BAB II

TINJAUAN PUSTAKA

A. Hasil Penelitian Terdahulu

Tabel 2.1 menunjukkan penelitian terdahulu yang relevan dengan penelitian yang dilaksanakan.

Tabel 2.1 Penelitian terdahulu

No	Peneliti	Objek kajian	Metode	Hasil penelitian
1.	Fuady *et al.* (2020)	Deteksi objek alat bantu untuk tongkat tunanetra	*Single Shot Multibox Detector*	Menghasilkan alat bantu pendeteksi objek berbasis kamera pada tongkat tunanetra dengan nilai akurasi 92% di luar ruangan.
2.	Ma *et al.* (2020)	Deteksi sampah berdasarkan jenisnya	*Single Shot Multibox Detector*	Menghasilkan pendeteksi sampah berdasarkan jenis bahan dengan mAP 74.02% dan 46 fps.
3.	Astuti *et al.* (2022)	Deteksi daun semanggi	*Single Shot Multibox Detector*	Menghasilkan pendeteksi daun semanggi dengan tingkat akurasi tertinggi sebesar 86.6% dan akurasi terendah 53,3%.
4.	Lu *et al.* (2022)	Deteksi retakan pada jembatan	*Improved Single Shot Multibox Detector*	Menghasilkan pendeteksi retak pada jembatan menggunakan metode *Improved Single Shot Multibox Detector*.

Tabel 2.1 Penelitian terdahulu (lanjutan)

No	Peneliti	Objek kajian	Metode	Hasil penelitian
5.	Saputra *et al.* (2022)	Deteksi jenis gulma pada tanaman jagung	*Single Shot Detector*	Menghasilkan pendeteksi jenis gulma pada tanaman jagung akurasi paling tinggi 81.57%.
6.	Apendi *et al.* (2023)	Deteksi isyarat SIBI Indonesia	*Single Shot Multibox Detector*	Menghasilkan *website* untuk mendeteksi bahasa isyarat SIBI dengan akurasi tinggi hingga 100% pada berbagai kondisi termasuk variasi jarak, pencahayaan dan sudut kamera.
7.	Fauzi *et al.* (2023)	Deteksi bahasa isyarat Malaysia	*Single Shot Detector*	Menghasilkan pendeteksi bahasa isyarat Malaysia dengan akurasi sebesar 75.2%.
8.	Pernando *et al.* (2023)	Deteksi sampah organik dan anorganik	*Single Shot Multibox Detector*	Menghasilkan aplikasi Android untuk mendeteksi sampah organik dan anorganik dengan 223 gambar *training* dan 55 gambar *testing* dengan tingkat akurasi sebesar 93% .
9.	Tanujaya & Lina (2023)	Deteksi bahan sembako	*Single Shot Multibox Detector*	Menghasilkan pendeteksi bahan sembako dengan *epoch* 150 dan akurasi 86.08%.

Tabel 2.1 Penelitian terdahulu (lanjutan)

No	Peneliti	Objek kajian	Metode	Hasil penelitian
10.	Al Ahmadi *et al.* (2024)	Deteksi isyarat alfabet Arab / huruf hijaiah	*You Only Look Once* (YOLO)	Menghasilkan model pendeteksi isyarat alfabet arab dengan *precision* dan *recall rate* mencapai 99%, serta mAP dihitung dengan IoU *threshold* diantara 50%-95%.
11.	Riswanto *et al.* (2024)	Deteksi kalori pada makanan tradisional Indonesia	*Single Shot Multibox Detector*	Menghasilkan aplikasi Android untuk mendeteksi jumlah kalori pada makanan dengan mAP sebesar 65.09%.

Tabel 2.1 menunjukkan berbagai penelitian sebelumnya dan penerapan metode kecerdasan buatan untuk mendukung deteksi objek. Model yang dikembangkan pada penelitian ini juga mengadopsi metode *Single Shot Multibox Detector* (SSD) karena sudah terbukti efektif seperti penelitian terdahulu yang sudah dijelaskan pada Tabel 2.1.

B. Landasan Teori

1. Isyarat Hijaiah

Isyarat hijaiah adalah bahasa gerak tangan yang digunakan untuk menerjemahkan alfabet Arab kepada penyandang tunarungu. Huruf hijaiah sendiri memiliki 28 huruf tunggal namun menjadi 30 huruf apabila memasukkan *lam-alif* (لا) sebagai huruf rangkap dan *hamzah* (ء) sebagai huruf dengan bacaan yang sama dengan *alif* (ا) (Nasution, 2020). Penelitian yang dilaksanakan menggunakan 28 huruf tunggal, antara lain :

ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي

Huruf hijaiah dimulai dari kanan ke kiri : alif - ba - ta - tsa - jim - ha - kho - da - dzal - ro - zai - sin - syin - shod - dhod - tho - zho – ain - ghain - fa - qof - kaf - lam - mim - nun - wawu - haa - ya.

Gambar 2.1 merupakan visualisasi dari isyarat hijaiah yang memiliki 28 jenis yang digunakan pada penelitian ini.

Gambar 2.1 Isyarat tangan 28 huruf hijaiah
(Al Ahmadi *et al.*, 2024)

2. *Machine Learning* (ML)

Machine learning adalah bidang pada *artificial intelligence* yang bertujuan untuk melatih atau mengajari mesin komputer dalam mengolah data tanpa di program secara eksplisit (Mahesh, 2020). ML membuat prediksi atau keputusan berdasarkan pola setelah belajar dari proses pelatihan. Klasifikasi teks dan citra merupakan contoh pengaplikasian ML.

ML dibagi menjadi tiga jenis proses pembelajaran model, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.

Supervised learning adalah proses pembelajaran model yang butuh data label untuk melakukan pelatihan. *Unsupervised learning* adalah proses pembelajaran model tanpa menggunakan data label sebagai *input*. *Reinforcement learning* adalah proses pembelajaran model dengan mengumpulkan informasi melalui respon dari lingkungan yang bereaksi pada tindakan.

3. *Deep Learning* (DL)

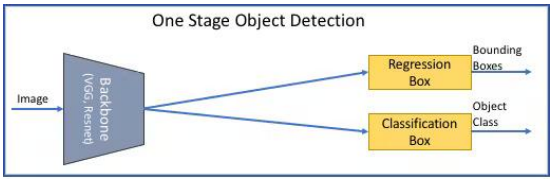
Machine learning beberapa dekade terakhir sudah mengalami kemajuan besar pada algoritma pembelajaran dan teknik pra pemrosesan yang efisien. Salah satunya perkembangan dari *machine learning*, yaitu *deep learning*. *Deep learning* adalah metode belajar yang memanfaatkan berlapis-lapis *artificial neural network*. *Artificial neural network* ini meniru otak manusia yang tersusun dari neuron-neuron yang sangat rumit (Raup *et al.*, 2022). Penerapan *deep learning* banyak digunakan pada berbagai bidang masa kini. Salah satu penerapan dari *deep learning* adalah deteksi objek.

4. Deteksi Objek

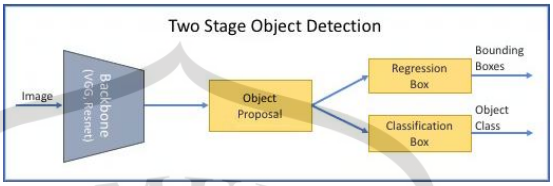
Deteksi objek adalah kemampuan komputer dalam mengidentifikasi objek pada gambar. Menurut Zhao *et al.* (2019) deteksi objek hadir untuk memberikan informasi yang berharga untuk memahami gambar dan video yang bisa menyambung dengan banyak pengaplikasian, termasuk klasifikasi gambar, analisis perilaku manusia, pengenalan wajah, mobil autopilot, dan lainnya. Tujuan utama deteksi objek, yaitu untuk

menemukan posisi objek dalam gambar yang sudah diberi label dan menghasilkan label yang sesuai dengan objek tersebut. Algoritma yang sering digunakan untuk deteksi objek, antara lain *Single Shot Multibox Detector* (SSD), *You Only Look Once* (YOLO), dan *Faster Region Based Convolutional Neural Network* (*Faster* R-CNN).

Deteksi objek memiliki dua pendekatan utama untuk menerapkannya, yaitu *one-stage object detectors* dan *two-stage object detectors*. *One-stage detector* hanya membutuhkan *input* gambar lalu langsung menghasilkan *output object class* dan *bounding boxes*. Algoritma yang termasuk jenis ini, antara lain SSD dan YOLO (*You Only Look Once*). *Two-stage detector* melakukan dua tugas berbeda yang kompleks. Proses dimulai dengan menghasilkan *object proposal* dari gambar *input*, kemudian menghasilkan *bounding boxes* dan *object class*. Algoritma yang menggunakan detektor jenis ini, yaitu *Faster* R-CNN (Hansen, 2023). Perbedaan cara kerja antara *one stage object detection* dan *two stage object detection* yang dijelaskan secara visual pada Gambar 2.2.



(a)



(b)

Gambar 2.2 Cara kerja *one-stage detector* (a) *two-stage detector* (b) (Han, 2025)

a. *One-Stage Detector*

Gambar yang sudah disiapkan diproses oleh *backbone*, contohnya *MobileNetV2*, *VGG* atau *Resnet* digunakan untuk mengekstraksi fitur pada gambar. Proses ini menghasilkan *regression box* yang menentukan koordinat *bounding box* pada objek dan *classification box* yang mengklasifikasi *object class* di dalam *bounding box*.

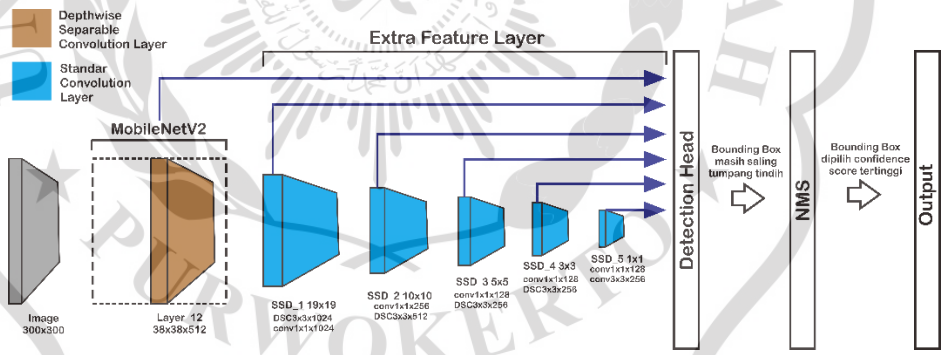
b. *Two-Stage Detector*

Gambar yang sudah disiapkan diproses oleh *backbone*, contohnya *MobileNetV2*, *VGG* atau *Resnet* digunakan untuk mengekstraksi fitur pada gambar. Fitur tersebut digunakan untuk membuat *object proposal* dalam menghasilkan kandidat area yang mengandung objek. *object proposal* menghasilkan *regression box* yang

menentukan koordinat *bounding box* pada objek dan *classification box* yang mengklasifikasi *object class* di dalam *bounding box*.

5. *Single Shot Multibox Detector* (SSD)

Single Shot Multibox Detector (SSD) adalah algoritma deteksi objek berbasis *Convolutional Neural Network* (CNN) yang dirancang untuk menghasilkan deteksi langsung dari data *input*. SSD menggunakan CNN sebagai jaringan dasar yang berguna untuk mengekstraksi fitur dengan menambahkan beberapa layer konvolusi tambahan dalam menghasilkan beberapa *bounding box* dan *confidence score* pada setiap kelasnya (Liu *et al.*, 2016). SSD dikenal cepat dan efisien dalam hal komputasi dibandingkan dengan metode lainnya. Struktur dari Arsitektur SSD ditampilkan pada Gambar 2.3.



Gambar 2.3 Arsitektur dari *single shot multibox detector* (Sandler *et al.*, 2019)

a. *Input* Gambar

Gambar *input* diubah ukurannya terlebih dahulu menjadi resolusi, contohnya menjadi 300×300 piksel sebelum masuk ke dalam *backbone*. Proses ini dilakukan sebagai langkah awal dalam arsitektur SSD. Ukuran yang telah diubah memiliki tujuan untuk memastikan

semua gambar memiliki ukuran yang sama dan lebih ringan sehingga proses menjadi lebih cepat saat melatih model.

b. *Feature Extraction*

Feature extraction merupakan proses mengekstraksi informasi penting pada gambar yang sudah dimasukkan. Tahap *feature extraction* menghasilkan dua jenis konvolusi, yaitu *depthwise separable convolution layer* dan *standard convolution layer*. Kedua konvolusi tadi menghasilkan *feature maps*. *Feature maps* adalah hasil kerja model dalam mempelajari gambar asli yang sudah dimasukkan sebelumnya. *Feature maps* ini menghasilkan *grid* dan *channel*. *Grid* merupakan kotak-kotak kecil pada gambar yang diolah oleh model, terdiri dari jumlah kotak tinggi dan lebar dengan skala S×S. *Channel* adalah lapisan atau ketebalan dalam *feature maps* sebagai tempat model menyimpan berbagai jenis informasi. Informasi tersebut berisi tepi, tekstur, dan pola lainnya. Banyaknya *channel* menunjukkan seberapa kompleks fitur yang berhasil diekstraksi dari gambar.

Data gambar yang sudah dimasukkan, masuk ke dalam *backbone MobileNetV2*. Layer_12 merupakan blok terakhir pada yang digunakan sebagai *backbone* detektor. SSD_1 merupakan layer pertama *extra feature layer* setelah melewati *backbone MobileNetV2*. *Extra feature layer* merupakan layer tambahan setelah *backbone* yang digunakan untuk mendeteksi objek pada berbagai skala. SSD_2 sampai

SSD_5 dibuat lebih ringan dan tidak terlalu mendalam dalam mengekstraksi fitur gambar.

c. *Detection Heads*

Detection head mengolah *feature maps* dengan menghasilkan sejumlah *anchor box* dengan berbagai ukuran dan posisi dari objek. Setiap *anchor box* digunakan untuk memprediksi pergeseran posisi dan ukuran *bounding box*, serta *confidence score* untuk tiap kelas objek. Banyak *anchor box* yang diuji dalam satu gambar mengakibatkan SSD menghasilkan banyak *bounding box* yang saling tumpang tindih. Tiap *anchor box* juga menghasilkan *confidence score* untuk setiap kelas objek sehingga yang dihasilkan cukup banyak. Hasil *bounding box* dan *confidence score* disaring pada proses *Non-Maximum Suppression*.

d. *Non-Maximum Suppression* (NMS)

Non-Maximum Suppression (NMS) memiliki tujuan untuk menghapus *bounding box* yang saling tumpang tindih yang dihasilkan dari proses *detection head*. *Bounding box* dengan *confidence score* paling tinggi dipertahankan, sedangkan yang nilainya lebih rendah atau mirip maka dihapus. NMS sangat berguna dalam menyaring hasil prediksi yang akurat sehingga akurasi dalam mendeteksi objek menjadi lebih optimal.

e. *Output*

Tahap akhir dari arsitektur SSD adalah *output*. Model menghasilkan informasi penting mengenai objek-objek yang berhasil

diprediksi dan disaring *bounding box* dengan nilai tertinggi. Setiap objek yang terdeteksi menampilkan posisi *bounding box*, nama atau label kelas objek dan *confidence score*.

Metode *Single Shot Multibox Detector* memiliki beberapa kelebihan dan kelemahan, yaitu :

a. Kelebihan :

1) SSD memiliki kecepatan yang tinggi dalam mendeteksi suatu objek yang bergerak sehingga sangat cocok untuk pengaplikasian secara *real-time.*
2) Arsitektur yang sederhana membuatnya mudah untuk diimplementasikan dan dilatih.

b. Kelemahan :

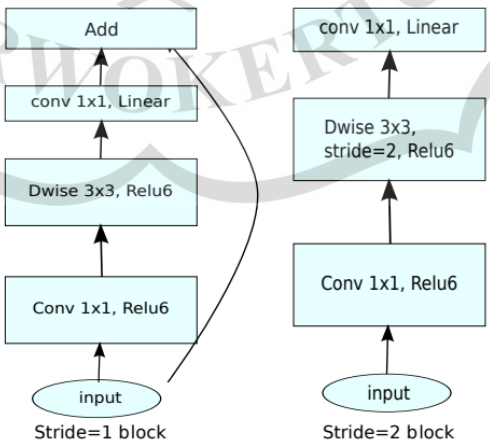
1) Akurasinya paling rendah dibanding metode lain apalagi untuk objek yang sangat kecil.
2) SSD membutuhkan data pelatihan yang bervariasi dan banyak supaya dapat memahami objek dengan baik, jika data kurang banyak maka SSD gagal dalam mendeteksi objek dengan baik.

6. *MobileNetV2*

MobileNetV2 adalah arsitektur CNN yang didesain untuk efisien dan model ringan *deep learning*, terutama dalam konteks tugas pada visi komputer (Shridhar *et al.*, 2023). *MobileNetV2* bekerja dengan mengidentifikasi pola visual seperti tekstur, tepi dan bentuk dari objek pada gambar dan diklasifikasikan dalam satu kelas tertentu. *MobileNetV2*

digunakan sebagai *backbone* dalam arsitektur SSD yang bertugas mengekstraksi fitur dari gambar masukan.

Desain ini merupakan peningkatan dari desain sebelumnya, yaitu *MobileNetV1*, yang dikembangkan oleh para peneliti di *Google AI*. Model *MobileNetV1* hanya memiliki satu *layer*, sedangkan model *MobileNetV2* memiliki dua *layer* yang membuat lebih cepat dalam memproses. *MobileNetV2* memiliki modul lapisan *inverted residuals* dan *linear bottlenecks* (Sandler *et al.*, 2019). *Inverted residuals* memungkinkan pengurangan jumlah parameter dengan tetap mempertahankan informasi penting, sedangkan *Linear bottleneck* menjaga agar tidak hilang dalam proses aktivasi non linear. *MobileNetV2* memiliki dua jenis lapisan konvolusi, yaitu 1×1 *convolution* dan 3×3 *depthwise convolution*. Setiap blok dalam *MobileNetV2* terdiri dari tiga lapisan, yaitu 1×1 *convolution with ReLU6*, *depthwise convolution*, dan 1×1 *convolution without any linearity*. Gambar 2.4 menunjukkan cara kerja *MobileNetV2*.



Gambar 2.4 Cara kerja *mobilenetv2*
(Sandler *et al.*, 2019)

7. *Python*

Python adalah bahasa pemrograman yang dirancang untuk mudah dipahami oleh manusia, serta mudah diproses mesin (Amri, 2024). Sintaks *python* sederhana dan fleksibel sebagai bahasa pemrograman sehingga dapat digunakan untuk berbagai kebutuhan contohnya untuk pengembangan *machine learning*. *Python* memiliki beragam pustaka yang mendukung semua tahap dari pengembangan mulai dari pra pemrosesan data hingga memprediksi sebuah model yang sudah dibuat.

8. *Confusion Matrix*

Confusion Matrix (CM) adalah kemampuan dari *machine learning* yang digunakan untuk mengukur klasifikasi pada tabel (Apendi *et al.*, 2023). Tabel kombinasi memiliki empat istilah, yaitu *true positive, false positive, false negative*, dan *true negative*. *True Positive* (TP) merupakan model berhasil mendeteksi adanya objek dengan benar. *True Negative* (TN) artinya model tidak mendeteksi objek karena memang objek tidak ada. *False Positive* (FP), yaitu model mendeteksi adanya objek padahal tidak ada objek. *False Negative* (FN) berarti model tidak mendeteksi objek padahal ada objek. TP, TN, FP, dan FN dapat menghasilkan nilai metrik evaluasi *accuracy*, *precision*, *recall*, serta *f1-score* (Murel, 2024).

a. *Accuracy*

Accuracy adalah metrik evaluasi yang menunjukkan prediksi benar dibagi dengan semua jumlah prediksi dari model. Nilai *accuracy* menunjukkan seberapa sering model memberikan hasil yang sesuai

dengan label sebenarnya. Setiap prediksi benar memberikan kontribusi pada peningkatan nilai *accuracy*. *Accuracy* digunakan untuk mengukur performa dari model secara keseluruhan yang ditunjukkan pada persamaan (1) (Vakili *et al*., 2020).

$$Accuracy = \frac{TN + TP}{TP + TN + FP + FN} \quad (1)$$

b. *Precision*

Precision adalah metrik yang mengukur ketepatan model dalam memprediksi kelas positif. Nilai *precision* dihasilkan dengan prediksi benar positif dibagi dengan jumlah prediksi positif. Setiap satu kesalahan model dalam memprediksi objek dapat menurunkan nilai *precision*. *Precision* berguna dalam menilai seberapa tepat model mengidentifikasi letak objek yang diperlihatkan pada persamaan (2) (Vakili *et al*., 2020).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

c. *Recall*

Recall adalah metrik yang mengukur model mampu mengenali kelas positif dengan benar. Metrik ini juga sering disebut sensitivitas. Nilai *recall* diperoleh dengan prediksi benar positif dibagi dengan jumlah objek sebenarnya pada data. *Recall* ini penting dengan fokus menghindari kehilangan deteksi terhadap objek yang harusnya dikenali model terdapat pada persamaan (3) (Vakili *et al*., 2020).

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

d. *F1-score*

F1-score adalah metrik yang mencerminkan keseimbangan antara *precision* dan *recall*. Metrik ini digunakan untuk menilai performa model ketika terdapat ketidakseimbangan antara data positif dan negatif ditampilkan pada persamaan (4) (Vakili *et al*., 2020).

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$