

# Sistem Rekognisi Akor Instrumen Musik Secara Otomatis Menggunakan PCP dan SVM

Gede Nicholas Tejasukmana Putra<sup>a1</sup>, Ngurah Agus Sanjaya<sup>a2</sup>

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Udayana  
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia  
<sup>1</sup>putra.2308561074@student.unud.ac.id  
<sup>2</sup>agus\_sanjaya@unud.ac.id (Corresponding Author)

## Abstract

*This study presents a system for automatic chord recognition from audio recordings using the Pitch Class Profile (PCP) and Support Vector Machine (SVM). PCP was chosen as the primary feature extraction method because it can represent the standard 12 pitch classes in music accurately. SVM was selected as the classification model because of its proven success in previous chord recognition studies, offering high accuracy while remaining efficient. Using the Piano Triads Wavset dataset, which contains 432 triad chords across 12 root notes and three chord types such as major, minor, and diminished, the model was trained and tested in an experiment. The audio data were processed to extract PCP features and normalized before being classified using SVM. Evaluation was carried out using both a default SVM configuration and GridSearchCV optimization. Results show that the optimized model achieved up to 82% accuracy across all chord classes, indicating that the proposed approach can recognize chords reliably even without using deep learning or additional features. The final system also includes real-time prediction by user audio input, using Python and streamlit framework.*

**Keywords:** Chord Recognition, PCP, SVM, GridSearchCV, Python

## 1. Pendahuluan

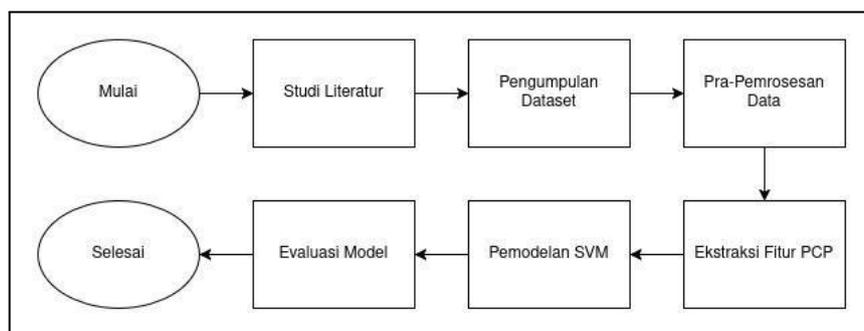
*Music Information Retrieval (MIR)* merupakan ilmu yang bertujuan untuk memperluas pencarian informasi ke area non-tekstual, secara singkat menjelaskan berbagai aspek yang terkait konten musik [1]. Dalam konteks MIR, terdapat beberapa macam implementasi, salah satunya adalah pengenalan akor. *Chord* atau akor sendiri merupakan salah satu elemen dasar dalam musik, yaitu gabungan beberapa nada yang dimainkan secara bersamaan dan membentuk harmoni tertentu. Mengenali akor secara otomatis dari rekaman suara terkadang menjadi tantangan, terutama karena suara musik memiliki banyak variasi tergantung pada jenis instrumen, teknik bermain, dan kondisi rekaman. Penelitian ini ingin mengajukan metode yang efisien dalam penggunaan sumber daya, namun dapat mengakomodasi keragaman suara musik dan memberikan hasil yang akurat. Dalam beberapa tahun terakhir, *Pitch Class Profile (PCP)* telah menjadi salah satu fitur yang banyak digunakan untuk representasi nada-nada dalam musik dan masih relevan sampai saat ini, karena kemampuannya untuk menangkap informasi pitch secara ringkas dan cukup akurat. PCP mengubah sinyal suara menjadi representasi 12-bin sesuai dengan 12 nada yang terdapat pada satu oktaf dalam teori musik [1]. Kesesuaian ini membuat PCP cocok digunakan untuk mengidentifikasi struktur akor dalam berbagai konteks musik. Beberapa penelitian terdahulu telah membuktikan bahwa penggunaan algoritma *Support Vector Machine (SVM)* bersama dengan fitur PCP mampu memberikan hasil pengenalan akor dengan tingkat akurasi yang cukup tinggi, bahkan lebih dari 91% [2]. Namun, penelitian tersebut umumnya hanya berfokus pada satu jenis instrumen atau lingkungan tertentu yang spesifik. Di sisi lain, terdapat juga beberapa penelitian terdahulu yang mencoba mengenali akor dari berbagai instrumen musik populer seperti piano, biola, dan akordion, tetapi pendekatannya menggunakan konsep *neural network* yang cenderung membutuhkan sumber daya komputasi lebih besar dan proses pelatihan yang lebih kompleks [1]. Berdasarkan pengetahuan ini, penelitian yang diajukan akan tetap menggunakan kombinasi SVM dan PCP sebagai pendekatan utama karena telah

terbukti cukup kuat dan ampuh dari sisi akurasi, tetapi juga ringan dari sisi kebutuhan komputasi karena tidak perlu menggunakan berbagai macam fitur lain yang belum tentu meningkatkan hasil akurasi. Penelitian ini akan melengkapi studi-studi sebelumnya dengan mencoba menguji kemampuan SVM dan PCP dalam mengenali akor dari beragam jenis instrumen musik populer walau dengan dataset yang terbatas. Sistem yang dirancang akan menyediakan fitur untuk merekam akor instrumen sesuai keinginan dan kemudian akan memberikan prediksi akor tersebut secara *real-time*.

## 2. Metode Penelitian

### 2.1. Alur Penelitian

Penelitian ini bertujuan untuk mengembangkan sistem pengenalan akor musik secara otomatis dari rekaman audio instrumen, khususnya instrumen musik populer, menggunakan fitur *Pitch Class Profile* (PCP) standar dan algoritma *Support Vector Machine* (SVM). Secara garis besar, penelitian ini mengikuti tahapan-tahapan sistematis mulai dari studi literatur, pengumpulan data yang diperlukan, kemudian pra-pemrosesan data tersebut, ekstraksi fitur dari dataset, pelatihan model, hingga evaluasi performa. Berikut merupakan diagram yang menggambarkan alur penelitian ini seperti yang terlihat pada Gambar 1.



Gambar 1. Diagram Alur Penelitian

### 2.2. Akor

Akor dalam konteks musik merupakan kumpulan not atau nada yang dibunyikan secara bersamaan untuk menghasilkan harmoni tertentu [3]. Akor dasar yang paling umum diketahui adalah akor *major*, *minor*, dan *diminished triads*. Setiap akor ini dibentuk dari kombinasi nada-nada tertentu yang memiliki hubungan interval (jarak antar not) tetap, misalnya akor C *major* terdiri dari not C, E, dan G. Terdapat banyak cara memainkan 1 akor karena tiap not dapat dimainkan dalam oktaf yang berbeda, terlebih lagi variasi yang membuat bunyi akor semakin kompleks. Namun, penelitian ini hanya akan berfokus pada pengenalan akor dalam 12 skala *pitch* standar yakni akor C, C#, D, D#, E, F, F#, G, G#, A, A#, dan B sebagai dasar penelitian yang dapat dikembangkan kembali selanjutnya.

### 2.3. Pengumpulan Data

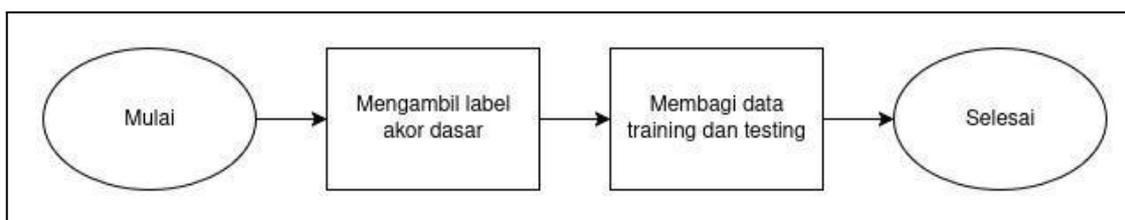
Dataset yang digunakan dalam penelitian ini bernama *Piano Triads Wavset*, dapat diperoleh dari situs *kaggle* dan dibuat oleh David Roberts. Penelitian ini sangat dimudahkan dengan adanya dataset ini karena sudah berisi 432 akor piano dalam cakupan 6 oktaf serta variasi masing-masing 12 akor *major*, *minor*, dan *diminished triads*. Seluruh kombinasi not yang dapat membentuk akor secara spesifik juga sudah ditambahkan dalam *file csv*, sehingga pencarian nama *file* dan pembuatan label menjadi tidak rumit. Secara keseluruhan, dataset disusun sedemikian rupa, dengan sangat rapi dan berisi informasi penting yang dibutuhkan dalam penelitian ini. Berikut merupakan rincian dari *file csv* yang tersedia dalam dataset ini tertera dalam bentuk Tabel 1.

**Tabel 1.** Rincian Dataset

No	Nama Kolom	Jumlah Entri	Deskripsi
1	Chord	360	Nama, variasi, kategori, oktaf, dan bentuk akor
2	Note1	360	Not pertama yang membentuk akor tertentu
3	Note2	360	Not kedua yang membentuk akor tertentu
4	Note3	360	Not ketiga yang membentuk akor tertentu

#### 2.4. Pra-Pemrosesan Data

Dikarenakan dataset yang sudah rapi, penamaan *file* yang seragam sesuai dengan label akor, *file csv* berisi daftar seluruh kombinasi not menjadi akor yang spesifik, menyebabkan proses pra-pemrosesan data ini tidak memerlukan terlalu banyak tahap untuk memenuhi keperluan penelitian yang akan dilakukan. Walau hanya dengan data audio piano, bentuk PCP dari instrumen lain cukup serupa sehingga tetap mampu dikenali oleh sistem [1]. Beberapa hal yang perlu dilakukan adalah memuat seluruh dataset untuk kemudian membuat *data frame* dari *file csv*. Program dapat mengambil nama *file* dan label dari kolom 'Chord' pada *data frame* tersebut yang akan digunakan dalam proses *training* dan *testing* model. Gambar 2 merupakan representasi diagram yang menggambarkan tahap pra-pemrosesan data ini.



**Gambar 2.** Diagram Pra-Pemrosesan Data

#### 2.5. Ekstraksi Fitur

Ekstraksi fitur dilakukan dengan menghitung *Pitch Class Profile* (PCP) 12-bin dari tiap *file* audio. PCP bekerja dengan mengubah sinyal frekuensi hasil *Short-Time Fourier Transform* (STFT) atau *Constant-Q Transform* (CQT) menjadi representasi energi untuk tiap dari 12 nada dalam satu oktaf, tanpa memperhatikan oktafnya [3]. Meskipun PCP tidak menangkap informasi durasi atau struktur waktu secara eksplisit, fitur ini tetap sangat efektif untuk mengenali struktur akor karena akor ditentukan oleh hubungan nada-nada, bukan posisi nada dalam waktu. Berikut merupakan rumus yang dapat digunakan dalam perhitungan PCP.

$$PCP(p) = \sum_{l \text{ s.t. } M(l)=p} ||X(l)||^2 \quad (1)$$

$$M(l) = \text{round}(12 \log_2 \frac{f_l}{f_{ref}}) \pmod{12} \quad (2)$$

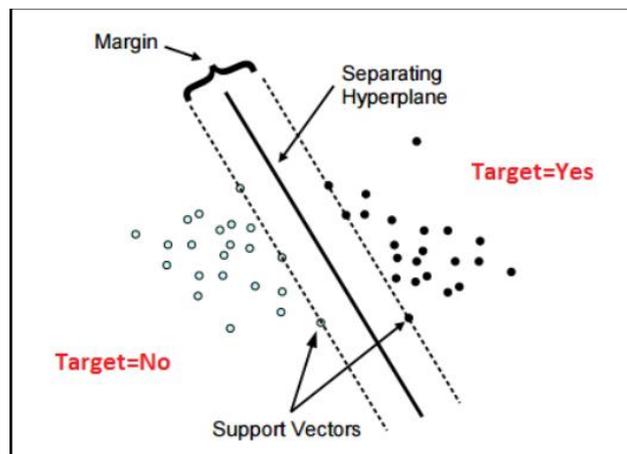
Persamaan 1 menghitung total energi untuk *pitch class p* dan persamaan 2 merupakan fungsi yang memetakan bin *l* ke dalam *pitch class*. Memilih *pitch class p*, misalkan P (0) untuk mencari akor C, berarti algoritma harus menemukan nilai bin *l*, dimana M(*l*) = 0. Setelah melewati banyak iterasi memeriksa setiap bin dalam STFT untuk menemukan semua nilai bin *l* tersebut,

algoritma menjumlahkan STFT dari bin  $l$  yang ditemukan sebagai total *pitch class* 0.

## 2.6. Pelatihan Model

Algoritma *machine learning* yang digunakan dalam penelitian ini adalah *Support Vector Machine* (SVM). SVM ini termasuk dalam algoritma supervised yang menggunakan pendekatan pelatihan dengan label pada dataset untuk mengenali pola dan melakukan prediksi terhadap data baru. SVM dipilih karena efisien untuk dataset dengan dimensi rendah (seperti PCP 12-bin), dan memiliki performa tinggi untuk klasifikasi non-linear. Beberapa hal di bawah ini merupakan terminologi yang ada dalam konsep SVM [4].

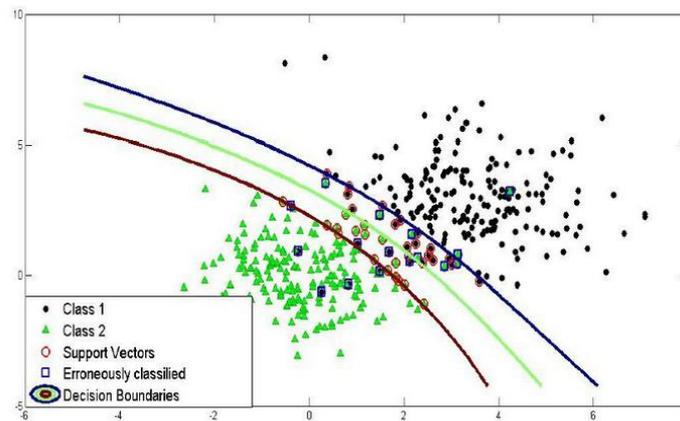
- a. Hyperplanes  
Garis yang menjadi pembatas untuk memisahkan kumpulan data dengan kelas label yang berbeda.
- b. Support Vectors  
Sampel data yang paling dekat dengan *hyperplane* dan menjadi acuan untuk membuat *hyperplane* dengan bantuan margin.
- c. Margins  
Jarak yang memisahkan 2 garis dari *support vector* kumpulan data sebelumnya. Untuk lebih memperjelas, selengkapnya dapat dilihat pada Gambar 3.



Gambar 3. Visualisasi SVM

- d. Kernel  
Fungsi yang memetakan data ke dalam dimensi yang lebih tinggi sehingga dapat dibedakan. Terdapat beberapa macam fungsi kernel, namun fungsi yang digunakan dalam penelitian ini adalah *Radial Basis Function* (RBF) seperti pada Gambar 4. Kernel RBF digunakan karena memungkinkan model membentuk batas keputusan kompleks tanpa perlu fitur tambahan. Berikut merupakan rumus yang mendefinisikan kernel RBF bersamaan dengan visualisasi klasifikasi SVM menggunakan kernel ini.

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \quad (3)$$



**Gambar 4.** Visualisasi Kernel RBF

Dalam proses pelatihan model, *Grid Search Cross-Validation (GridSearchCV)* digunakan untuk meningkatkan performa model. Teknik ini bekerja dengan cara mencoba seluruh kombinasi parameter yang diberikan, untuk kemudian digunakan dalam menguji model. Dengan begitu, teknik ini memberikan konfigurasi parameter terbaik yang dapat menghasilkan performa model terbaik, dari sisi akurasi maupun biaya komputasi.

## 2.7. Evaluasi Model

Evaluasi model diukur dengan menggunakan *classification report* yang menampilkan beberapa metrik pada umumnya, yakni *accuracy*, *precision*, *recall*, dan *f1-score*. Prediksi yang dibuat model dapat direpresentasikan dengan beberapa istilah, yakni *True Positives (TP)*, *True Negatives (TN)*, *False Positives (FP)*, dan *False Negative (FN)*. TP terjadi ketika model benar memprediksi sesuatu yang memang benar adanya, begitu pun dengan TN yang terjadi ketika model benar memprediksi bahwa sesuatu bukan termasuk kategori tertentu. Sebaliknya FP terjadi ketika model salah memprediksi sesuatu sebagai positif, padahal kenyataannya negatif. FN merupakan hal yang paling fatal karena terjadi ketika model salah memprediksi sesuatu sebagai negatif, padahal kenyataannya positif. Rumus perhitungan dari masing-masing metrik adalah sebagai berikut [5].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

### 3. Hasil dan Diskusi

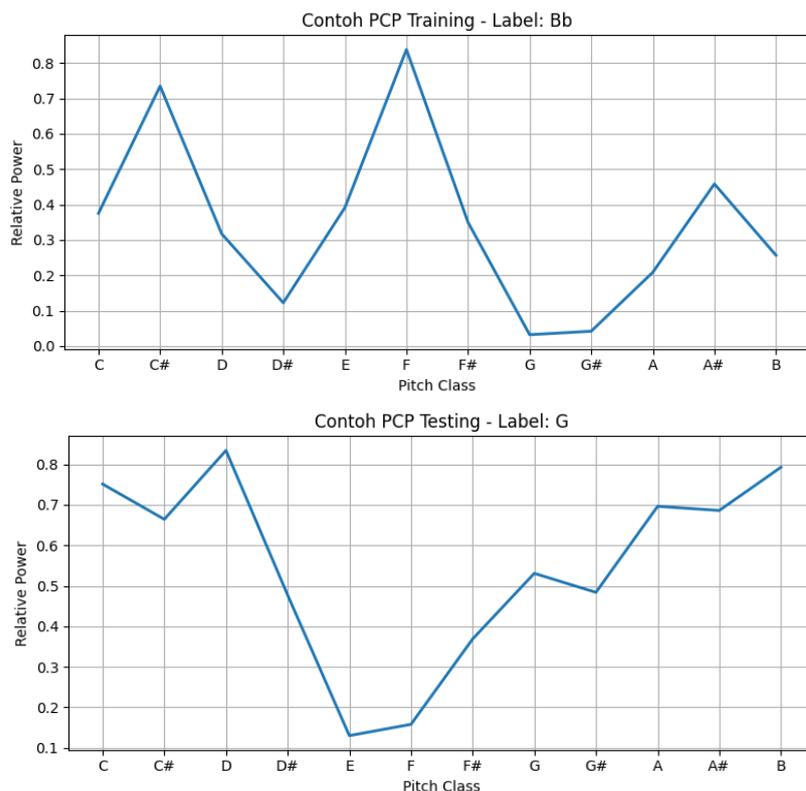
#### 3.1. Perancangan Sistem

##### a. Pra-Pemrosesan Data

Seperti yang sudah disebutkan sebelumnya, dataset *Piano Triads Wavset* ini tidak memerlukan banyak pemrosesan sebelum digunakan untuk melatih dan menguji model. Pada umumnya, dataset yang terkait dengan akor musik dibersihkan dari noise agar tidak mengganggu analisis frekuensi akor [5]. Namun, dalam penelitian ini, hasil akhir yang diinginkan berupa sistem yang menyediakan fitur rekam *audio*, dan karena kemungkinan adanya perbedaan kualitas perangkat, noise yang ada tidak berpengaruh. Dataset tetap dibagi 2 menjadi data *training* dan data *testing*, dengan rasio 80:20, untuk kemudian dinormalisasi dengan *standard scaler*. Selain itu, label akor dasar diambil dari file csv sebagai representasi 12 skala *pitch* standar. Namun, pendekatan ini memiliki kekurangan karena semua variasi akor *major*, *minor*, dan *diminished* disatukan dalam 1 label, sehingga prediksi akor yang diberikan tidak spesifik. Untuk memberikan gambaran yang lebih jelas, dengan memainkan akor *C major*, *minor*, ataupun *diminished*, sistem akan mengenalinya sebagai akor C.

##### b. Ekstraksi Fitur

Fitur *Pitch Class Profile* (PCP) diekstrak dengan menggunakan bantuan *library librosa*. Sistem akan memeriksa setiap bin spektrum dan menjumlahkan energi dari bin yang termasuk ke dalam kelas tersebut. Setelah semua *pitch class* dihitung, akan diperoleh vektor PCP berdimensi 12. Dalam penelitian ini, nilai rata-rata dari seluruh *frame* waktu ini digunakan sebagai representasi PCP akhir untuk setiap *file audio*. Representasi ini selanjutnya digunakan sebagai input fitur ke model klasifikasi. Index 12-bin berupa angka 0-11, namun untuk memudahkan pemahaman akor, index tersebut direpresentasikan dengan 12 skala *pitch* standar seperti pada gambar berikut.



Gambar 5. Visualisasi PCP

Pada Gambar 5, terlihat pada akor dengan label Bb, not yang mendominasi adalah C#, F, dan A#, sehingga jika sistem mendeteksi bentuk yang serupa, maka sistem akan memprediksi akor tersebut sebagai akor Bb, demikian juga dengan akor label G.

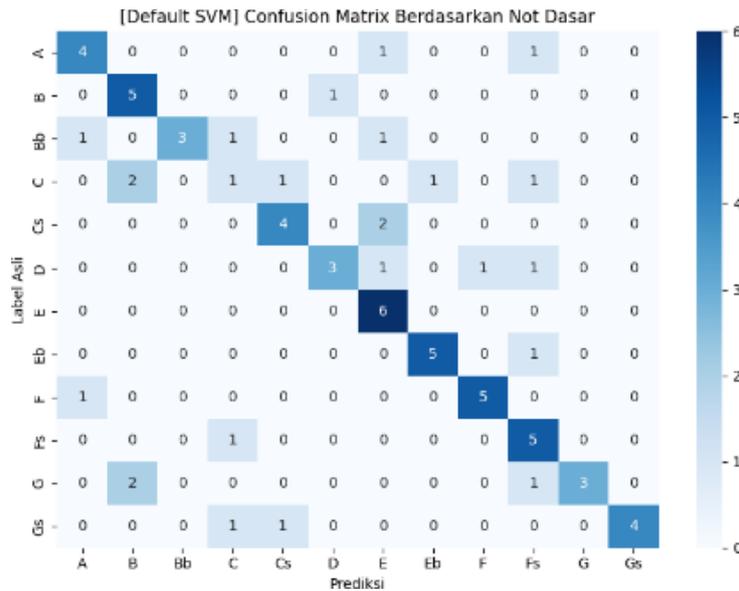
### 3.2. Pengujian Sistem

#### a. Evaluasi Model

Keseluruhan sistem dirancang dengan menggunakan bahasa pemrograman *Python*. Setelah melatih model, performa model SVM kemudian diukur dari akurasi prediksi akor yang dibuat berdasarkan data *testing*. Model ini akan melalui 2 macam evaluasi sebagai perbandingan, pertama dengan konfigurasi *default* dan berikutnya setelah menggunakan *GridSearchCV* sebagai *hyperparameter tuning*. Metrik yang menjadi acuan sesuai dengan yang ada pada *classification report* pada umumnya, yakni *accuracy*, *precision*, *recall*, dan *f1-score*. Berikut merupakan tabel hasil evaluasi beserta dengan *confusion matrix* model *Default SVM*.

**Tabel 2.** Classification Report Default SVM

Kelas	Accuracy	Precision	Recall	F1-Score
A	0.67	0.67	0.67	0.67
B	0.67	0.56	0.83	0.67
Bb	0.67	1.00	0.50	0.67
C	0.67	0.25	0.17	0.20
Cs	0.67	0.67	0.67	0.67
D	0.67	0.75	0.50	0.60
E	0.67	0.55	1.00	0.71
Eb	0.67	0.83	0.83	0.83
F	0.67	0.83	0.83	0.83
Fs	0.67	0.50	0.83	0.62
G	0.67	1.00	0.50	0.67
Gs	0.67	1.00	0.67	0.80

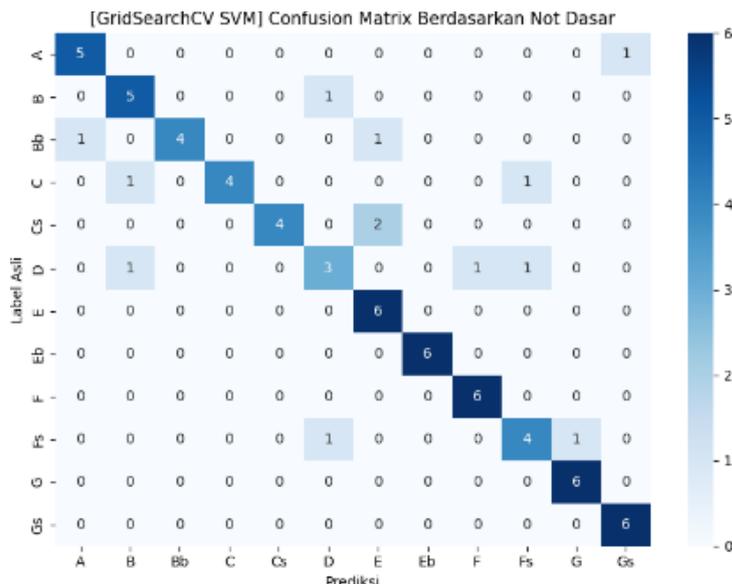


**Gambar 6.** Confusion Matrix Default SVM

Hasil evaluasi pada Tabel 2 dan Gambar 6 menunjukkan nilai *accuracy* yang cukup rendah, yakni pada 67%. Beberapa kelas akor seperti E atau F dapat dikenali dengan baik, sementara kelas lain seperti C menunjukkan hasil yang jauh lebih rendah. Hal ini menunjukkan bahwa model belum dapat membedakan seluruh kelas secara konsisten. Tentunya hal ini tidak menutup kemungkinan akan adanya peningkatan hasil evaluasi setelah melakukan *hyperparameter tuning*. *GridSearchCV* dijalankan dengan *5-fold cross validation*, dengan beberapa pilihan parameter. Pertama nilai regularisasi *gamma* dengan opsi 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 dan C dengan opsi 0.1, 1, 10, 100, 1000. Berikut merupakan tabel hasil evaluasi beserta dengan *confusion matrix* model *GridSearchCV* SVM.

**Tabel 3.** Classification Report GridSearchCV SVM

Kelas	Accuracy	Precision	Recall	F1-Score
A	0.82	0.83	0.83	0.83
B	0.82	0.71	0.83	0.77
Bb	0.82	1.00	0.67	0.80
C	0.82	1.00	0.67	0.80
Cs	0.82	1.00	0.67	0.80
D	0.82	0.60	0.50	0.55
E	0.82	0.67	1.00	0.80
Eb	0.82	1.00	1.00	1.00
F	0.82	0.86	1.00	0.92
Fs	0.82	0.67	0.67	0.67
G	0.82	0.86	1.00	0.92
Gs	0.82	0.86	1.00	0.92

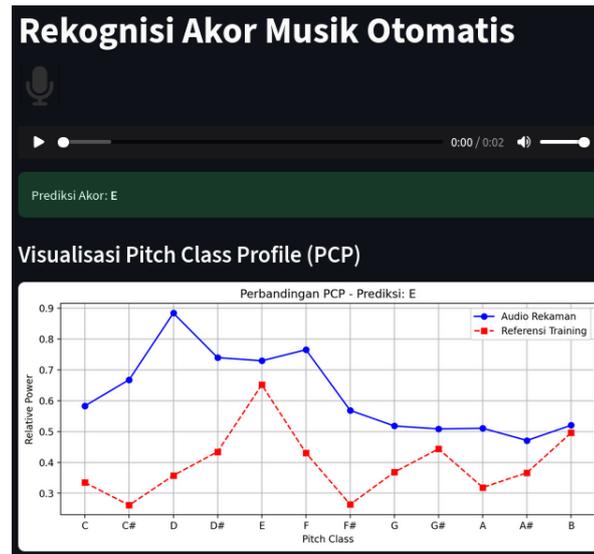


Gambar 7. Confusion Matrix GridSearchCV SVM

Melalui Tabel 3 dan Gambar 7, terbukti bahwa dengan menggunakan *GridSearchCV* sebagai hyperparameter tuning, model dapat menghasilkan *accuracy* yang lebih tinggi yakni sebesar 82% dengan pilihan parameter *gamma* bernilai 0.1 dan *C* bernilai 100. Tidak hanya itu, metrik lain juga mendapat peningkatan yang serupa dan kali ini sudah cukup seragam, banyak kelas akor yang mendapat nilai *precision*, *recall*, dan *f1-score* yang lebih besar dari 0.80, namun terdapat juga kelas yang mengalami penurunan. Walau tidak banyak kelas yang tertukar dalam prediksi, kemungkinan hal ini dapat terjadi karena kemiripan dari bunyi akor. Dengan menyelesaikan tahap evaluasi ini, model terbaik disimpan dan akan digunakan dalam implementasi sistem rekognisi akor otomatis berdasarkan *input* pengguna. Sistem ini diimplementasikan dengan bantuan *framework streamlit* untuk mempermudah visualisasi data dan pembuatan antarmuka sederhana. *Confusion matrix* seperti pada Gambar 6 dan Gambar 7 sebenarnya hanya menghasilkan satu nilai akurasi keseluruhan (*overall accuracy*), bukan akurasi per kelas. Hal ini karena perhitungan akurasi menggunakan rumus Persamaan 4 yang mempertimbangkan semua prediksi yang benar (*True Positive* dan *True Negative*) dibandingkan dengan seluruh jumlah data uji. Jadi akurasi memberikan gambaran seberapa banyak prediksi model yang benar secara total dari semua kelas yang diuji.

b. Eksperimen Instrumen

Tentunya pelatihan model yang tidak sesuai membuat hasil prediksi tidak sepenuhnya akurat apabila digunakan untuk instrumen musik lain. Walau model dilatih hanya dengan data audio instrumen musik piano, namun sistem juga dapat mengenali akor dari instrumen musik lain. Hal ini sejalan dengan gambaran fitur PCP dari instrumen musik piano dan gitar yang serupa, sehingga masih dapat dikenali model [1]. Pendekatan ini dapat ditingkatkan kembali dengan penggunaan dataset yang lebih lengkap, pemrosesan data yang lebih baik, atau model yang sekiranya lebih cocok dalam konteks penelitian terkait rekognisi akor. Gambar 8 menunjukkan pengujian sistem dengan menggunakan instrumen ukulele.



Gambar 8. Pengujian Sistem Menggunakan Ukulele

#### 4. Kesimpulan

Penelitian ini menunjukkan bahwa sistem rekognisi akor musik berbasis *Pitch Class Profile* (PCP) dan *Support Vector Machine* (SVM) dapat memberikan hasil yang cukup akurat dan efisien. Model SVM yang dirancang telah diuji dalam dua skenario, yakni dengan konfigurasi *default* dan optimasi parameter menggunakan *GridSearchCV*. Hasil pengujian SVM dengan konfigurasi *default* menunjukkan bahwa model belum mampu mengklasifikasikan akor dengan baik, terlihat dari ketimpangan nilai metrik evaluasi di antara kelas, dengan rata-rata akurasi masih berada di 67%. Di lain sisi, setelah dilakukan optimasi parameter menggunakan *GridSearchCV*, performa model meningkat menjadi 82%, dengan peningkatan yang stabil pada nilai metrik lain. Menariknya, meskipun sistem ini hanya dilatih dengan piano, model mampu mengenali akor dari instrumen lain dengan akurasi yang tetap dapat diterima. Hal ini mengindikasikan bahwa fitur PCP tidak spesifik terhadap bunyi instrumen tertentu. Ini membuka peluang bagi pengembangan lebih lanjut untuk mengenali akor dari berbagai jenis instrumen. Kesuksesan sistem ini dalam mengenali akor dengan pendekatan sederhana membuktikan bahwa kombinasi antara fitur *chroma* seperti PCP dan model klasik seperti SVM masih sangat relevan. Dengan data yang lebih lengkap dan pelatihan yang lebih rinci, sistem ini dapat berkembang menjadi solusi rekognisi musik yang lebih baik dan praktis.

#### Daftar Pustaka

- [1] J. Osmalsky, J.-J. Embrechts, V. Droogenbroeck, and S. Pierard, "Neural networks for musical chords recognition," *Journées d'informatique musicale*, 2024, doi: <https://hdl.handle.net/2268/115963>.
- [2] Suwatchai Kamonsantiroj, "Chord Recognition in Music Using a Robust Pitch Class Profile (PCP) Feature and Support Vector Machines (SVM)," *IJIIS International Journal of Informatics and Information Systems*, vol. 7, no. 1, pp. 01–07, Jan. 2024, doi: <https://doi.org/10.47738/ijiis.v7i1.191>.
- [3] Z. Rao, X. Guan, and J. Teng, "Chord Recognition Based on Temporal Correlation Support Vector Machine," *Applied Sciences*, vol. 6, no. 5, pp. 157–157, May 2016, doi: <https://doi.org/10.3390/app6050157>.
- [4] D. Mustafa Abdullah and A. Mohsin Abdulazeez, "Machine Learning Applications based on SVM Classification A Review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 81–90, Apr. 2021, doi: <https://doi.org/10.48161/qaj.v1n2a50>.
- [5] G. Varoquaux and O. Colliot, "Evaluating Machine Learning Models and Their Diagnostic Value," *Neuroinformatics*, pp. 601–630, Jan. 2023, doi: [https://doi.org/10.1007/978-1-0716-3195-9\\_20](https://doi.org/10.1007/978-1-0716-3195-9_20).

Halaman ini sengaja dibiarkan kosong