SEMINAR NASIONAL 2026 Universitas Ma Chung
Prosiding Seminar Nasional Universitas Ma Chung 2026
(Sistem Informasi dan Informatika)
pp. 295-305
E-ISSN: 2808-7771

# SEGMENTASI SEMANTIK ELEMEN *UI/UX* UNTUK EVALUASI AKSESIBILITAS DESAIN *DIGITAL*

**Syahtria Akbaruari[1], Dwi Wahyu Prabowo[2], Nurahman[3]**

[1]Sistem Informasi, Universitas Darwan Ali, Jalan Batu Berlian No.10, Kotawaringin Timur, Indonesia, 74323

***Email korespondensi:** syahtriabelajar@gmail.com

*Received: 27 05 2026 – Revised: 29 07 2026 - Accepted: 04 08 2026 - Published: 05 08 2026*

**Abstrak.** Aksesibilitas antarmuka pengguna (*User Interface/UI*) merupakan aspek penting dalam pengembangan platform mobile dan web untuk memastikan seluruh pengguna, termasuk penyandang disabilitas, dapat mengakses layanan digital secara setara. Namun, proses evaluasi aksesibilitas yang masih dilakukan secara manual umumnya membutuhkan waktu yang lama, biaya yang tinggi, serta berpotensi menghasilkan penilaian yang bersifat subjektif. Penelitian ini mengusulkan pendekatan otomatis berbasis *deep learning* menggunakan DeepLabV3+ dan backbone MobileNetV3-Large untuk melakukan segmentasi semantik lima kelas elemen UI, yaitu *Background, Teks, Tombol, Gambar/Ikon,* dan *Navigasi*. Dataset yang digunakan terdiri atas RICO yang memuat 10.000 citra antarmuka Android dan WebSRC yang berisi 1.000 citra antarmuka web. *Pseudo-label* dibangun secara otomatis melalui proses *parsing view hierarchy* berbasis JSON pada dataset RICO serta pendekatan heuristik berbasis *computer vision* pada dataset WebSRC. Pelatihan model menerapkan strategi *progressive unfreezing backbone* yang dikombinasikan dengan mekanisme *reset counter* pada *early stopping* guna meningkatkan stabilitas dan efektivitas proses pembelajaran model. Evaluasi performa dilakukan menggunakan *Mean Intersection over Union* (mIoU), *Pixel Accuracy*, dan *Dice Coefficient*. Hasil terbaik diperoleh pada epoch ke-83 dengan nilai mIoU sebesar 0,3537, *Pixel Accuracy* sebesar 0,6523, dan *Mean Dice* sebesar 0,4800. Kelas *Navigasi* menghasilkan nilai *Intersection over Union* tertinggi, yaitu 0,5652, sedangkan kelas *Tombol* mengalami peningkatan performa dari 0,1666 menjadi 0,2044 atau meningkat sebesar 22,7% dibandingkan model dasar (*baseline*). Hasil penelitian ini menunjukkan bahwa model yang dikembangkan berpotensi menjadi fondasi awal dalam membangun sistem evaluasi aksesibilitas antarmuka pengguna yang lebih otomatis, konsisten, dan efisien di masa mendatang.

**Kata kunci:** Segmentasi Semantik, Aksesibilitas UI, *Deep Learning, Human-Centered Design,* Evaluasi Otomatis

## PENDAHULUAN

Perkembangan teknologi digital telah meningkatkan penggunaan aplikasi *mobile* dan *web* pada berbagai sektor, seperti pendidikan, layanan publik, perdagangan elektronik, dan komunikasi. Kondisi tersebut menuntut perancangan antarmuka pengguna (*User Interface/UI*) yang tidak hanya berorientasi pada estetika, tetapi juga mampu menjamin kemudahan akses bagi seluruh kelompok pengguna. Aksesibilitas digital menjadi salah satu aspek penting dalam pengembangan sistem karena berperan dalam memastikan bahwa layanan digital dapat digunakan secara setara oleh pengguna dengan berbagai kondisi fisik, sensorik, maupun kognitif. Menurut *World Health Organization* (WHO), sekitar 1,3 miliar

orang atau 16% populasi dunia hidup dengan suatu bentuk disabilitas, sehingga penerapan aksesibilitas digital menjadi kebutuhan yang semakin mendesak (WHO, 2023).

Untuk mendukung implementasi aksesibilitas digital, *World Wide Web Consortium* (W3C) melalui *Web Content Accessibility Guidelines* (WCAG) 2.1 menyediakan pedoman internasional yang berlandaskan empat prinsip utama, yaitu *perceivable, operable, understandable,* dan *robust*, sebagai acuan dalam merancang antarmuka digital yang inklusif (World Wide Web Consortium, 2018). Meskipun demikian, implementasi aksesibilitas pada antarmuka digital masih menghadapi berbagai tantangan. Laporan The WebAIM Million 2023 menunjukkan bahwa sebanyak 96,3% halaman web pada satu juta situs teratas masih memiliki setidaknya satu kesalahan aksesibilitas yang terdeteksi secara otomatis (WebAIM, 2023).Selain itu, Vigo, dkk. menjelaskan bahwa evaluasi aksesibilitas yang hanya mengandalkan pengujian manual maupun alat berbasis aturan (*rule-based tools*) belum mampu mendeteksi seluruh permasalahan aksesibilitas secara konsisten (Vigo et al., 2013). Keterbatasan tersebut menunjukkan perlunya pendekatan yang lebih adaptif dan otomatis untuk mendukung proses evaluasi aksesibilitas antarmuka pengguna.

Salah satu pendekatan yang berkembang pesat adalah pemanfaatan *deep learning* pada bidang *computer vision*. Teknik segmentasi semantik (*semantic segmentation*) memungkinkan sistem memberikan label pada setiap piksel citra sehingga mampu mengidentifikasi lokasi, bentuk, dan batas objek secara rinci. Menurut, Lateef & Ruichek. segmentasi semantik telah banyak diterapkan pada berbagai bidang, seperti kendaraan otonom, robotika, dan analisis citra medis karena kemampuannya dalam memahami struktur visual suatu adegan (Lateef & Ruichek, 2019). Dalam konteks antarmuka digital, teknik ini berpotensi digunakan untuk mengidentifikasi elemen-elemen UI secara otomatis dari sebuah *screenshot*, sehingga dapat mendukung proses evaluasi aksesibilitas yang lebih objektif dan efisien.

Penelitian mengenai pemahaman antarmuka pengguna (*UI understanding*) juga mengalami perkembangan yang signifikan. Sunkara, dkk. memperluas dataset RICO dengan lebih dari 500 ribu anotasi semantik untuk meningkatkan kemampuan model dalam memahami hubungan antar elemen UI, seperti ikon, teks, dan label fungsional (Sunkara et al., 2022). Selain itu, Zhang, dkk. mengembangkan sistem *screen recognition* yang mampu menghasilkan metadata aksesibilitas secara otomatis dari citra antarmuka aplikasi *mobile* untuk mendukung teknologi pembaca layar (*screen reader*) (Zhang et al., 2021). Temuan

tersebut menunjukkan bahwa informasi visual yang terkandung dalam antarmuka digital dapat dimanfaatkan untuk mendukung pengembangan sistem aksesibilitas yang lebih cerdas. Sayangnya, sebagian besar penelitian sebelumnya berfokus pada deteksi komponen UI dan pembangkitan metadata aksesibilitas, sedangkan pemanfaatan segmentasi semantik untuk mendukung evaluasi aksesibilitas *UI/UX* secara otomatis masih relatif jarang, terutama pada penelitian yang mengombinasikan dataset aplikasi *mobile* dan antarmuka *web*. Kondisi tersebut menunjukkan adanya peluang penelitian untuk mengembangkan pendekatan yang lebih umum (*generalizable*) dan efisien.

Salah satu arsitektur yang banyak digunakan pada tugas segmentasi semantik adalah DeepLabV3+. Arsitektur ini mengintegrasikan mekanisme *Atrous Spatial Pyramid Pooling* (ASPP) dan struktur *encoder-decoder* untuk menangkap konteks multi-skala sekaligus mempertahankan detail batas objek (Chen et al., 2018). Untuk meningkatkan efisiensi komputasi, DeepLabV3+ dapat dipadukan dengan backbone MobileNetV3-Large yang dirancang untuk menghasilkan keseimbangan antara akurasi dan kecepatan inferensi pada perangkat dengan sumber daya terbatas (Howard et al., 2019). Kombinasi kedua arsitektur tersebut relevan diterapkan pada segmentasi elemen UI karena mampu mempertahankan performa yang baik dengan kebutuhan komputasi yang relatif rendah.

Penelitian ini memanfaatkan dataset publik RICO yang berisi ribuan *screenshot* antarmuka aplikasi *Android* beserta struktur *view hierarchy* yang menyertainya (Deka et al., 2017), serta dataset WebSRC yang merepresentasikan antarmuka berbasis web. Penelitian ini bertujuan membangun model segmentasi semantik untuk mengidentifikasi lima kelas elemen UI, yaitu *Background, Teks, Tombol, Gambar/Ikon,* dan *Navigasi*. Selain itu, penelitian ini mengimplementasikan strategi *progressive unfreezing backbone* yang dikombinasikan dengan mekanisme *reset counter* pada *early stopping* guna meningkatkan efektivitas proses pelatihan model. Hasil penelitian diharapkan dapat menjadi fondasi awal bagi pengembangan sistem evaluasi aksesibilitas antarmuka pengguna yang lebih otomatis, konsisten, dan efisien.

## MASALAH

Aksesibilitas antarmuka pengguna (*User Interface/UI*) merupakan salah satu aspek penting dalam pendekatan *human-centered design* yang bertujuan memastikan sistem digital dapat digunakan oleh seluruh kelompok pengguna, termasuk penyandang disabilitas

(International Organization for Standardization, 2019). Meskipun berbagai penelitian telah memanfaatkan pendekatan berbasis *computer vision* untuk memahami struktur antarmuka pengguna, sebagian besar penelitian masih berfokus pada deteksi komponen UI dan pembangkitan metadata aksesibilitas, sementara pemanfaatan segmentasi semantik untuk mendukung evaluasi aksesibilitas UI/UX secara otomatis masih relatif terbatas.

Penelitian yang dilakukan oleh Zhang, dkk. menunjukkan bahwa informasi visual pada antarmuka aplikasi mobile dapat dimanfaatkan untuk menghasilkan metadata aksesibilitas secara otomatis guna mendukung teknologi *screen reader (Zhang et al., 2021)*. Selain itu, Deka, dkk. memperkenalkan dataset RICO yang menyediakan ribuan *screenshot* aplikasi Android beserta struktur *view hierarchy* yang dapat dimanfaatkan untuk berbagai tugas pemahaman antarmuka pengguna (Deka et al., 2017). Namun, penelitian-penelitian tersebut belum secara khusus mengintegrasikan segmentasi semantik sebagai dasar evaluasi aksesibilitas UI/UX pada platform mobile dan web secara bersamaan. Kondisi tersebut menunjukkan adanya kebutuhan untuk mengembangkan pendekatan yang lebih umum (*generalizable*), otomatis, dan efisien untuk mendukung proses evaluasi aksesibilitas antarmuka pengguna.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan penerapan segmentasi semantik berbasis *deep learning* menggunakan arsitektur DeepLabV3+ dengan backbone MobileNetV3-Large untuk mengidentifikasi lima kategori elemen UI, yaitu *Background, Teks, Tombol, Gambar/Ikon,* dan *Navigasi*. Pendekatan ini diharapkan dapat menjadi fondasi awal dalam pengembangan sistem evaluasi aksesibilitas antarmuka pengguna yang lebih objektif, konsisten, dan terotomatisasi.

## METODE PELAKSANAAN

A. Dataset dan Persiapan Data

Penelitian ini menggunakan dua dataset publik, yaitu RICO dan WebSRC. Dataset RICO diperkenalkan oleh Deka, dkk. dan terdiri atas sekitar 10.000 *screenshot* antarmuka aplikasi Android beserta berkas *view hierarchy* dalam format JSON yang mendeskripsikan struktur dan atribut setiap elemen *User Interface* (UI) (Deka et al., 2017). Selain itu, penelitian ini memanfaatkan dataset WebSRC yang berisi sekitar 1.000 *screenshot* antarmuka web yang diperoleh melalui *streaming dataset* HuggingFaceM4/WebSight untuk memperluas keragaman data antarmuka berbasis web.

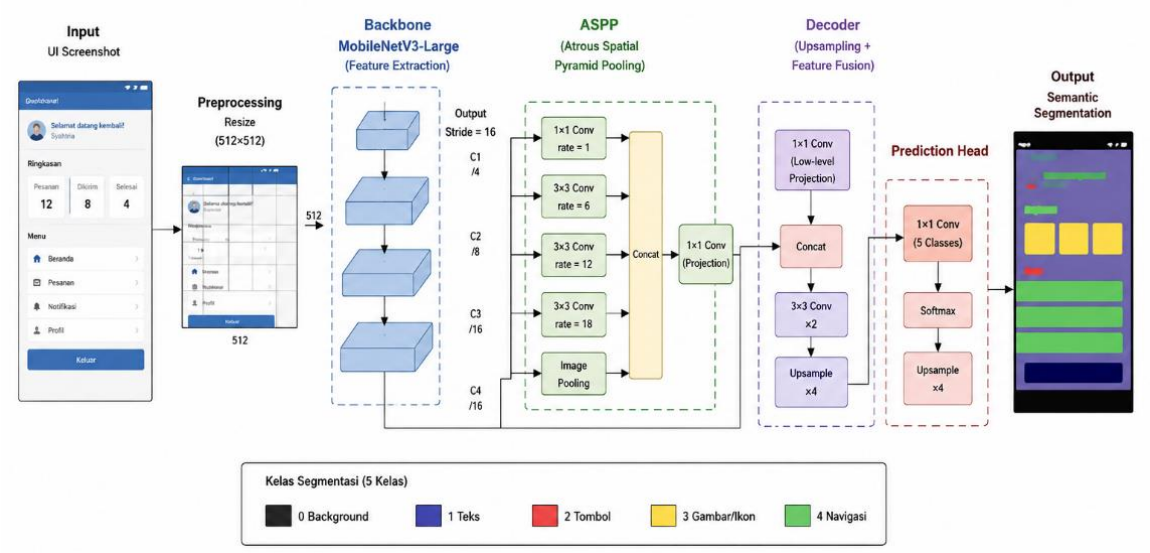
Pada dataset RICO, *pseudo-label* segmentasi dihasilkan dengan melakukan *parsing* terhadap berkas *view hierarchy* menggunakan pemetaan kelas *Android widget* ke dalam lima kelas segmentasi. Kelas *Background* (0) ditetapkan secara implisit, kelas *Teks* (1) dipetakan dari android.widget.TextView dan android.widget.EditText, kelas *Tombol* (2) dipetakan dari android.widget.Button dan turunannya, kelas *Gambar/Ikon* (3) dipetakan dari android.widget.ImageView, sedangkan kelas *Navigasi* (4) dipetakan dari android.widget.ListView, RecyclerView, dan komponen navigasi lainnya. Seluruh citra dan *mask* kemudian diubah ukurannya menjadi resolusi 512×512 piksel. Dataset WebSRC, *pseudo-label* menggunakan kombinasi deteksi tepi (*Canny edge detection*), analisis saturasi HSV, dan deteksi kontur untuk mengidentifikasi area teks, tombol, dan gambar. Seluruh dataset yang telah diproses menghasilkan 11.000 sampel yang disimpan dalam format NumPy Compressed (NPZ). Selanjutnya, dataset dibagi secara acak dengan rasio 80% untuk data pelatihan, 10% untuk data validasi, dan 10% untuk data pengujian.

B. Arsitektur Model

Penelitian ini menggunakan arsitektur DeepLabV3+ dengan backbone MobileNetV3-Large untuk melakukan segmentasi semantik elemen antarmuka pengguna (User Interface/UI). Arsitektur DeepLabV3+ dipilih karena memiliki kemampuan yang baik dalam menangkap informasi kontekstual pada berbagai skala melalui mekanisme Atrous Spatial Pyramid Pooling (ASPP). Selain itu, struktur encoder-decoder yang dimiliki DeepLabV3+ memungkinkan model mempertahankan detail batas objek sehingga hasil segmentasi menjadi lebih presisi (Chen et al., 2018). Kemampuan tersebut sangat penting dalam proses identifikasi elemen UI yang memiliki ukuran, bentuk, dan posisi yang beragam pada antarmuka digital. Penggunaan pendekatan segmentasi semantik memungkinkan setiap piksel pada citra antarmuka diklasifikasikan ke dalam kategori elemen yang sesuai.

Sementara itu, MobileNetV3-Large digunakan sebagai backbone karena dirancang untuk menghasilkan keseimbangan antara akurasi dan efisiensi komputasi pada perangkat dengan sumber daya terbatas (Howard et al., 2019). Backbone ini berperan dalam mengekstraksi fitur visual penting dari citra masukan sebelum diproses lebih lanjut oleh modul segmentasi. Sebagaimana ditunjukkan pada Gambar 1, citra masukan berukuran 512×512 piksel terlebih dahulu diproses oleh MobileNetV3-Large untuk mengekstraksi fitur visual multi-level. Fitur yang dihasilkan kemudian diteruskan ke modul ASPP untuk memperoleh representasi kontekstual pada berbagai skala. Selanjutnya, fitur tersebut

diproses oleh decoder melalui mekanisme upsampling dan feature fusion sebelum diproyeksikan ke lapisan prediksi menggunakan konvolusi 1×1. Model menghasilkan prediksi segmentasi terhadap lima kelas elemen UI, yaitu Background, Teks, Tombol, Gambar/Ikon, dan Navigasi.




**Gambar 1.** Arsitektur DeepLabV3+ dengan *backbone* MobileNetV3-Large untuk segmentasi semantik elemen UI.

Sebagaimana ditunjukkan pada **Gambar 1**, citra masukan (*input screenshot*) berukuran 512×512 piksel terlebih dahulu melalui tahap *preprocessing* berupa *resize* sebelum diproses oleh backbone MobileNetV3-Large untuk mengekstraksi fitur visual multi-level. Fitur yang dihasilkan kemudian diteruskan ke modul *Atrous Spatial Pyramid Pooling* (ASPP) untuk memperoleh representasi kontekstual pada berbagai skala. Selanjutnya, fitur diproses oleh *decoder* melalui mekanisme *upsampling* dan *feature fusion* sebelum diproyeksikan ke lapisan prediksi menggunakan konvolusi 1×1. Model menghasilkan segmentasi terhadap lima kategori elemen UI, yaitu *Background, Teks, Tombol, Gambar/Ikon,* dan *Navigasi*.

Rincian konfigurasi arsitektur yang digunakan pada penelitian ini ditunjukkan pada **Tabel 1**. Penggunaan MobileNetV3-Large sebagai backbone bertujuan untuk mengurangi kompleksitas komputasi tanpa mengorbankan performa segmentasi secara signifikan. Selain itu, modul ASPP memungkinkan model memperoleh informasi kontekstual multi-skala sehingga dapat meningkatkan kemampuan identifikasi elemen UI yang memiliki ukuran dan

posisi yang beragam. Kombinasi tersebut menjadikan model sesuai diterapkan pada lingkungan komputasi terbatas, seperti Google Colab dengan GPU NVIDIA Tesla T4.

**Tabel 1.** Spesifikasi Arsitektur DeepLabV3+ dengan Backbone MobileNetV3-Large

| Komponen | Konfigurasi |
|---|---|
| *input* | *Screenshot* UI 512×512 piksel |
| *Backbone* | MobileNetV3-Large |
| *Feature Extraction* | *Multi-level feature extraction* |
| ASPP | *Atrous Spatial Pyramid Pooling* |
| *Decoder* | *Upsampling* dan *feature fusion* |
| *Prediction Head* | Konvolusi 1×1 |
| Aktivasi *Output* | *Softmax* |
| Jumlah Kelas | 5 kelas |
| Kelas Segmentasi | *Background*, Teks, Tombol, Gambar/Ikon, Navigasi |
| Bobot Awal | *ImageNet pre-trained* |

C. Strategi Pelatihan

Pelatihan model dilakukan menggunakan strategi *progressive unfreezing* yang terdiri atas dua fase. Pada fase pertama (*Frozen Backbone*, *epoch* 1–14), hanya lapisan *classifier* yang dilatih menggunakan *learning rate* sebesar $1\times10^{-4}$, sedangkan seluruh parameter *backbone* dibekukan (*freeze*). Pada fase kedua (*Full Fine-tuning*, epoch 15 dan seterusnya), seluruh lapisan backbone dibuka kembali (*unfreeze*) dengan menerapkan *differential learning rate*, yaitu $1\times10^{-5}$ untuk *backbone* dan $1\times10^{-4}$ untuk *classifier*.

Penelitian ini juga mengimplementasikan mekanisme *reset counter* pada *early stopping* ketika proses *unfreeze* dilakukan. Strategi tersebut bertujuan memberikan kesempatan tambahan bagi model untuk beradaptasi terhadap perubahan konfigurasi optimizer, sehingga dapat mengurangi risiko penghentian pelatihan secara prematur.Fungsi loss yang digunakan adalah *CrossEntropyLoss* dengan class weights [0,2; 1,8; 2,0; 1,8; 1,0] untuk kelas [*Background*; Teks; Tombol; Gambar; Navigasi]. *Optimizer* yang digunakan adalah AdamW dengan weight decay $1\times10^{-4}$, dan *learning rate* dijadwalkan menggunakan *CosineAnnealingLR*. *Batch size* yang digunakan adalah 8 dengan *mixed precision training* (AMP). *Early stopping* dengan *patience*=10 diterapkan berdasarkan nilai validasi mIoU.

Untuk meningkatkan kemampuan generalisasi model, diterapkan augmentasi data berupa *horizontal flip* (p = 0,5), *color jitter* (brightness=0,2; contrast=0,2; saturation=0,2; p=0,5), dan *Gaussian noise* (p=0,2). Tingkat augmentasi dijaga dalam batas moderat untuk menghindari distorsi bentuk persegi panjang (*rectangular shape*) yang merupakan karakteristik utama elemen UI.

D. Metrik Evaluasi

Evaluasi model dilakukan pada data pengujian menggunakan tiga metrik utama, yaitu *Mean Intersection over Union* (mIoU), *Pixel Accuracy*, dan *Dice Coefficient*.

1. ***Mean Intersection over Union (mIoU)*** digunakan untuk mengukur tingkat kesesuaian antara hasil prediksi dan *ground truth* pada setiap kelas segmentasi.

$$mIoU = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i + FN_i} \quad (1)$$

2. ***Pixel Accuracy*** digunakan untuk menghitung proporsi piksel yang berhasil diklasifikasikan dengan benar.

$$Pixel\ Accuracy = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FN_i)} \quad (2)$$

3. ***Dice Coefficient*** digunakan untuk mengukur tingkat kemiripan antara hasil segmentasi dan *ground truth*.

$$Dice = \frac{2TP}{2TP + FP + FN} \quad (3)$$

Pada persamaan (1), (2), dan (3), $TP$ (*True Positive*) menyatakan jumlah piksel yang berhasil diprediksi dengan benar sesuai kelas sebenarnya. $FP$ (*False Positive*) menyatakan jumlah piksel yang diprediksi sebagai suatu kelas, tetapi sebenarnya termasuk ke dalam kelas lain. $FN$ (*False Negative*) menyatakan jumlah piksel yang seharusnya termasuk ke dalam suatu kelas, tetapi gagal diidentifikasi oleh model. Sementara itu, $N$ menyatakan jumlah kelas segmentasi yang digunakan pada penelitian ini, yaitu lima kelas yang terdiri atas *Background, Teks, Tombol, Gambar/Ikon,* dan *Navigasi*.

## HASIL DAN PEMBAHASAN

### A. Performa Model Segmentasi

Model terbaik diperoleh pada *epoch* ke-83 dengan nilai validasi mIoU sebesar 0,3629. Setelah dievaluasi pada test set yang independen, model mencapai metrik sebagaimana ditunjukkan pada Tabel 2.

**Tabel 2.** Hasil Evaluasi Model pada Test Set

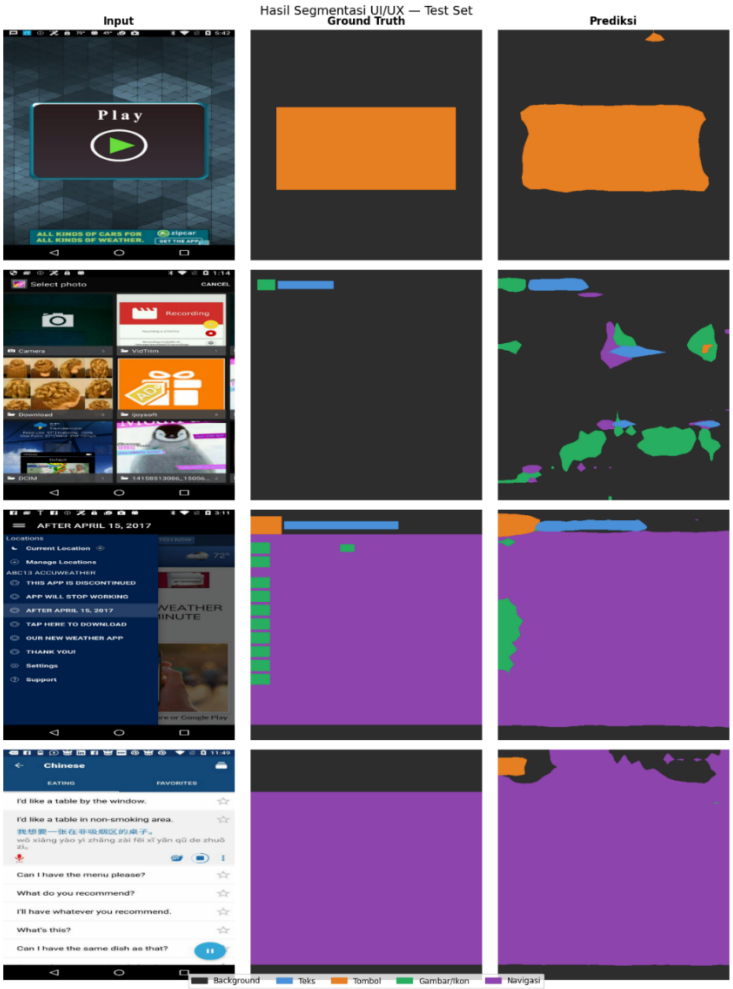
| Metrik | Nilai |
|---|---|
| *Pixel Accuracy* | 200 |
| *Mean IoU (mIoU* | 400 |
| *Mean Dice Coefficient* | 550 |

Hasil evaluasi per kelas ditunjukkan pada Tabel 3. Kelas Navigasi memperoleh nilai IoU tertinggi (0,5652) karena konsistensi posisi dan visual pada antarmuka *mobile*. Kelas *Background* juga memperoleh nilai yang baik (0,4965) karena mendominasi area layar secara visual.

**Tabel 3.** Hasil IoU dan Dice per Kelas pada Test Set

| Kelas | IoU | *Dice Coefficient* |
|---|---|---|
| *Background* | 0,4965 | 0,6511 |
| Teks | 0,2586 | 0,3886 |
| Tombol | 0,2044 | 0,3147 |
| Gambar/Ikon | 0,2441 | 0,3418 |
| Navigasi | 0,5652 | 0,7037 |
| *Mean* | 0,3538 | 0,4800 |
| Background | 0,4965 | 0,6511 |

Kelas Tombol memperoleh IoU terendah (0,2044), namun hal ini merupakan peningkatan signifikan sebesar 22,7% dibandingkan *baseline* V1 (0,1666). Tombol merupakan kelas yang paling menantang karena variasi bentuk, ukuran, dan gaya visual yang sangat beragam antara aplikasi *mobile* dan *web*, ditambah keterbatasan akurasi *pseudo-label* otomatis untuk elemen ini. Visualisasi hasil prediksi model pada empat sampel test set ditunjukkan pada Gambar 2.




**Gambar 2.** Hasil segmentasi model V4 (Final) pada empat sampel test set. Kolom kiri: input asli; kolom tengah: ground truth; kolom kanan: hasil prediksi model 002E.l

Sebagaimana ditunjukkan pada Gambar 2, model mampu mengidentifikasi beberapa elemen antarmuka pengguna pada empat sampel *test set* dengan tingkat akurasi yang bervariasi pada setiap kelas segmentasi. Hasil prediksi menunjukkan bahwa kelas *Navigasi* dan *Tombol* cenderung terdeteksi dengan lebih baik dibandingkan kelas lainnya, sedangkan beberapa kesalahan segmentasi masih ditemukan pada kelas *Teks* dan *Gambar/Ikon* yang memiliki ukuran relatif kecil dan distribusi yang tidak merata. Secara keseluruhan, model telah mampu menangkap struktur utama antarmuka pengguna sehingga berpotensi mendukung proses evaluasi aksesibilitas UI/UX secara otomatis.

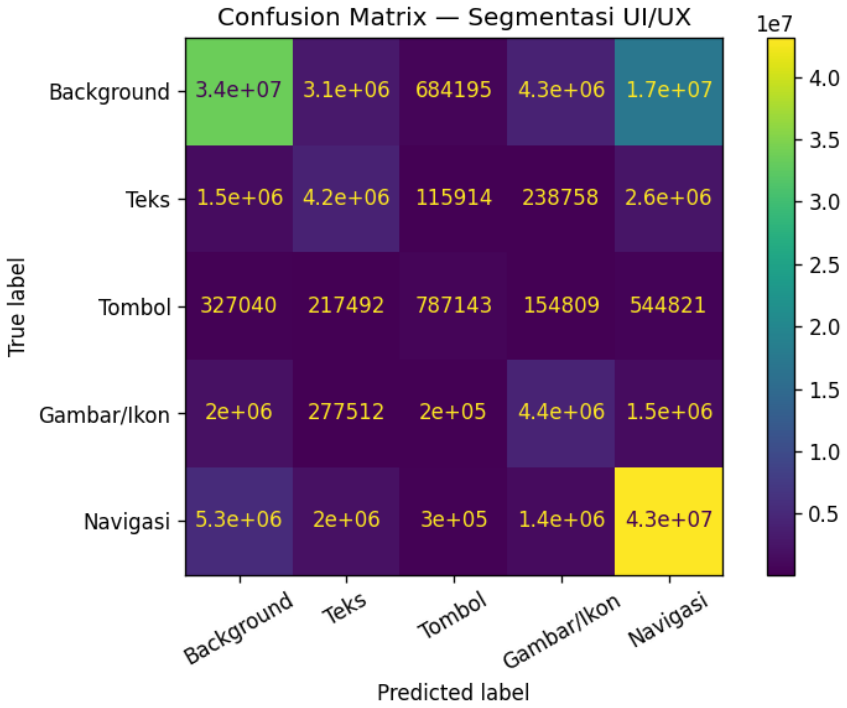
### B. Analisis Komparatif Strategi Pelatihan

Penelitian ini melibatkan empat iterasi pengembangan model untuk menemukan strategi pelatihan yang optimal, sebagaimana ditunjukkan pada Tabel 4.

**Tabel 4.** Perbandingan Performa Antar Versi Model

| Versi | *Class Weights* Tombol | Augmentasi | *Unfreeze* | *Es Reset* | mIoU (Test) |
|---|---|---|---|---|---|
| V1 (*Baseline*) | 1,5 | *Moderate* | Ep. 30 | Tidak | 0,3129 |
| V2 | 4,0 (Ekstrem) | *Agresif* | Ep. 25 | Tidak | 0,2752 |
| V3 | 2,0 | *Moderate* | Ep. 30 | Tidak | 0,2853 |
| V4 (*Final*) | 2,0 | *Moderate* | Ep. 15 | Ya | 0,3537 |

Berdasarkan Tabel 4, performa terbaik diperoleh pada model V4 (Final) dengan nilai mIoU sebesar 0,3537. Peningkatan performa tersebut dicapai melalui penggunaan class weights Tombol sebesar 2,0, augmentasi moderat, percepatan proses unfreeze pada epoch ke-15, serta penerapan mekanisme early stopping reset. Sebaliknya, model V2 yang menggunakan class weights ekstrem sebesar 4,0 dan augmentasi agresif menghasilkan performa terendah dengan mIoU sebesar 0,2752.




**Gambar 3.** *Confusion matrix* hasil prediksi model V4 (*Final*) pada test set. Nilai diagonal menunjukkan prediksi benar per kelas.

Beberapa temuan penting dari analisis komparatif ditunjukkan pada Gambar 3. Pertama, V2 dengan *class weights* ekstrem dan augmentasi agresif menghasilkan penurunan

performa signifikan (−12% dari baseline). Augmentasi geometrik seperti *ElasticTransform* dan *GridDistortion* terbukti kontraproduktif untuk dataset UI yang didominasi elemen berbentuk persegi panjang lurus.

Kedua, V3 dengan pengaturan moderat tetap menghasilkan performa di bawah *baseline* karena early stopping terpicu sebelum unfreeze *backbone* tercapai (berhenti di *epoch* 25, sementara *unfreeze* dijadwalkan di *epoch* 30). Hal ini mengidentifikasi *bug scheduling* yang kritis: ketika nilai *PATIENCE* lebih kecil dari *UNFREEZE_EPOCH*, *backbone* tidak pernah diaktifkan sama sekali.

Ketiga, V4 memperbaiki bug tersebut dengan dua perubahan: (a) memajukan jadwal *unfreeze* ke epoch 15 agar bertepatan dengan plateau performa *classifier-only*; dan (b) mereset *counter early stopping* menjadi nol saat *unfreeze* dilakukan. Perubahan ini memberikan model kesempatan penuh dengan 10 *epoch* segar untuk mengeksplorasi ruang parameter setelah backbone diaktifkan. Hasilnya, V4 berhasil melampaui semua versi sebelumnya dengan mIoU 0,3537, meningkat 13% dari baseline V1.

### C. Kurva Konvergensi Training V4

Analisis kurva pelatihan V4 menunjukkan tiga fase yang jelas. Fase pertama (*Epoch* 1–14, *frozen backbone*): mIoU validasi naik dari 0,2024 ke 0,2693 secara stabil. Fase kedua (*Epoch* 15, *unfreeze event*): *backbone* diaktifkan dan *counter early stopping* direset. *Epoch* pertama setelah *unfreeze* menunjukkan penurunan sementara pada val mIoU karena penyesuaian *optimizer,* namun segera pulih pada epoch selanjutnya. Fase ketiga (Epoch 16–83, full fine-tuning): terjadi peningkatan konsisten, val mIoU meningkat dari 0,2695 ke 0,3629 sebelum early stopping aktif di epoch 93.

Gap antara train mIoU (0,5592 di *epoch* 83) dan val mIoU (0,3629) mengindikasikan adanya *overfitting* moderat, yang merupakan konsekuensi yang dapat diprediksi dari penggunaan pseudo-label dengan tingkat noise tertentu.

### D. Kesimpulan

Penelitian ini berhasil mencapai target yang ditetapkan, yaitu membangun model segmentasi semantik elemen antarmuka pengguna (UI/UX) berbasis arsitektur DeepLabV3+ dengan backbone MobileNetV3-Large yang mampu mengidentifikasi lima kelas elemen UI, yaitu *Background, Teks, Tombol, Gambar/Ikon,* dan *Navigasi*, menggunakan dataset gabungan RICO dan WebSRC. Model terbaik, yaitu V4 (Final), memperoleh nilai *Mean*

*Intersection over Union* (mIoU) sebesar 0,3537, *Pixel Accuracy* sebesar 0,6523, dan *Mean Dice Coefficient* sebesar 0,4800 pada epoch ke-83, serta menunjukkan peningkatan performa pada kelas *Tombol* sebesar 22,7% dibandingkan model baseline melalui penerapan strategi *progressive unfreezing backbone* yang dikombinasikan dengan mekanisme *reset counter* pada *early stopping*. Hasil penelitian ini memberikan dampak berupa tersedianya fondasi model *deep learning* yang mampu mengidentifikasi elemen UI secara otomatis dari *screenshot* antarmuka *mobile* dan *web*, sehingga berpotensi meningkatkan efisiensi, konsistensi, dan objektivitas proses evaluasi aksesibilitas digital yang selama ini masih banyak dilakukan secara manual. Untuk penelitian selanjutnya, disarankan melakukan anotasi manual pada sebagian data yang memiliki tingkat kesulitan tinggi guna meningkatkan kualitas *ground truth*, mengeksplorasi penggunaan *backbone* yang lebih kompleks seperti ResNet50 atau *Vision Transformer*, serta mengembangkan *pipeline* evaluasi aksesibilitas secara *end-to-end* yang mengintegrasikan hasil segmentasi dengan analisis berbasis standar *Web Content Accessibility Guidelines* (WCAG).

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Program Studi Sistem Informasi Universitas Darwan Ali atas dukungan dan bimbingan dalam pelaksanaan penelitian ini. Penelitian ini memanfaatkan sumber daya komputasi Google Colaboratory dengan GPU NVIDIA Tesla T4.

## DAFTAR PUSTAKA

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11211 LNCS*, 833–851. https://doi.org/10.1007/978-3-030-01234-2_49

Deka, B., Huang, Z., Franzen, C., Hibschman, J., Afergan, D., Li, Y., Nichols, J., & Kumar, R. (2017). Rico: A mobile app dataset for building data-driven design applications. *UIST 2017 - Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 845–854. https://doi.org/10.1145/3126594.3126651

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L. C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Le, Q., & Adam, H. (2019). Searching for mobileNetV3. *Proceedings of the IEEE International Conference on Computer Vision*, 1314–1324. https://doi.org/10.1109/ICCV.2019.00140

International Organization for Standardization. (2019). ISO 9241-210:2019 - Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. In *https://www.iso.org/standard/77520.html*. https://www.iso.org/standard/77520.html

Lateef, F., & Ruichek, Y. (2019). Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, *338*, 321–348. https://doi.org/10.1016/j.neucom.2019.02.003

Sunkara, S., Wang, M., Liu, L., Baechler, G., Hsiao, Y. C., Chen, J., Sharma, A., & Stout, J. (2022). Towards Better Semantic Understanding of Mobile Interfaces. *Proceedings - International Conference on Computational Linguistics, COLING*, *29*, 5636–5650.

Vigo, M., Brown, J., & Conway, V. (2013). Benchmarking web accessibility evaluation tools: Measuring the harm of sole reliance on automated tests. *W4A 2013 - International Cross-Disciplinary Conference on Web Accessibility*. https://doi.org/10.1145/2461121.2461124

World Wide Web Consortium, https://www.w3.org/TR/2018/REC-WCAG21-20180605/ (2018). https://www.w3.org/TR/2018/REC-WCAG21-20180605/

WebAIM. (2023, March 29). *The WebAIM MillionThe 2023 report on the accessibility of the top 1,000,000 home pages*. Https://Webaim.Org/Projects/Million/2023/. https://webaim.org/projects/million/2023/

WHO. (2023). Disability. In *https://www.who.int/news-room/fact-sheets/detail/disability-and-health*.

Zhang, X., Greef, L. De, & White, S. (2021, May). Screen Recognition: Creating Accessibility Metadata for Mobile Applications from Pixels. *Conference on Human Factors in Computing Systems - Proceedings*. https://doi.org/10.1145/3411764.3445186