk mencapai hasil yang memuaskan dan interaktif. [7]

3.2. Frontend Next.js

Frontend adalah bagian dari arsitektur yang bertanggung jawab untuk antarmuka pengguna (UI), interaksi, dan penyajian hasil try-on. Dalam pengembangan aplikasi VTO, frontend harus mampu menangani banyak interaksi pengguna yang dinamis dan memberikan pengalaman yang intuitif serta responsif. Next.js adalah framework yang sangat tepat untuk pengembangan frontend dalam aplikasi semacam ini.

Next.js memiliki kemampuan untuk melakukan *server-side rendering* (SSR), yang memungkinkan pengiriman halaman dengan cepat bahkan sebelum JavaScript dieksekusi sepenuhnya di sisi klien [8]. Hal ini sangat bermanfaat dalam aplikasi seperti VTO, yang membutuhkan kecepatan *loading* tinggi karena pengguna sering kali mengunggah gambar untuk diproses. Selain

SSR, Next.js juga mendukung pengembangan aplikasi satu halaman (*single-page application*), yang memberikan pengalaman interaktif tanpa perlu memuat ulang seluruh halaman [7].

Dengan adanya fitur seperti server-side rendering dan pengelolaan rute dinamis, Next.js memberikan keunggulan dalam membangun antarmuka yang dapat menangani permintaan API secara efisien, mengelola data interaktif, dan mempermudah integrasi dengan backend.

3.3. Backend API Firebase

Backend dalam arsitektur virtual try-on berfungsi sebagai penghubung antara frontend dan sistem pemrosesan gambar. Tugas utamanya mencakup penanganan permintaan dari frontend, seperti unggahan gambar pengguna, komunikasi dengan sistem penyimpanan, dan layanan pemrosesan gambar. Firebase merupakan platform pengembangan aplikasi yang komprehensif, memungkinkan pengembang untuk fokus pada pengalaman pengguna tanpa harus mengelola infrastruktur server. Dengan fitur seperti Firebase Realtime Database, pengembang dapat menyimpan dan menyinkronkan data antara pengguna secara real-time, yang sangat penting untuk aplikasi interaktif seperti virtual try-on. [9]

Pada Firebase terdapat layanan Cloud Functions yang memungkinkan eksekusi kode backend tanpa perlu mengelola server secara manual. Fitur ini memudahkan pengembangan API yang dapat menangani logika aplikasi dan integrasi dengan layanan lain [10]. Dengan memanfaatkan Firebase sebagai backend, pengembang dapat membangun aplikasi virtual try-on yang efisien dan scalable, menggunakan dukungan fitur-fitur yang telah disediakan platform.

3.4. Hugging Face

Hugging Face adalah platform yang menyediakan berbagai model pembelajaran mesin yang dapat diterapkan dalam sistem virtual try-on. Salah satu contohnya adalah model “DeepVTO,” yang dibangun berdasarkan prinsip *stable diffusion* dan *vector embeddings*, dilatih menggunakan model DreamBooth. Model ini dirancang untuk menghasilkan simulasi pakaian yang realistis pada gambar pengguna secara real-time [11]. Selain itu, Hugging Face juga menyediakan berbagai model lain yang dapat digunakan untuk keperluan virtual try-on, seperti model “IDM-VTON” yang merupakan implementasi dari penelitian “Improving Diffusion Models for Authentic Virtual Try-on in the Wild.” [12]

Dengan memanfaatkan model-model yang telah tersedia, pengembang dapat membangun aplikasi virtual try-on yang efisien dan akurat, memberikan pengalaman interaktif lebih baik bagi pengguna.

3.5. Firebase Storage

Firebase Storage adalah layanan penyimpanan berbasis cloud yang dirancang untuk menyimpan dan mengelola file dalam skala besar, termasuk gambar dan video. Dalam aplikasi virtual try-on, Firebase Storage digunakan untuk menyimpan gambar yang diunggah oleh pengguna, seperti foto pakaian dan foto pengguna yang akan diproses. Keunggulan utama Firebase Storage terletak pada kemudahan integrasinya dengan layanan lain dalam ekosistem Firebase, Authentication untuk mengamankan akses file dan Firestore untuk menyimpan metadata file nama, ukuran, dan waktu unggahan. Selain itu, Firebase Storage menggunakan protokol yang aman untuk transfer data, memastikan bahwa file yang diunggah dan diunduh tetap terlindungi. Hal ini menjadikan Firebase Storage cocok untuk aplikasi VTO yang memerlukan pengelolaan data gambar dengan jumlah besar dan skalabilitas tinggi. [13]

3.6. Integrasi dan Arsitektur Sistem

Dalam pengembangan aplikasi virtual try-on, arsitektur sistem yang digunakan adalah client-server. Dalam arsitektur ini, frontend bertindak sebagai klien yang berinteraksi langsung dengan pengguna, sementara backend berfungsi sebagai server yang mengelola logika aplikasi dan menghubungkan berbagai layanan. Frontend yang dibangun dengan Next.js berkomunikasi dengan backend untuk mengirim permintaan, seperti mengunggah gambar pengguna atau mengambil hasil pemrosesan try-on. Backend yang dirancang dengan API, bertanggung jawab menangani permintaan tersebut, mengelola autentikasi pengguna menggunakan Firebase Authentication, menyimpan gambar di Firebase Storage, dan meneruskan permintaan ke pemrosesan gambar menggunakan model dari Hugging Face. Pendekatan client-server ini memastikan bahwa tugas-tugas intensif, pemrosesan gambar, dilakukan di server sehingga klien tetap ringan dan responsif. Dengan memanfaatkan arsitektur ini, aplikasi VTO dapat memberikan pengalaman pengguna yang cepat dan efisien.

Salah satu contoh implementasi arsitektur client-server dalam aplikasi VTO adalah sistem yang dikembangkan oleh VICO-DR, yang menggunakan server terpusat untuk mengelola dan memproses aliran media, sehingga mengurangi beban pada perangkat klien dan meningkatkan efisiensi sistem secara keseluruhan [14]. Selain itu, penelitian yang dilakukan dalam “Designing an AI-Based Virtual Try-On Web Application” juga menunjukkan pentingnya arsitektur client-server dalam pengembangan aplikasi VTO berbasis AI, dimana server bertanggung jawab untuk pemrosesan gambar dan manajemen data, klien fokus pada interaksi pengguna dan penyajian hasil try-on [7].

BAB IV

ANALISIS DAN PERANCANGAN

4.1. Analisis Sistem

Pada bab ini, akan dibahas mengenai tahapan dalam membangun aplikasi Virtual Try-On yang digunakan untuk memberikan pengalaman mencoba pakaian secara virtual. Proses analisis sistem ini mencakup dua bagian penting, yaitu definisi umum aplikasi dan analisis kebutuhan. Definisi umum aplikasi memberikan gambaran mengenai bagaimana aplikasi bekerja, sedangkan analisis kebutuhan berfokus pada aspek teknis dan fungsional yang diperlukan untuk memastikan aplikasi berjalan dengan optimal.

4.1.1. Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk memastikan web VTO dapat berfungsi dengan baik dan memenuhi harapan pengguna. Kebutuhan sistem ini mencakup aspek perangkat keras, perangkat lunak, serta kebutuhan pengguna dan admin.

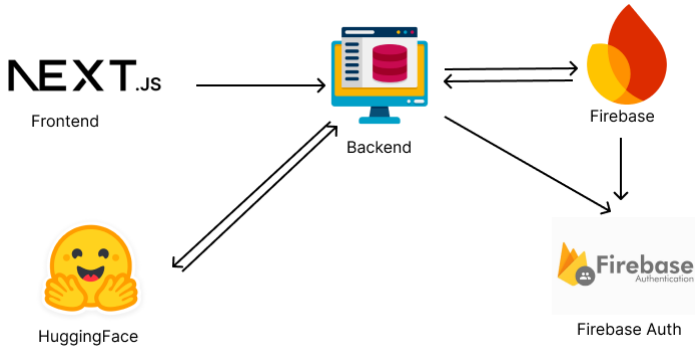
1. Kebutuhan Perangkat Keras

- a. Server:** Dibutuhkan server yang cukup kuat untuk menangani pemrosesan gambar dan beban aplikasi secara keseluruhan.
- b. Penyimpanan:** Firebase Storage digunakan untuk menyimpan gambar pengguna dan pakaian. Penyimpanan ini harus cukup besar untuk menampung berbagai file gambar yang diunggah, memungkinkan akses cepat dan aman ke file tersebut.

2. Kebutuhan Perangkat Lunak

- a. Frontend:** Dibangun menggunakan **Next.js**, memastikan aplikasi dapat berjalan dengan baik di berbagai platform perangkat dan menawarkan pengalaman pengguna yang cepat dan responsif.

4.2.1. Desain Arsitektur Sistem



Gambar 4.2.1 Desain Arsitektur Sistem

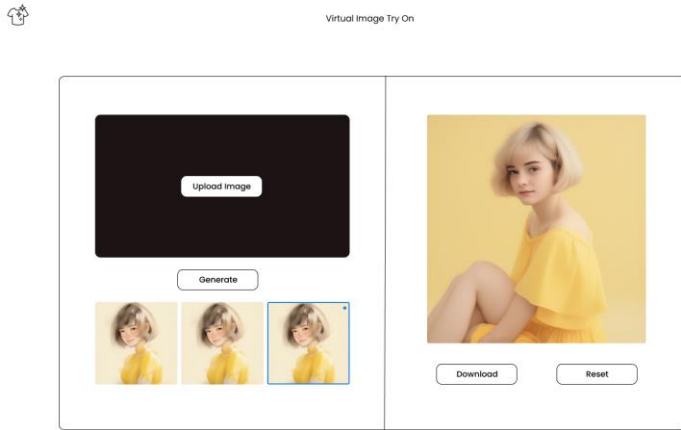
Arsitektur aplikasi VTO menggunakan pola client-server yang terdiri dari beberapa komponen utama yang saling terhubung, yaitu frontend, backend, dan layanan cloud untuk pemrosesan gambar serta penyimpanan data. Setiap komponen memiliki peran spesifik dalam memastikan aplikasi berjalan dengan baik dan responsif.

- a. **Frontend (Client):** Frontend aplikasi dibangun dengan Next.js karena mampu menyediakan rendering sisi server (SSR) dan pembuatan aplikasi web yang cepat dan responsif. Frontend bertugas untuk memberikan antarmuka pengguna yang intuitif, memungkinkan pengguna mengunggah gambar diri, memilih pakaian yang ingin dicoba, menampilkan hasil try-on secara real-time. Next.js juga memungkinkan frontend berkomunikasi dengan backend dengan mudah.
- b. **Backend (Server):** Backend aplikasi menggunakan Flask menangani logika pemrosesan data, autentikasi pengguna, dan interaksi dengan layanan eksternal. Backend berfungsi sebagai jembatan antara frontend dan sistem pemrosesan gambar serta penyimpanan data. Semua permintaan gambar pengguna dikirim ke backend, yang kemudian mengelola dan

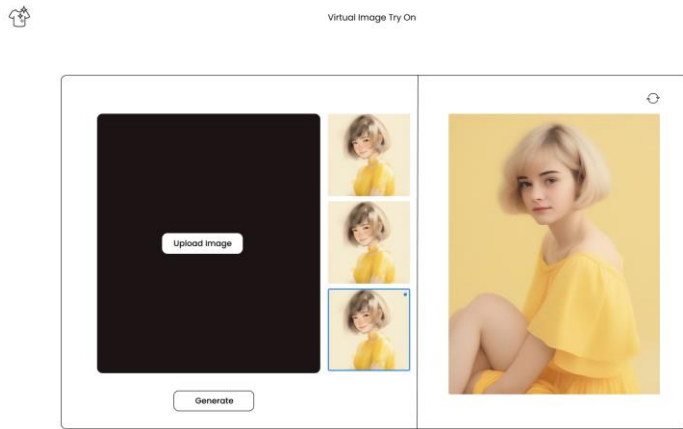
meneruskan gambar tersebut ke Hugging Face untuk pemrosesan dan Firebase Storage untuk penyimpanan.

- c. Pemrosesan Gambar (Hugging Face):** Pemrosesan gambar dilakukan menggunakan model machine learning yang menyediakan platform untuk mengakses model generatif try-on. Model ini diimplementasikan untuk mencocokkan pakaian dengan gambar pengguna secara otomatis dan menghasilkan gambar try-on yang realistis. Dengan ini, aplikasi dapat memanfaatkan kemampuan komputasi tanpa perlu sumber daya lokal yang besar.
- d. Penyimpanan Data (Firebase Storage):** Firebase Storage digunakan untuk menyimpan gambar pengguna dan pakaian yang akan diproses. Gambar pengguna dan produk akan diunggah ke Firebase Storage, yang kemudian dapat diakses dan digunakan oleh backend untuk pemrosesan lebih lanjut. Penyimpanan ini memungkinkan akses data yang cepat dan aman dari berbagai perangkat.
- e. Keamanan dan Autentikasi:** Untuk menjaga keamanan data pengguna, diimplementasikan Firebase Authentication, yang memungkinkan pengguna untuk login dengan aman. Autentikasi ini berfungsi mengamankan data pribadi pengguna, termasuk gambar yang diunggah. Backend aplikasi akan memastikan hanya pengguna terautentikasi yang dapat mengakses data dan mengelola preferensi.

4.3. Perancangan Antarmuka Pengguna (UI)



Gambar 4.3.1 Desain UI pertama



Gambar 4.3.2 Desain UI Kedua

Perancangan UI untuk aplikasi Virtual Try-On dilakukan untuk memastikan pengalaman pengguna yang intuitif dan responsif. UI dirancang dengan mempertimbangkan kemudahan

penggunaan, estetika, serta fungsionalitas yang mendukung proses mencoba pakaian secara virtual. Beberapa elemen utama yang dipertimbangkan dalam perancangan UI:

a. Antarmuka Pengguna yang Intuitif

UI mempermudah pengguna dalam mengunggah gambar dan memilih pakaian yang ingin dicoba. Desain yang dibuat bersih dan sederhana sehingga memudahkan pengguna memahami cara kerjanya. Pengguna akan diminta untuk mengunggah foto diri, kemudian memilih pakaian yang tersedia. Pengguna dapat melihat pratinjau hasil try-on dalam waktu nyata, sehingga keputusan dapat diambil dengan mudah.

b. Navigasi yang Jelas

Navigasi yang jelas dan mudah diakses diterapkan agar pengguna dapat dengan cepat beralih ke fitur lain seperti mengunggah gambar, memilih pakaian, dan melihat hasil try-on. Tombol dan elemen navigasi ditempatkan di bagian kiri layar agar pengguna dapat menavigasi dengan nyaman.

c. Visualisasi Hasil Try-On

Hasil dari proses try-on ditampilkan dengan jelas. Hal ini dilakukan dengan menampilkan gambar pada bagian kanan layar. Setelah proses try-on selesai, pengguna memiliki kemampuan mendownload gambar untuk mendapatkan pandangan lebih detail dan mereset gambar untuk proses try-on selanjutnya.

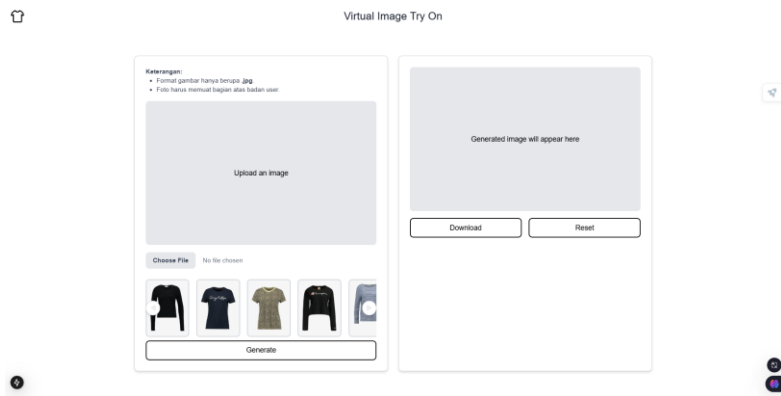
Berdasarkan elemen-elemen yang telah dipertimbangkan, desain UI yang dipilih adalah desain pertama pada Gambar 4.3.1. Desain ini dinilai lebih umum dan intuitif, terutama dengan fitur scrolling pilihan pakaian secara horizontal yang sudah familier bagi banyak pengguna. Selain itu, tombol “Download” dan “Reset” dirancang menggunakan teks yang jelas sehingga lebih mudah dipahami dibandingkan simbol. Dengan merancang UI yang memenuhi kebutuhan fungsional dan estetika, aplikasi VTO diharapkan menjadi lebih mudah diakses, digunakan, dan memberikan pengalaman yang menyenangkan bagi pengguna. Desain UI yang baik juga berkontribusi pada peningkatan interaksi

pengguna dengan aplikasi, sekaligus memastikan tingkat kepuasan yang lebih tinggi terhadap pengalaman virtual try-on yang ditawarkan.

[Halaman ini sengaja dikosongkan]

BAB V IMPLEMENTASI

5.1. Implementasi Next.js



Gambar 5.1 Implementasi Next.js

Frontend dirancang menggunakan framework Next.js, yang memungkinkan pembuatan antarmuka pengguna interaktif. Komponen utama adalah halaman upload, di mana pengguna dapat:

- Mengunggah gambar model.
- Memilih pakaian pada daftar pakaian yang diambil dari firebase.
- Mengirim data ke *backend* untuk diproses.

Penjelasan Alur kode:

1. Deklarasi Awal dan State Management

Kode dimulai dengan impor dependensi React seperti `useState`, `useEffect`, dan `useRef`. Dependensi Firebase (`ref`, `listAll`, `getDownloadURL`) digunakan untuk mengambil data dari penyimpanan Firebase. State `person`, `clothes`, `selectedClothes`, `imageUrl`, dan `isLoading` digunakan untuk menyimpan data pengguna seperti gambar yang diunggah, daftar pakaian, pakaian

yang dipilih, URL hasil gambar yang dihasilkan, serta status pemrosesan. State dan ref tambahan digunakan untuk mengelola drag-scroll pada carousel pakaian.

```
import React, { useState, useEffect, useRef } from
"react";
import "../globals.css";
import { ref, listAll, getDownloadURL } from
"firebase/storage";
import { storage } from "../lib/firebase";

const Upload: React.FC = () => {
  const [person, setPerson] = useState<File |
null>(null);
  const [clothes, setClothes] =
useState<string[]>([]);
  const [selectedClothes, setSelectedClothes] =
useState<string>("");
  const [imageUrl, setImageUrl] = useState<string |
null>(null);
  const [isLoading, setIsLoading] =
useState<boolean>(false);

  const carouselRef = useRef<HTMLDivElement>(null);
  const isDragging = useRef(false);
  const startX = useRef(0);
  const scrollLeft = useRef(0);
```

2. Fetch Data Pakaian

useEffect pertama digunakan untuk mengambil daftar pakaian dari Firebase Storage. Fungsi fetchClothes memanfaatkan listAll untuk mendapatkan referensi item dalam direktori “clothes” dan getDownloadURL untuk menghasilkan URL akses ke masing-masing pakaian. Hasilnya disimpan dalam state *clothes*.

```

useEffect(() => {
  const fetchClothes = async () => {
    const listRef = ref(storage, "clothes");
    const res = await listAll(listRef);

    const clothesUrls = await Promise.all(
      res.items.map(async (itemRef) => {
        const url = await getDownloadURL(itemRef);
        return url;
      })
    );

    setClothes(clothesUrls);
  };

  fetchClothes();
}, []);

const scrollLeft = useRef(0);

```

3. Unggah Foto

Fungsi `handlePersonUpload` menangani unggahan foto oleh pengguna. Fungsi ini memvalidasi format file agar hanya menerima .jpg. Jika file tidak valid, input akan di-reset, dan pengguna diberi peringatan.

```

const handlePersonUpload = (event:
React.ChangeEvent<HTMLInputElement>) => {
  if (event.target.files && event.target.files[0]) {
    const file = event.target.files[0];
    if (!file.name.toLowerCase().endsWith(".jpg")) {
      alert("Format file tidak diterima. Pastikan
formatnya berupa .jpg.");
      event.target.value = "";
    }
  }
};

```

```

        return;
    }
    setPerson(file);
}
};

```

4. Submit dan Kirim Data ke Backend

Fungsi `handleSubmit` memproses unggahan gambar dan pakaian yang dipilih. Data dikirim ke endpoint backend `http://localhost:8000/upload` menggunakan metode POST dengan objek `FormData`. Fungsi ini mengelola status loading dengan `setIsLoading` serta mengupdate URL hasil gambar (`imageUrl`) berdasarkan respons backend. Jika terjadi kesalahan, pesan kesalahan akan ditampilkan.

```

const handleSubmit = async (e: React.FormEvent) => {
    e.preventDefault();
    if (!person || !selectedClothes) {
        alert("Please upload a person image and select clothes.");
        return;
    }

    setIsLoading(true);

    const formData = new FormData();
    const decodedClothesUrl =
decodeURIComponent(selectedClothes);
    const clothesFileName =
decodedClothesUrl.split("/").pop()?.split("?")[0] ||
selectedClothes;
    formData.append("person", person);
    formData.append("garment", clothesFileName);

```

```

    try {
        const response = await
fetch("http://localhost:8000/upload", {
    method: "POST",
    body: formData,
});
const result = await response.json();

    if (result.status === "success") {
        setImageUrl(result.image_url);
    } else {
        alert("Upload failed: " + result.message);
    }
} catch (error) {
    console.error("Error uploading files:", error);
    alert("Upload failed!");
} finally {
    setIsLoading(false);
}
};

```

5.2. Implementasi Firebase

Backend bertugas mengolah data yang dikirimkan dari frontend, yaitu gambar pengguna dan pakaian yang dipilih. *FastAPI* digunakan untuk mengelola permintaan dan menangani logika pengolahan data.

Penjelasan Alur kode:

1. Inisialisasi Firebase Admin SDK

Kode dimulai dengan mengimpor dependensi dan inisialisasi dengan memuat file kredensial Firebase menggunakan

credentials.Certificate yang berfungsi mengautentikasi aplikasi dengan Firebase. Selanjutnya, fungsi initialize_app digunakan untuk menginisialisasi Firebase Admin SDK dan menghubungkan aplikasi ke bucket penyimpanan Firebase dengan nama viton-403b8.appspot.com.

```
import uuid
from fastapi import FastAPI, File, UploadFile, Form
from fastapi.responses import JSONResponse
from fastapi.staticfiles import StaticFiles
from fastapi.middleware.cors import CORSMiddleware
from app.upload import process_images
import os
import firebase_admin
from firebase_admin import credentials, storage

cred =
credentials.Certificate("C:/Users/widia/Downloads/ogik
-hana-be-main/credentials.json")
firebase_admin.initialize_app(cred, {
    "storageBucket": "viton-403b8.appspot.com"
})
```

2. Konfigurasi FastAPI

Konfigurasi dimulai dengan menambahkan middleware CORS menggunakan app.add_middleware yang memungkinkan semua domain untuk mengakses API (allow_origins=["*"]) mengizinkan metode HTTP dan header (allow_methods=["*"] dan allow_headers=["*"]). Konfigurasi ini menyertakan StaticFiles melalui app.mount yang menyediakan akses ke file statis dari folder bernama static sehingga file gambar atau skrip dapat diakses dengan mudah.

```
app = FastAPI()
app.add_middleware(
    CORSMiddleware,
```

```

    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

app.mount("/static", StaticFiles(directory="static"),
name="static")

```

3. Endpoint /upload

Pada endpoint diterima dua parameter utama: person, yaitu file gambar tubuh pengguna yang diunggah, dan garment, nama file pakaian yang diambil dari Firebase Storage. Proses dimulai dengan menyimpan gambar tubuh pengguna di folder static/images menggunakan nama file unik yang dihasilkan oleh uuid. File ini dibaca dari parameter person dan disimpan secara lokal. Selanjutnya, data gambar pakaian diunduh dari Firebase Storage berdasarkan nama file yang diterima dari parameter garment dan disimpan secara lokal.

Fungsi process_images digunakan untuk menggabungkan gambar tubuh dengan pakaian secara virtual, menghasilkan dua gambar: gambar hasil dan gambar bertopeng (masked image). Kedua gambar tersebut diberi nama unik dan diunggah kembali ke Firebase Storage di folder output dan masked_output. URL publik dari gambar hasil dan masked image kemudian diambil dari Firebase Storage, hasilnya akan dikembalikan sebagai respons JSON yang berisi status sukses dan URL gambar hasil.

```

@app.post("/upload")
async def upload_image(
    person: UploadFile = File(...),
    garment: str = Form(...),
):
    person_filename =
f"static/images/{uuid.uuid4()}.png"

```



```

    garment_filename =
f"static/images/{uuid.uuid4()}.png"

    try:
        bucket = storage.bucket()
        blob = bucket.blob(f"clothes/{garment}")
        garment_data = blob.download_as_bytes()

    try:
        result_image, masked_image =
process_images(person_filename, garment_filename)

        output_filename = f"output/{uuid.uuid4()}.png"
        masked_filename =
f"masked_output/{uuid.uuid4()}.png"
        output_blob = bucket.blob(output_filename)
        output_blob.upload_from_filename(result_image)
        masked_blob = bucket.blob(masked_filename)
        masked_blob.upload_from_filename(masked_image)

        result_image_url = output_blob.public_url
        masked_image_url = masked_blob.public_url

    return JsonResponse(content={
        "status": "success",
        "image_url": result_image_url,
        "masked_image_url": masked_image_url
    })

```

4. Endpoint /

Endpoint digunakan untuk mengambil daftar semua gambar pakaian yang tersimpan di Firebase Storage pada folder clothes/. Prosesnya dimulai dengan mengakses bucket Firebase

melalui `storage.bucket()`, lalu menggunakan metode `list_blobs` untuk mendapatkan semua file yang berada dalam folder tersebut dengan prefix `clothes/`. Nama-nama file diekstrak dari path lengkap menggunakan `blob.name.split("/")` sehingga hanya bagian nama file yang diperoleh tanpa prefix foldernya. Data ini kemudian dikemas dalam format JSON dengan status sukses dan dikembalikan sebagai respons.

```
@app.get("/")
async def fetch_all_images_from_firebase_cloth():
    try:
        bucket = storage.bucket()
        blobs = bucket.list_blobs(prefix="clothes/")
        return JsonResponse(content={"status":
"success", "data": [blob.name.split("/") [1] for blob
in blobs]})
```

5. Error Handling

Pada setiap operasi yang berpotensi gagal, blok `try-except` digunakan untuk menangani error. Jika terjadi kesalahan dalam proses pengunduhan file atau pemrosesan gambar, exception akan ditangkap dan diproses dalam blok `except`. Pesan error yang informatif kemudian dikembalikan dalam format JSON dengan status "error" dan pesan kesalahan yang sesuai. Respons ini disertai dengan kode status HTTP 400, menandakan bahwa permintaan tersebut gagal karena alasan tertentu pada sisi server atau aplikasi.

```
except Exception as e:
    return JsonResponse(
        content={"status": "error", "message":
str(e)},
        status_code=400
    )
```

5.3. Implementasi Hugging Face

Implementasi Hugging Face dilakukan melalui client Gradio API. Koneksi ini memungkinkan aplikasi mengirim gambar pengguna (person image) dan gambar pakaian (garment image) ke model di platform Hugging Face untuk diproses.

Penjelasan Alur kode:

1. Inisialisasi Client Gradio

Pada kode ini, dilakukan impor dependensi dan didefinisikan fungsi `process_images` untuk memproses dua gambar input: gambar tubuh pengguna (`person_image_path`) dan gambar pakaian (`garment_image_path`). File ini digunakan dalam model virtual try-on untuk menghasilkan gambar hasil dari pakaian yang dikenakan oleh pengguna. Untuk menghubungkan aplikasi dengan model Gradio, dibuat instansi client yang terhubung ke model Gradio yang dipublikasikan dengan nama `Nymbo/Virtual-Try-On`, memungkinkan aplikasi mengirimkan data gambar dan menerima hasil prediksi dari model tersebut.

```
from gradio_client import Client, handle_file

def process_images(person_image_path: str,
                  garment_image_path: str):
    client = Client("Nymbo/Virtual-Try-On")
```

2. Mempersiapkan Input untuk Prediksi

Input disiapkan dengan mengirimkan data ke API model menggunakan metode `client.predict()`. Gambar tubuh pengguna (background) diambil dari path `person_image_path` dan diproses dengan mengirimkan file yang telah di-handle. Parameter dict menyertakan input gambar tubuh, layers yang kosong dan composite yang diset ke `None` sebagai konfigurasi tambahan. Gambar pakaian diambil dari path `garment_image_path` dan diproses dengan cara yang sama menggunakan `handle_file()`.

Deskripsi pakaian diberikan dengan nilai “Tryon Test”, untuk menguji model. Jumlah langkah pengurangan noise pada gambar diatur dengan parameter `denoise_steps` untuk menghasilkan gambar yang lebih bersih, `seed` digunakan untuk memastikan prediksi yang bersifat deterministik menghasilkan hasil konsisten jika dieksekusi dengan `seed` yang sama. `api_name` merujuk pada endpoint API yang digunakan `"/tryon"` yang menandakan bahwa gambar tubuh dan pakaian akan digabungkan untuk melakukan virtual try-on.

```
result = client.predict(
    dict={"background":
handle_file(person_image_path), "layers": [],
"composite": None},
    garm_img=handle_file(garment_image_path),
    garment_des="Tryon Test",
    is_checked=True,
    is_checked_crop=False,
    denoise_steps=30,
    seed=42,
    api_name="/tryon"
)
```

3. Hasil Prediksi

Setelah model melakukan prediksi, hasilnya diambil dengan mengakses elemen pertama dan kedua dari objek `result`. `result[0]` berisi gambar hasil dari virtual try-on, `result[1]` berisi masked image yang menunjukkan bagian tubuh yang tertutup oleh pakaian. Fungsi `process_images` kemudian mengembalikan kedua gambar, `result_image` dan `masked_image` sebagai hasil dari proses virtual try-on.

```
result_image = result[0]
masked_image = result[1]

return result_image, masked_image
```

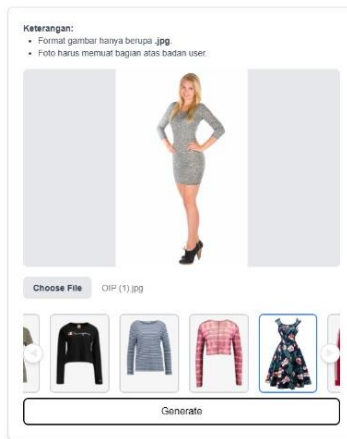

BAB VI

HASIL DAN PEMBAHASAN

6.1. Hasil Evaluasi

Pengujian aplikasi dilakukan dengan melibatkan 5 orang subjek uji yang dipilih secara acak. Subjek ini berasal dari berbagai latar belakang dengan rentang usia 18–35 tahun. Subjek tidak memerlukan pengetahuan teknis terkait aplikasi, sehingga pengujian mencerminkan perspektif pengguna umum. Pengujian dilakukan untuk memastikan bahwa semua fitur utama aplikasi berfungsi sesuai dengan tujuan. Fitur yang diuji meliputi:

1. Unggah gambar pengguna



Gambar 6.1.1 Tampilan Aplikasi Virtual Try-On: Pilih Gambar, Pakaian, dan Generate

Pada tahap ini subjek akan diberi arahan untuk melakukan unggah gambar pengguna yang memperlihatkan badan dan berformat .jpg dengan menekan tombol “Choose File”. Apabila format sesuai maka gambar akan terupload, namun jika gagal maka sistem akan memberikan alert “Format file tidak diterima. Pastikan formatnya berupa .jpg”.

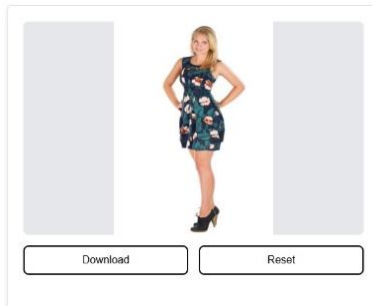
2. Pemilihan pakaian dari database

Pada tahap ini subjek akan diberikan daftar foto pakaian yang telah disediakan di database firebase. Pada dasarnya pelatihan model menggunakan dataset VTON-HD, namun penggunaan dataset custom juga tetap bisa diproses.

3. Proses rendering pakaian

Pengujian dilakukan untuk mengevaluasi kecepatan proses rendering gambar serta kesesuaian hasil gambar pakaian dengan tubuh pengguna. Subjek akan diarahkan untuk menekan tombol “generate” agar proses rendering berjalan.

4. Hasil akhir



Gambar 6.1.2 Tampilan Aplikasi Virtual Try-On: Hasil Berupa gambar pengguna yang telah mengenakan pakaian diproses untuk memastikan kualitas dan akurasi hasil akhir.

Tabel 6.1. Hasil Evaluasi Pengujian

Fitur	Skenario	Langkah dan Hasil	Catatan
Unggah Gambar	Gambar berformat .jpg	Langkah pengujian: 1. Klik tombol "Choose File". 2. Pilih file dengan format .jpg.	Format .jpg diterima dengan baik.

		<p>3. Klik "Upload".</p> <p>Hasil yang diharapkan: Gambar berhasil terunggah tanpa error.</p> <p>Hasil aktual: Gambar berhasil terunggah.</p>	
	Gambar berformat selain .jpg	<p>Langkah pengujian:</p> <ol style="list-style-type: none"> 1. Klik tombol "Choose File". 2. Pilih file dengan format .png atau lainnya. 3. Klik "Upload" <p>Hasil yang diharapkan: Muncul pesan alert "Format file tidak diterima. Pastikan formatnya berupa .jpg".</p> <p>Hasil aktual: Pesan alert muncul sesuai ekspektasi..</p>	Validasi format file bekerja dengan benar.
Pemilihan Pakaian	Data pakaian VTON-HD	<p>Langkah pengujian:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar pakaian. 2. Pilih salah satu pakaian. <p>Hasil yang diharapkan: Pakaian muncul dan dapat dipilih tanpa error.</p> <p>Hasil aktual: Pakaian muncul dan dapat dipilih.</p>	Data pakaian berhasil dimuat dari database Firebase.

	Data pakaian <i>custom</i>	<p>Langkah pengujian:</p> <ol style="list-style-type: none"> 1. Sistem menampilkan daftar pakaian. 2. Pilih salah satu pakaian. <p>Hasil yang diharapkan: Pakaian muncul dan dapat dipilih tanpa error.</p> <p>Hasil aktual: Pakaian muncul dan dapat dipilih.</p>	Data pakaian berhasil dimuat dari database Firebase.
Rendering	Rendering berjalan normal	<p>Langkah pengujian: Klik tombol "Generate".</p> <p>Hasil yang diharapkan: Proses rendering selesai dengan rata-rata waktu 50 detik.</p> <p>Hasil aktual: Proses selesai dalam 50 detik.</p>	Rendering berjalan sesuai desain.
Hasil Akhir	Hasil sesuai	<p>Langkah pengujian: Proses rendering selesai.</p> <p>Hasil yang diharapkan: Gambar menunjukkan pakaian sesuai tubuh pengguna.</p> <p>Hasil aktual: Gambar sesuai dengan tubuh pengguna.</p>	Hasil akhir sesuai dengan desain.

6.2. Pembahasan

Pengujian dilakukan untuk memastikan bahwa semua fitur utama aplikasi berjalan sesuai dengan desain dan tujuan awal. Hasil pengujian untuk setiap fitur dijelaskan sebagai berikut:

1. Unggah gambar pengguna

Fitur unggah gambar diuji dengan berbagai skenario, termasuk pengunggahan file dengan format yang sesuai (.jpg), format yang tidak didukung (.png atau lainnya), serta skenario di mana pengguna tidak memilih file sebelum menekan tombol "Upload".

- a. Hasil Positif: Semua subjek berhasil mengunggah gambar berformat .jpg tanpa kendala, menunjukkan bahwa sistem telah mampu menangani file dengan format yang sesuai.
- b. Hasil Negatif: Subjek yang mencoba mengunggah file dengan format selain .jpg menerima pesan alert yang sesuai, yaitu "Format file tidak diterima. Pastikan formatnya berupa .jpg". Begitu pula, jika file tidak dipilih, sistem memberikan pesan alert "Tidak ada file yang dipilih".
- c. Kesimpulan: Validasi format file bekerja dengan baik, menunjukkan ketahanan sistem dalam menangani input yang tidak sesuai. Namun, pengoptimalan kecepatan proses unggah untuk file berukuran besar dapat dipertimbangkan untuk pengalaman pengguna yang lebih baik.

2. Pemilihan pakaian dari database

Fitur pemilihan pakaian diuji menggunakan dua jenis data: pakaian berbasis dataset VTON-HD dan dataset custom yang telah ditambahkan ke Firebase. Hasilnya Semua subjek berhasil melihat daftar pakaian, baik dari dataset VTON-HD maupun custom, dan dapat memilih pakaian tanpa kesalahan.

3. Proses rendering pakaian

Proses rendering diuji untuk mengukur kecepatan dan hasil penggabungan gambar pakaian dengan gambar pengguna. Hasilnya proses rendering selesai dengan rata-rata waktu 50 detik, sesuai dengan desain awal. Hasilnya menunjukkan pakaian terlihat natural dan mengikuti tubuh pengguna.

4. Hasil akhir

Gambar akhir yang dihasilkan menunjukkan kualitas yang baik, memberikan hasil yang memuaskan bagi pengguna. Proses unggah gambar hasil ke Firebase berjalan lancar meski sedikit melambat pada koneksi yang kurang stabil. Tidak ada masalah signifikan yang ditemukan terkait hasil akhir ini, dan akurasi pakaian yang dikenakan sesuai dengan desain yang dipilih pengguna.

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Berdasarkan hasil implementasi dan evaluasi terhadap aplikasi virtual try-on, dapat disimpulkan bahwa:

1. Perancangan Arsitektur Aplikasi Web Virtual Try-On

Aplikasi web virtual try-on berhasil dirancang menggunakan framework Next.js, yang memberikan performa cepat dan kemudahan pengembangan. Arsitektur aplikasi dirancang dengan mempertimbangkan modularitas dan pengalaman pengguna (user experience), sehingga pengguna dapat mengoperasikan aplikasi dengan mudah tanpa memerlukan pengetahuan teknis.

2. Integrasi Firebase dan Hugging Face

Integrasi Firebase berhasil dilakukan untuk menyimpan data pakaian dan hasil gambar akhir. Selain itu, Hugging Face digunakan sebagai platform pemrosesan gambar dengan model VTON-HD, yang mampu menghasilkan hasil rendering pakaian sesuai tubuh pengguna dengan tingkat akurasi tinggi. Kedua teknologi ini terintegrasi dengan baik, memungkinkan proses pengambilan data, rendering, dan penyimpanan berjalan lancar.

3. Evaluasi Aplikasi Web Virtual Try-On

Evaluasi terhadap fitur utama aplikasi menunjukkan hasil yang memuaskan. Pada fitur unggah gambar, validasi file berjalan sesuai dengan desain, di mana file berformat .jpg berhasil diunggah tanpa kendala, sementara file dengan format yang tidak sesuai memunculkan pesan alert yang relevan. Fitur pemilihan pakaian juga menunjukkan kinerja yang optimal, dengan data pakaian berhasil dimuat dari Firebase, baik menggunakan dataset default VTON-HD maupun dataset custom, serta waktu respons yang memadai. Proses rendering pakaian berlangsung dengan rata-rata waktu

50 detik dan menghasilkan gambar dengan tingkat akurasi yang tinggi, meskipun terdapat kesalahan minor pada pose tubuh yang kompleks. Hasil akhir menunjukkan kualitas gambar yang baik, dengan pakaian yang sesuai dengan desain pilihan pengguna, sehingga memberikan pengalaman yang memuaskan.

7.2. Saran

Meskipun sistem telah berfungsi dengan baik, beberapa perbaikan dapat dilakukan untuk meningkatkan kualitas dan kenyamanan pengguna. Salah satunya adalah mempercepat proses rendering gambar, dengan mengoptimalkan algoritma yang digunakan dalam pemrosesan gambar. Peningkatan pada akurasi penyesuaian pakaian pada pose tubuh yang lebih kompleks juga diperlukan untuk menghindari kesalahan rendering pada posisi tertentu. Selain itu, pengujian lebih lanjut dengan file gambar yang lebih beragam, baik dari sisi ukuran maupun kualitas gambar, akan membantu dalam meminimalisir masalah yang mungkin terjadi. Fitur tambahan seperti penyesuaian warna dan ukuran pakaian sesuai preferensi pengguna dapat meningkatkan tingkat kepuasan dan personalisasi dalam aplikasi. Ke depan, integrasi dengan platform e-commerce dan penyempurnaan sistem backend dapat memperluas cakupan aplikasi ini untuk digunakan dalam skala yang lebih luas.

DAFTAR PUSTAKA

- [1] C. Chen, J. Ni, and P. Zhang, “Virtual try-on systems in fashion consumption: A systematic review,” *Applied Sciences*, vol. 14, no. 24, p. 11839, Dec. 2024. DOI: 10.3390/app142411839.
- [2] R. Roy and S. Ramakrishnan, “Embracing the future of retail with virtual try-on technology,” in *Data-Driven Intelligent Business Sustainability*, pp. 344-359, Dec. 2023. DOI: 10.4018/979-8-3693-0049-7.ch023.
- [3] “Sejarah,” Departemen Teknik Informatika. <https://www.its.ac.id/informatika/id/tentang-kami/sejarah/>
- [4] “Visi dan Misi,” Departemen Teknik Informatika. <https://www.its.ac.id/informatika/id/tentang-kami/visi-dan-misi/>
- [5] “Struktur Organisasi,” Departemen Teknik Informatika. <https://www.its.ac.id/informatika/id/dosen-staff/struktur-organisasi/>
- [6] “Laboratorium,” Departemen Teknik Informatika. <https://www.its.ac.id/informatika/id/fasilitas/laboratorium/>
- [7] D. Marelli, S. Bianco, and G. Ciocca, “Designing an AI-Based Virtual Try-On Web Application,” *MDPI*, vol. 22, no. 10, p. 3832, May 2022. DOI: 10.3390/s22103832.
- [8] “Documentation,” Next.js. <https://nextjs.org/docs>
- [9] L. Moroney, *The Firebase Realtime Database*, in *The Definitive Guide to Firebase*, 1st ed. New York, NY, USA: Apress, 2017, pp. 51-71. doi: 10.1007/978-1-4842-2943-9_3.
- [10] “Cloud Functions for Firebase,” Firebase. <https://firebase.google.com>
- [11] Goutham L, “RAOS Virtual Try-On Model,” Hugging Face. <https://huggingface.co/gouthaml/raos-virtual-try-on-model>
- [12] Yisol, “IDM-VTON: Improving Diffusion Models for Authentic Virtual Try-on in the Wild,” Hugging Face. <https://huggingface.co/yisol/IDM-VTON>

- [13] “Firebase Storage,” Firebase. <https://firebase.google.com/>
- [14] G. Manfredi, G. Gilio, V. Baldi, H. Youssef, and U. Erra, “VICO-DR: A Collaborative Virtual Dressing Room for Image Consulting,” *J. Imaging*, vol. 9, no. 4, p. 76, Mar. 2023. DOI: 10.3390/jimaging9040076.

BIODATA PENULIS I

Nama : Widian Sasi Disertasiani
Tempat, Tanggal Lahir : Jambi, 23 Agustus 2003
Jenis Kelamin : Perempuan
Telepon : +6285283426511
Email : 5025211024@student.its.ac.id

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2021
Semester : 7 (Tujuh)

BIODATA PENULIS II

Nama : Hana Maheswari
Tempat, Tanggal Lahir : Gresik, 7 November 2003
Jenis Kelamin : Perempuan
Telepon : +62895366243899
Email : 5025211182@student.its.ac.id

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2021
Semester : 7 (Tujuh)

[Halaman ini sengaja dikosongkan]