

Metode Pemrosesan Citra dan Pembelajaran Mesin Konvensional untuk Mendeteksi Akor Gitar Visual

Ryan Samuel Chandra (13521140)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): cadanganryansc@gmail.com

Abstract—Gitar adalah salah satu alat musik yang relatif mudah untuk dipelajari. Akan tetapi, belajar atau mengajarkan bermain gitar tetap butuh kesabaran dan ketekunan. Di era modern, kita mengenal pembelajaran mesin (*machine learning*). Makalah ini membahas tentang pemanfaatan pembelajaran mesin tradisional (SVM dan kNN) dalam mengenali akor dasar gitar melalui citra tangan). Selain implementasi, makalah ini juga mengandung pembahasan lebih mendalam tentang perbandingan variasi prapemrosesan citra sebelum digunakan.

Keywords—Gitar, Pemrosesan Citra, *Machine Learning*, Akor Dasar, SVM, kNN, CNN

I. PENDAHULUAN

Manusia pada dasarnya diciptakan untuk bekerja dan berkarya di dunia. Meskipun demikian, manusia pasti merasa lelah setelah menghabiskan seluruh energi untuk melakukan pekerjaannya. Salah satu hal yang dapat membantu manusia dalam mempercepat pemulihan adalah karya seni. Menurut KBBI, seni adalah ciptaan yang dapat menimbulkan rasa indah bagi orang yang melihat, mendengar, atau merasakannya. Hingga kini, ada beragam jenis seni, mulai dari seni rupa, seni bela diri, seni peran, hingga seni musik. Guna membawa kenyamanan bagi penikmat, seniman juga kerap dijadikan pekerjaan oleh sebagian orang. Seorang pelukis profesional mahir dalam menciptakan gambar yang estetik secara visual. Musisi mampu menghasilkan suara yang merdu dengan bernyanyi atau menggunakan alat musik, demikian seterusnya.

Gitar adalah sebuah alat musik yang diminati oleh banyak orang karena mudah dibawa dan dipelajari. Namun, tetap dibutuhkan konsistensi dan kesabaran dalam mempelajarinya. Saat ini, ada banyak alat bantu yang dapat digunakan untuk mempelajari gitar. Hal yang membuat penulis penasaran adalah cara menerapkan kecerdasan buatan (*artificial intelligence*) dalam mengenali akor gitar. Apalagi, saat ini pembelajaran mesin (*machine learning*) lebih mudah dilakukan karena kekuatan komputasi sudah meningkat.

II. TINJAUAN PUSTAKA

A. Alat Musik Gitar

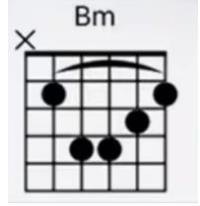
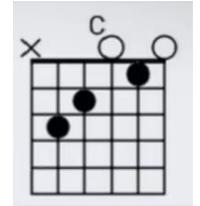
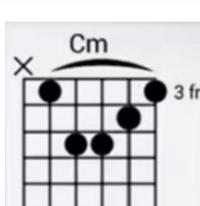
Gitar merupakan salah satu alat musik melodis yang sangat tidak asing bagi masyarakat Indonesia. Alat musik petik ini

mudah dimainkan untuk berbagai genre musik, jika kunci dasarnya dipahami dan dikuasai dengan benar. Memahami kunci dasar bermain gitar tidaklah sulit. Anda bisa memulainya dengan mengenal bentuk dari masing-masing kunci dasar dan belajar praktek secara berkala. Dengan giat dan sering menerapkan kunci, tentunya Anda akan lebih cepat memahami kunci dasar untuk belajar bermain gitar dengan cepat.

Secara umum, kunci (akor) dasar bermain gitar ada 7 yaitu A, B, C, D, E, F dan G. Masing-masing memiliki nada mayor dan minor. Pengembangan dari kunci dasar gitar ini juga banyak, mulai dari *sharp* (#) hingga *sus*. Berikut adalah beberapa contoh (tidak lengkap) gambar kunci dasar gitar.

Tabel I.1 Beberapa Akor Dasar Gitar [3]

Akor	Finger Map	Bentuk Tangan
A		
Am		
B		

Bm		
C		
Cm		
dan seterusnya...		

B. Deteksi Tepi dalam Pemrosesan Citra

Tepi dalam konteks citra adalah perubahan nilai intensitas keabuan yang mendadak (besar) dalam jarak yang singkat [1]. Tepi biasanya terdapat pada batas antara dua daerah. Deteksi tepi bertujuan untuk meningkatkan penampakan garis batas atau objek di dalam citra. Pendeteksian tepi merupakan bagian dari analisis citra, yaitu mengidentifikasi parameter-parameter yang diasosiasikan dengan ciri (*feature*) dari objek di dalam citra, untuk selanjutnya digunakan dalam menginterpretasikan citra (klasifikasi) [1].



Gambar II.1 Contoh Deteksi Tepi pada Citra

(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/18-Pendeteksian-Tepi-Bagian1-2024.pdf>)

Ada banyak operator deteksi tepi, mulai dari operator gradien (turunan pertama), karena perubahan intensitas dalam jarak yang singkat dipandang sebagai fungsi yang memiliki kemiringan besar. Selain itu, ada juga operator turunan kedua (Laplace) yang lebih akurat dalam mendeteksi tepi curam. Operator ini juga sering digabungkan dengan operator Gauss

menjadi LoG (*Laplacian of Gaussian*) untuk menghilangkan derau aditif sebelum mendeteksi tepi.

Operator lain dalam deteksi tepi adalah Prewitt, yaitu magnitudo gradien yang dihitung dengan rumus $M = \sqrt{p_x^2 + p_y^2}$. Nilai p_x dan p_y diperoleh dari kombinasi piksel-piksel sekitar untuk setiap titik piksel (x, y) di dalam citra. Agar mempersingkat perhitungan, p_x dan p_y dapat diperoleh melalui konvolusi citra dengan 2 buah *mask* berukuran 3×3 berikut.

$$P_x = [[-1, 0, 1], [-1, 0, 1], [-1, 0, 1]]$$

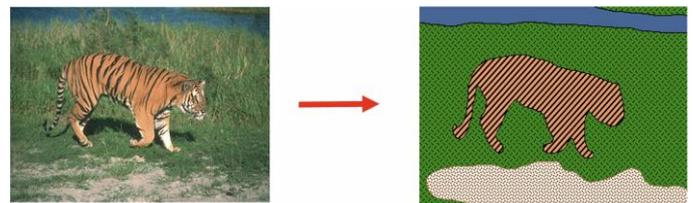
$$P_y = [[1, 1, 1], [0, 0, 0], [-1, -1, -1]]$$

Operator Canny juga sangat terkenal dalam deteksi tepi karena mampu menghasilkan tepi dengan ketebalan 1 piksel. Pertama-tama, citra asli dilewatkan pada penapis *Gaussian* untuk menghaluskan citra sekaligus menghilangkan derau. Kemudian, gradien dihitung operator Sobel, Prewitt, atau Roberts. Operator Canny sendiri adalah pengembangan (*thresholding*) sebagai lanjutan operator-operator sebelumnya.

Terdapat dua nilai ambang yang digunakan, yaitu T1 dan T2 ($T1 < T2$). Jika magnitudo gradien lebih besar dari T2, piksel tersebut adalah tepi kuat (*strong edge*). Setiap piksel yang terhubung dengan tepi kuat dan memiliki magnitudo gradien lebih besar dari T1, adalah tepi lemah (*weak edge*). Terakhir, dilakukan penautan tepi (*edge-linking*) untuk menghubungkan tepi lemah dengan tepi kuat.

C. Segmentasi Citra

Segmentasi citra adalah proses mempartisi citra menjadi sejumlah bagian atau objek, sehingga setiap bagian terdiri dari sekumpulan piksel yang terhubung satu sama lain [5]. Segmentasi citra bertujuan untuk membagi citra menjadi segmen-segmen atau objek-objek yang berbeda atau memisahkan objek dengan latar belakangnya. Segmentasi ini merupakan tahapan sebelum *image/object recognition*, *image understanding*, dan lain-lain.



Gambar II.2 Contoh Segmentasi Citra Harimau

(Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/22-Segmentasi-Citra-Bagian1-2024.pdf>)

Segmentasi citra dapat dilakukan berdasarkan diskontinuitas (*edge detection*) atau berdasarkan *similarity* (kemiripan area menurut properti yang ditentukan).

D. Ekstraksi Fitur Citra

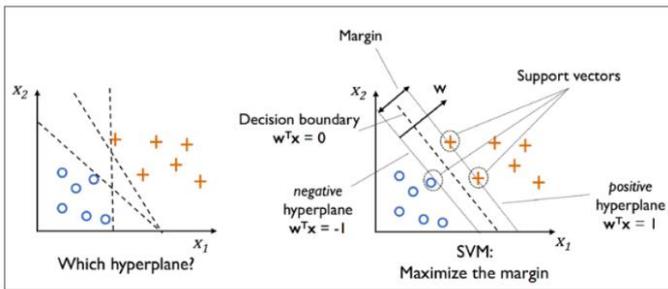
Ekstraksi fitur dari sebuah citra menjadi senarai angka yang representatif dapat dilakukan dengan berbagai cara. Salah satunya adalah menggunakan *Histogram of Oriented Gradients* (HOG). Citra dibagi menjadi daerah kecil yang disebut *cell*. Setiap *cell* biasanya berukuran 8×8 piksel. Nilai

gradien dihitung untuk setiap piksel dalam *cell* menggunakan teknik konvolusi, sering kali dengan filter Sobel. Gradien ini mencakup komponen horizontal dan vertikal.

Berdasarkan nilai gradien, dihitunglah histogram orientasi gradien untuk setiap *cell*. Histogram menunjukkan distribusi arah gradien di dalam *cell* dan biasanya memiliki 9 *bin* untuk orientasi dari 0 hingga 180 derajat. Beberapa *cell* digabungkan menjadi blok, dan nilai histogram pada blok tersebut dinormalisasi untuk mengurangi efek perubahan pencahayaan. Setelah normalisasi, hasilnya digabungkan menjadi vektor fitur HOG yang dapat digunakan untuk deteksi objek.

E. Support Vector Machines (SVM)

SVM adalah salah satu algoritma pembelajaran mesin konvensional (tradisional) dalam *supervised learning* yang biasa digunakan untuk klasifikasi. SVM digunakan untuk mencari *hyperplane* terbaik dengan memaksimalkan jarak antar kelas. *Hyperplane* adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas.



Gambar II.3 Ilustrasi *Support Vector Machines*
(Sumber: <https://medium.com/@samsudiney/penjelasan-sederhana-tentang-apa-itu-svm-149fec72bd02>)

F. K-Nearest Neighbors (kNN)

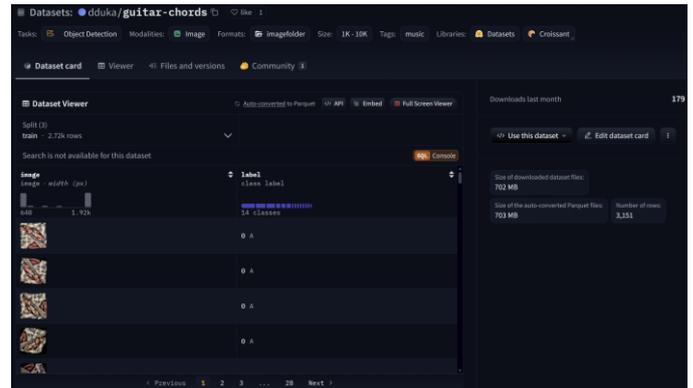
kNN adalah algoritma yang digunakan untuk mengidentifikasi adanya persamaan antara data baru dan lama. Kemudian, algoritma satu ini akan memasukkan data baru tersebut dalam kategori yang paling mirip dengan kategori yang telah ada sebelumnya [6]. Dengan kata lain, *k-nearest neighbors* menyimpan harus seluruh data lama.

Kelebihan kNN adalah algoritmanya yang sederhana dan mudah diimplementasikan. Tidak diperlukan model kompleks, pengaturan parameter yang rumit, serta beragam asumsi. Algoritma kNN ini serbaguna, bisa untuk membuat klasifikasi, regresi, dan pencarian data. Namun, algoritma kNN juga memiliki kelemahan. Algoritma ini bisa menjadi lebih lambat dan berat komputasinya secara signifikan karena jumlah contoh atau prediksi variabel independennya meningkat. Selain itu, kNN juga selalu memerlukan penentuan nilai *k* paling baik.

III. IMPLEMENTASI

Implementasi dilakukan di dalam Google Colab atas dasar aksesibilitas yang tinggi dan gratis. *Dataset* yang digunakan berasal dari Hugging Face, yaitu “*dduka/guitar-chords*”.

Dataset terdiri atas dua kolom, dengan kolom pertama berupa gambar tangan yang sedang menekan akor gitar dan kolom kedua adalah nama akornya. Ada 14 buah akor, yaitu "A", "Am", "B", "Bm", "C", "Cm", "D", "Dm", "E", "Em", "F", "Fm", "G", "Gm".



Gambar III.1 *Dataset* Akor Gitar dari Hugging Face
(Sumber: <https://huggingface.co/datasets/dduka/guitar-chords>)

Algoritma pembelajaran mesin konvensional yang digunakan adalah SVM dan kNN. Ada beberapa variasi yang diterapkan terhadap citra sebelum diekstraksi fiturnya. Variasi yang dimaksud adalah menggunakan operator deteksi tepi Canny, menggunakan operator deteksi tepi Prewitt, menggunakan segmentasi citra, serta tanpa pemrosesan sama sekali.

Algoritma Deteksi Tepi: Operator Prewitt

```
def prewitt_edge_detection(image):
    image = np.array(image)

    prewitt_kernel_x = np.array([[1, 0, -1], [1, 0, -1], [1, 0, -1]])
    prewitt_kernel_y = np.array([[1, 1, 1], [0, 0, 0], [-1, -1, -1]])

    # Convert to grayscale
    gray_hand_crop = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Apply the filter
    prewitt_x = cv2.filter2D(gray_hand_crop, cv2.CV_64F, prewitt_kernel_x)
    prewitt_y = cv2.filter2D(gray_hand_crop, cv2.CV_64F, prewitt_kernel_y)
    prewitt = cv2.magnitude(prewitt_x, prewitt_y)

    prewitt_uint8 = cv2.convertScaleAbs(prewitt)
    return (Image.fromarray(prewitt_uint8))
```

Algoritma Ekstraksi Fitur: HOG

```
def extract_features(image):
    image_np = np.array(image)

    if len(image_np.shape) == 2:
        gray = image_np
    else:
        gray = cv2.cvtColor(image_np, cv2.COLOR_BGR2GRAY)

    resized = cv2.resize(gray, (128, 128)) # Resize gambar ke ukuran tetap

    # Ekstraksi fitur HOG
    hog_features = hog(resized, orientations=9, pixels_per_cell=(8, 8),
                       cells_per_block=(2, 2), block_norm='L2-Hys',
                       visualize=False)

    return (hog_features)
```


Tabel IV.1 Akurasi Model terhadap Data Validasi

Variasi <i>Preprocessing</i>	SVM	kNN
Deteksi Tepi: Canny	47,59%	47,24%
Deteksi Tepi: Prewitt	12,76%	7,93%
Segmentasi Citra	19,66%	24,48%
Tanpa <i>Preprocessing</i>	100%	67,93%

Berdasarkan hasil tersebut, diketahui bahwa model konvensional yang mendapatkan akurasi paling tinggi justru adalah model SVM yang dilatih dengan data mentah (tidak dilakukan *preprocessing*). Hal ini mungkin bisa terjadi karena *preprocessing* justru membuat beragam fitur esensial citra malah hilang. Apalagi, setelah melalui eksplorasi, tidak ada algoritma segmentasi citra yang berhasil cukup baik dalam memisahkan gambar gitar dan tangan dari latar belakangnya.

Dua model terbaik, yaitu model dengan *preprocessing* data pelatihan menggunakan operator Canny, dan model tanpa *preprocessing* data pelatihan, masuk ke tahap selanjutnya yaitu pengujian terhadap data uji. Berdasarkan pengujian tersebut, didapatkan hasil sebagai berikut.

Tabel IV.2 Akurasi Model terhadap Data Uji

Variasi <i>Preprocessing</i>	SVM	kNN
Tanpa <i>Preprocessing</i>	100%	67,93%
Deteksi Tepi: Canny	41,55%	51,41%

Maka dari itu, terbukti bahwa model terbaik justru adalah model yang dilatih dengan data tanpa *preprocessing*. Percobaan dengan CNN sempat dilakukan tetapi gagal karena akurasi setiap *epoch* malah menurun (15 *epoch* secara keseluruhan). *Library* “mediapipe” untuk memotong gambar menjadi sekitar area tangan saja juga gagal digunakan karena memakan RAM yang sangat besar.

V. SARAN DAN PROSPEK PENGEMBANGAN

Penelitian kali sangat berpotensi memiliki kualitas *dataset* yang sangat jelek, karena kelas-kelasnya *imbalance*. Oleh karena itu, diperlukan lebih banyak *data augmentation*, termasuk menyeimbangkan jumlah kelas, menyaring akor menjadi akor-akor yang paling sederhana dulu (C, D, E, F, G, A, B), juga memotong citra menjadi bagian tangannya saja. Jika penelitian semacam ini berhasil, bisa lanjut ke tingkat yang lebih tinggi yaitu dengan input berupa video.

Terkait dengan model CNN yang gagal, perlu eksperimen menggunakan CCN yang arsitekturnya lebih kompleks, dengan parameter yang lebih disesuaikan lagi. Jika ada kekuatan komputasi yang lebih besar hal ini bisa dicoba. Untuk model kNN, harus mencari nilai optimal k dengan algoritma lain.

PRANALA PENTING

1. Google Colab eksperimentasi dan pembangunan model:

- SVM (*Basic*) dan Percobaan CNN

https://colab.research.google.com/drive/1vchZLdPnm_d404fG1_z4J1F5qL2IRCw0s?usp=sharing

- SVM dan KNN (+ *preprocessing image*)

https://colab.research.google.com/drive/1DBYcm_YqBgU_PyS-6RnZB8wz6fcAcieH?usp=sharing

2. *Dataset* yang digunakan:

<https://huggingface.co/datasets/dduka/guitar-chords/viewer/default/train?p=2>

UCAPAN TERIMA KASIH

Pertama-tama, penulis memanjatkan puji dan syukur kepada Tuhan Yang Maha Esa, atas hikmat dan kelancaran yang telah diberikan-Nya kepada penulis, sehingga makalah ini dapat selesai dengan baik dan tepat waktu. Penulis juga mengucapkan terima kasih kepada kedua orang tua, serta teman-teman yang sudah mendukung serta memotivasi selama pembuatan makalah ini.

Kemudian, penulis juga berterima kasih secara khusus kepada Bapak Dr. Ir. Rinaldi, M.T., selaku dosen pengajar mata kuliah IF4073 Pemrosesan Citra Digital pada tahun ajaran 2024-2025, yang telah memberikan ilmu dan kesempatan kepada penulis untuk mengerjakan makalah ini.

DAFTAR REFERENSI

- Munir, R. (2024). *Pendeteksian Tepi (Bagian 1) – Bahan Ajar IF4073 Pemrosesan Citra Digital Institut Teknologi Bandung* (diakses dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/18-Pendeteksian-Tepi-Bagian1-2024.pdf> pada tanggal 15 Januari 2025).
- Munir, R. (2024). *Pendeteksian Tepi (Bagian 2) – Bahan Ajar IF4073 Pemrosesan Citra Digital Institut Teknologi Bandung* (diakses dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/19-Pendeteksian-Tepi-Bagian2-2024.pdf> pada tanggal 15 Januari 2025).
- Yamaha Corporation. Tanpa tanggal. *Kunci Dasar Gitar* (diakses dari https://id.yamaha.com/id/news_events/2022/20220711_kuncidasargitar.html pada tanggal 15 Januari 2025).
- Samsudiney. (25 Juli 2019). *Penjelasan Sederhana tentang Apa Itu SVM?* (diakses dari <https://medium.com/@samsudiney/penjelasan-sederhana-tentang-apa-itu-svm-149fec72bd02> pada tanggal 25 Januari 2025).
- Munir, R. (2024). *Pendeteksian Tepi (Bagian 2) – Bahan Ajar IF4073 Pemrosesan Citra Digital Institut Teknologi Bandung* (diakses dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2024-2025/22-Segmentasi-Citra-Bagian1-2024.pdf> pada tanggal 15 Januari 2025).
- Anonim. (22 September 2022). *Mengenal K-nearest Neighbor dan Pengaplikasiannya* (diakses pada tanggal 15 Januari 2025 dari <https://blog.algorit.ma/k-nearest-neighbor/>).

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 15 Januari 2024

A handwritten signature in blue ink, consisting of stylized, overlapping letters that appear to be 'RSC'.

Ryan Samuel Chandra (13521140)