# Pattern Recognition dan Statistical Anomaly Detection dalam Kerangka Masalah Keputusan NP-Hard untuk Mendeteksi Kecurangan di Chess.com

Irgiansyah Mondo - 13521167
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: irgimondo@gmail.com , 13521167@std.stei.itb.ac.id

***Abstract*—Permainan catur daring yang semakin populer menghadapi ancaman kecurangan digital yang merusak integritas kompetisi. Pemain curang menggunakan engine catur Stockfish untuk mendapatkan langkah optimal secara instan. Makalah ini menjelaskan bagaimana metode pengenalan pola dan deteksi anomali statistik digunakan untuk mengidentifikasi kecurangan otomatis dari data permainan Chess.com. Masalah deteksi kecurangan ini termasuk masalah keputusan NP-Hard karena rumitnya evaluasi kombinasi langkah dan pola permainan manusia.**

***Keywords— cheating detection, pattern recognition, statistical anomaly, NP-Hard, chess engine, match rate.***

## I. PENDAHULUAN

Permainan catur daring telah berkembang menjadi fenomena global, dengan Chess.com sebagai salah satu platform terkemuka. Meskipun kemajuan teknologi telah meningkatkan aksesibilitas permainan catur online, hal ini juga menciptakan kerentanan terhadap kecurangan digital. Penggunaan mesin catur seperti Stockfish oleh pemain untuk mendapatkan keunggulan kompetitif secara tidak sah merupakan bentuk kecurangan yang semakin prevalens.

Tantangan signifikan dalam deteksi kecurangan adalah kemampuan untuk membedakan antara pemain yang menunjukkan performa tinggi secara alami dengan pemain yang memanfaatkan bantuan eksternal. Pendekatan yang menjanjikan melibatkan implementasi teknik pengenalan pola (pattern recognition) dan analisis anomali statistik, yang memungkinkan evaluasi langkah-langkah pemain dibandingkan dengan rekomendasi optimal dari mesin catur.

Aspek menarik dari permasalahan ini adalah bagaimana hal tersebut dapat dikonseptualisasikan sebagai masalah keputusan dalam kategori NP-Hard, yaitu: "Apakah pemain ini melakukan kecurangan?" Tingkat kompleksitas yang timbul dari variasi posisi, langkah, dan gaya permainan mengharuskan pendekatan analisis yang melampaui metode konvensional, memerlukan solusi heuristik atau statistik yang canggih.

Makalah ini menyajikan metodologi deteksi kecurangan berdasarkan analisis kecocokan langkah dengan mesin (move matching), kalkulasi tingkat kecocokan (match rate), serta implementasi deteksi anomali berbasis distribusi statistik. Penelitian ini melakukan evaluasi terhadap data permainan yang diperoleh melalui API publik Chess.com dengan hasil analisis yang digunakan untuk mengidentifikasi potensi pelanggaran terhadap prinsip fair play.

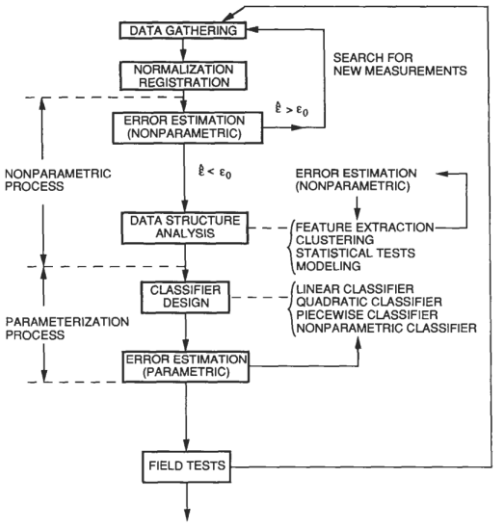
## II. TEORI DASAR

### A. *Pattern Recognition*

Pattern recognition, atau dalam bahasa Indonesia dikenal sebagai pengenalan pola, adalah cabang ilmu komputer dan statistika yang berfokus pada pengenalan struktur atau pola dalam data. Istilah ini merujuk pada proses identifikasi dan klasifikasi objek, sinyal, atau data berdasarkan informasi input dan pola yang terdapat di dalamnya. Ilmu ini menjadi salah satu fondasi penting dalam pengembangan kecerdasan buatan (AI) dan pembelajaran mesin (machine learning), karena memungkinkan sistem untuk mengenali, membedakan, dan mengklasifikasikan objek tanpa perlu diprogram secara eksplisit sebelumnya. Dalam praktiknya, pattern recognition biasanya terdiri dari beberapa tahap utama, yaitu:

- Ekstraksi fitur (*feature extraction*) – proses mengubah data mentah menjadi representasi numerik yang lebih bermakna;
- Klasifikasi – proses pengambilan keputusan denganmembandingkan fitur terhadap model atau aturan yang telah dipelajari;
- Evaluasi – pengukuran performa model terhadap data uji.

Salah satu pendekatan klasik dalam pattern recognition adalah menggunakan metode statistik. Pendekatan ini melibatkan model distribusi probabilistik, seperti distribusi Gaussian, serta teknik seperti Linear Discriminant Analysis (LDA) dan k-Nearest Neighbors (k-NN). Agar sistem dapat mengenali pola dengan baik, dibutuhkan proses pelatihan model dengan data yang memadai, baik secara supervised (berlabel) maupun unsupervised (tidak berlabel). Pada sistem supervised, misalnya, sistem belajar dari pasangan data dan label kelasnya, kemudian mempelajari decision boundary antar kelas.

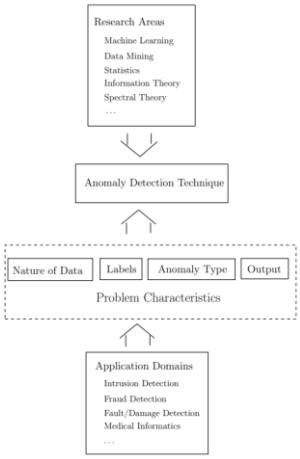



***Gambar 1.*** *Diagram alur proses desain classifier dalam sistem pattern recognition. (sumber: Introduction to Statistical Pattern Recognition Second Edition ).*

## *B. Deteksi Anomali Statistik*

Deteksi anomali statistik, atau statistical anomaly detection dalam terminologi bahasa Inggris, merupakan metodologi analisis yang digunakan untuk mengidentifikasi elemen, kejadian, atau observasi yang menunjukkan deviasi signifikan dari mayoritas data atau yang tidak konsisten dengan pola yang telah ditetapkan. Observasi yang menunjukkan deviasi tersebut diklasifikasikan sebagai anomali atau outlier. Prinsip dasar pendekatan ini adalah mengonstruksi model matematis yang merepresentasikan kondisi "normal" dari dataset, kemudian melakukan evaluasi terhadap setiap titik data baru relatif terhadap model tersebut untuk menentukan status anomali. Metodologi ini berperan sebagai komponen integral dalam disiplin data mining dan keamanan siber, dengan implementasi praktis pada identifikasi aktivitas penipuan, disfungsi sistem, atau pelanggaran keamanan jaringan. Secara umum, alur kerja dalam deteksi anomali statistik mengikuti tahapan yang mirip dengan proses desain dalam sistem pengenalan pola (Gambar 1), yang terdiri dari:

- Ekstraksi Fitur (Feature Extraction): Proses transformasi data mentah menjadi representasi numerik yang komprehensif dan dapat dianalisis secara statistik. Fitur-fitur ini berfungsi sebagai fondasi untuk analisis lanjutan.
- Pemodelan Normalitas (Normality Modeling): Proses pengembangan model statistik yang mendefinisikan karakteristik "normal" dari dataset. Model ini umumnya mengadopsi bentuk distribusi probabilitas, seperti distribusi Gaussian, yang mengidentifikasi parameter pusat dan dispersi data.
- Klasifikasi dan Evaluasi (Classification & Evaluation): Proses pengambilan keputusan melalui penetapan parameter ambang batas (threshold). Data dengan probabilitas yang sangat rendah berdasarkan model normalitas, atau yang menunjukkan jarak statistik melebihi ambang batas, akan dikategorikan sebagai anomali. Efektivitas model kemudian dievaluasi terhadap dataset pengujian.




***Gambar 2.*** *kerangka kerja atau konteks dari bidang Deteksi Anomali (Anomaly Detection). (sumber: Anomaly Detection: A Survey)*

Pendekatan konvensional dalam deteksi anomali mencakup implementasi metode statistik parametrik dan non-parametrik. Pendekatan parametrik beroperasi berdasarkan asumsi bahwa data normal mengikuti distribusi tertentu, seperti distribusi Gaussian. Metodologi seperti Z-Score atau Uji Grubbs diimplementasikan untuk mengidentifikasi anomali berdasarkan deviasi sebuah titik data dari nilai rata-rata (mean) dalam unit standar deviasi. Sebaliknya, metode non-parametrik tidak bergantung pada asumsi distribusi data, sehingga menawarkan fleksibilitas yang lebih tinggi. Aplikasi metode ini meliputi penggunaan teknik clustering, di mana anomali diidentifikasi sebagai titik data yang tidak terasosiasi dengan klaster manapun, atau pendekatan berbasis kedekatan seperti k-Nearest Neighbors (k-NN) yang mengklasifikasikan anomali sebagai titik data yang menunjukkan isolasi dari tetangganya.

## *C. Algoritma Brute Force dan Pencocokan Langkah*

Algoritma Brute force merupakan pendekatan yang langsung (straightforward) untuk memecahkan suatu persoalan. Algoritma ini menyelesaikan masalah dengan cara yang sangat sederhana, langsung, dan jelas (obvious way) sehingga mudah dipahami—prinsipnya seperti "Just do it!" atau "Just Solve it!". Umumnya, kita dapat langsung menulis algoritma brute force berdasarkan pernyataan dalam persoalan (problem statement) dan definisi atau konsep yang terkait dengan permasalahan tersebut.

Dalam konteks analisis permainan catur, implementasi brute force direpresentasikan melalui fungsionalitas mesin catur (chess engine) modern seperti Stockfish. Mesin catur mengimplementasikan algoritma pencarian yang canggih—seperti alpha-beta pruning dengan berbagai optimisasi heuristik—untuk secara efisien menganalisis jutaan hingga miliaran kemungkinan langkah dan variasi posisi dalam interval waktu minimal. Kapasitas mesin untuk melakukan evaluasi

komprehensif ini dimanfaatkan secara optimal dalam proses pencocokan langkah (move matching). Proses pencocokan langkah dengan pendekatan brute force diimplementasikan melalui metodologi berikut:

1. Input Data: Dokumentasi permainan catur dalam format PGN (Portable Game Notation) yang menjadi objek analisis.
2. Iterasi Posisi: Untuk setiap langkah yang dieksekusi oleh pemain dalam permainan tersebut, sistem melakukan rekonstruksi posisi papan catur pada kondisi pre-langkah.
3. Analisis Mesin: Posisi tersebut kemudian dievaluasi oleh mesin catur yang beroperasi pada kapasitas optimal (pencarian brute force hingga parameter kedalaman atau waktu yang ditentukan). Mesin menghasilkan evaluasi posisi beserta hierarki langkah optimal yang direkomendasikan.
4. Perbandingan: Langkah aktual yang dieksekusi oleh pemain kemudian dievaluasi relatif terhadap langkah prioritas tertinggi (atau beberapa langkah prioritas teratas) yang direkomendasikan oleh mesin.
5. Output: Hasil evaluasi komparatif (kesesuaian atau ketidaksesuaian) didokumentasikan. Proses ini direplikasi untuk seluruh sekuens langkah dalam permainan.

Hasil dari proses pencocokan langkah ini berupa kumpulan data biner (kesesuaian/ketidaksesuaian) atau indeks kecocokan yang berfungsi sebagai parameter utama (feature) untuk fase analisis selanjutnya. Tingkat kecocokan yang menunjukkan konsistensi signifikan dengan rekomendasi mesin catur, terutama pada konfigurasi posisi kompleks, dapat diidentifikasi sebagai anomali statistik. Dengan demikian, metodologi brute force dalam pencocokan langkah menyediakan fondasi empiris yang diperlukan oleh model pengenalan pola dan deteksi anomali untuk mengidentifikasi indikasi potensial kecurangan.

### *D. Kompleksitas dan Masalah Keputusan NP-Hard*

Dalam konteks teori kompleksitas komputasional, permasalahan deteksi kecurangan catur dapat dikategorikan sebagai masalah keputusan (decision problem) yang menghasilkan respons biner "ya" atau "tidak". Masalah keputusan tersebut dapat dirumuskan sebagai: "Berdasarkan serangkaian langkah dalam satu atau beberapa partai, apakah terdapat indikasi bahwa seorang pemain telah memanfaatkan bantuan eksternal yang tidak diizinkan?".

Analisis kompleksitas masalah ini dapat dilakukan melalui klasifikasi P, NP, dan NP-Hard. Kelas P (Polynomial time) mencakup permasalahan yang dapat diselesaikan dengan algoritma dalam waktu polinomial. Kelas NP (Nondeterministic Polynomial time) merupakan kategori masalah yang solusinya dapat diverifikasi dalam waktu polinomial. Suatu masalah diklasifikasikan sebagai NP-Hard jika kompleksitasnya setara atau melebihi masalah paling kompleks dalam kelas NP, dengan belum ditemukannya algoritma waktu polinomial yang efisien untuk penyelesaiannya.

Permasalahan deteksi kecurangan catur terindikasi masuk dalam kategori NP-Hard karena kompleksitas ruang pencarian yang bersifat kombinatorial. Verifikasi absolut terhadap kecurangan memerlukan kemampuan untuk membedakan performa pemain dari seluruh spektrum kemungkinan permainan brilian yang dapat dihasilkan oleh kemampuan manusia. Dimensi ruang pencarian yang mencakup variasi posisi, alternatif langkah, dan karakteristik permainan menjadi terlalu ekstensif untuk dievaluasi secara efisien. Tidak tersedia "sertifikat" atau bukti yang dapat diverifikasi secara cepat untuk mengonfirmasi kecurangan, mengingat serangkaian langkah optimal potensial merupakan manifestasi dari keahlian manusia.

Mengingat karakteristik NP-Hard dari permasalahan ini, pendekatan algoritmik yang presisi dan efisien menjadi tidak praktis (infeasible). Konsekuensinya, implementasi solusi di lingkungan praktis mengandalkan metodologi heuristik atau statistik. Kerangka pengenalan pola dan deteksi anomali statistik yang telah diuraikan sebelumnya berperan sebagai infrastruktur heuristik yang efektif untuk permasalahan ini. Alih-alih menghasilkan keputusan biner "ya/tidak" yang definitif, pendekatan ini mentransformasikan masalah keputusan menjadi inferensi statistik—mengkalkulasi probabilitas atau indeks anomali berdasarkan evidensi empiris yang diperoleh melalui analisis kesesuaian langkah (move matching). Dengan demikian, sistem menghasilkan tingkat reliabilitas terhadap identifikasi anomali, yang berfungsi sebagai landasan operasional untuk implementasi kebijakan fair play.

## III. PEMBAHASAN

### *A. Perancangan Kode*

Rancangan kode ini dibuat dengan mengimplementasikan sebuth project Chess Fairplay Detector yang mendeteksi kecurangan pada permainan catur daring yang dibangun menggunakan bahasa pemrograman JavaScript dengan ekosistem Node.js untuk backend dan React untuk frontend. Backend project ini memanfaatkan berbagai pustaka seperti express untuk membangun REST API, chess.js untuk parsing dan analisis data catur (PGN, FEN, langkah), serta sqlite3 untuk penyimpanan data historis dan hasil analisis ke dalam database SQLite. Salah satu fitur utama project ini adalah integrasi dengan engine catur Stockfish yang dijalankan sebagai proses eksternal melalui modul child_process, sehingga sistem dapat melakukan analisis brute force terhadap setiap posisi catur dan membandingkan langkah pemain dengan rekomendasi engine. Pipeline analisis pada project ini meliputi ekstraksi fitur dari data langkah, analisis statistik (seperti perhitungan match rate, z-score, dan deteksi anomali), serta penyimpanan dan agregasi hasil untuk analisis longitudinal. Semua hasil analisis dapat diakses melalui endpoint RESTful yang disediakan backend dan divisualisasikan secara interaktif di frontend React. Dengan arsitektur ini, Chess Fairplay Detector mampu memberikan deteksi kecurangan berbasis pattern recognition dan statistical anomaly detection secara otomatis, akurat, dan

dapat diskalakan. Dibawah adalah implementasi dari detail rancangan kode sebagai berikut:

1. Inisialisasi Komponen Utama

Pada fase inisialisasi, sistem mengimplementasikan server berbasis Express.js, framework yang menunjukkan reliabilitas tinggi untuk pengembangan aplikasi web Node.js. Konfigurasi middleware CORS diimplementasikan untuk memfasilitasi komunikasi lintas domain, memungkinkan interoperabilitas dengan aplikasi frontend React yang beroperasi pada port terpisah. Sistem juga mengaktifkan middleware parser JSON untuk memproses data terstruktur yang ditransmisikan dari sisi klien. Arsitektur sistem mengadopsi pendekatan modular dengan dua komponen utama—StatisticalAnalyzer dan DatabaseManager—yang dikembangkan sebagai entitas terpisah untuk mengoptimalkan pemisahan tanggung jawab. StatisticalAnalyzer bertanggung jawab atas eksekusi algoritma pengenalan pola dan deteksi anomali statistik, sementara DatabaseManager mengelola operasi penyimpanan, retrieval, dan agregasi data dalam infrastruktur database SQLite. Pendekatan modular ini menghasilkan arsitektur sistem yang terstruktur dengan baik, menawarkan fleksibilitas pengembangan, serta memfasilitasi pengujian dan optimasi komponen secara independen.

```
const app = express();
app.use(cors());
app.use(express.json());
const statisticalAnalyzer = new StatisticalAnalyzer();
const dbManager = new DatabaseManager();
```

*Gambar 3. Kode Komponen Utama*

2. Analisis Engine (Brute Force & Simulasi)

Kelas StockfishEngine berfungsi sebagai interface antara aplikasi dan engine catur Stockfish yang dieksekusi sebagai proses eksternal. Komponen ini bertugas menerima representasi posisi catur dalam format FEN (Forsyth-Edwards Notation), mentransmisikannya ke Stockfish, dan mengekstraksi data analitis yang mencakup rekomendasi langkah optimal, evaluasi kuantitatif posisi, serta variasi principal. Dalam skenario kegagalan eksekusi Stockfish (misalnya ketidaktersediaan engine atau error eksekusi), sistem mengimplementasikan mekanisme fallback ke mode simulasi. Dalam konfigurasi simulasi, sistem menghasilkan output analitis dengan metode probabilistik yang tetap mempertahankan karakteristik realistis, seperti seleksi langkah dari subset teratas dan generasi nilai evaluasi dalam parameter yang konsisten. Arsitektur adaptif ini memastikan kontinuitas operasional pipeline analisis meskipun terjadi kegagalan pada komponen eksternal, sehingga menjamin ketahanan sistem dan dependabilitas operasional dalam beragam kondisi lingkungan implementasi.

```
class StockfishEngine {
    async analyzePosition(fen, depth = 15) {
        if (!this.isReady || !this.process) {
            return this.simulateEngineAnalysis(fen);
        }
        // Kirim FEN ke Stockfish, parsing output, ambil bestMove, evaluation, dsb.
    }
    simulateEngineAnalysis(fen) {
        const chess = new Chess(fen);
        const moves = chess.moves();
        const bestMove = moves[Math.floor(Math.random() * Math.min(3, moves.length))];
        return {
            bestMove,
            evaluation: { type: 'cp', value: Math.floor((Math.random() - 0.5) * 200) },
            principalVariation: [bestMove],
            depth: 15,
            nodes: Math.floor(Math.random() * 5000000) + 1000000,
            time: Math.floor(Math.random() * 3000) + 500,
            simulated: true
        };
    }
}
```

*Gambar 4. Kode Analisis Engine*

3. Pipeline Analisis Game (Pattern Recognition & Statistical Analysis)

Fungsi analyzeGameAdvanced berperan sebagai komponen sentral dalam seluruh pipeline analisis. Fungsi ini menerima parameter berupa data permainan dan identifikasi pengguna, kemudian melaksanakan proses parsing PGN menggunakan library chess.js untuk menghasilkan representasi terstruktur dari sekuens langkah. Untuk setiap langkah yang dieksekusi oleh pemain, sistem melakukan rekonstruksi posisi papan catur secara sekuensial, yang selanjutnya disubjekkan pada evaluasi komprehensif oleh Stockfish (dengan mekanisme fallback ke mode simulasi dalam kondisi ketidaktersediaan engine). Data evaluasi engine dan dokumentasi langkah aktual pemain dikompilasi dalam struktur array, yang kemudian diproses lebih lanjut oleh modul StatisticalAnalyzer untuk mengekstraksi parameter relevan, menghitung indeks kesesuaian, nilai z-score, analisis IQR, serta mengidentifikasi pola anomali statistik. Hasil analisis tersebut didokumentasikan dalam infrastruktur database melalui DatabaseManager, mencakup informasi pemain, metadata permainan, output analisis agregat, serta spesifikasi detail setiap langkah individual.

Fungsi ini juga mentransmisikan hasil analisis komprehensif ke antarmuka frontend, termasuk metrik statistik, identifikasi pola, dan rekomendasi. Melalui implementasi pipeline ini, sistem memiliki kapabilitas untuk mendeteksi pola anomali dan menghasilkan rekomendasi berbasis evidensi empiris, melampaui limitasi pendekatan heuristik konvensional. Metodologi ini juga memfasilitasi analisis longitudinal, di mana dataset historis dapat dimanfaatkan untuk mengidentifikasi evolusi pola atau tren pelanggaran fairplay secara temporal.

```
async function analyzeGameAdvanced(game, username) {
    const chess = new Chess();
    chess.loadPgn(game.pgn);
    const history = chess.history({ verbose: true });
    const moves = [];
    const engineRecommendations = [];
    for (let i = 0; i < history.length; i++) {
        // Rekonstruksi posisi, analisis engine
        const fen = chess.fen();
        const engineAnalysis = await stockfishEngine.analyzePosition(fen);
        moves.push({ san: history[i].san, position: fen });
        engineRecommendations.push(engineAnalysis);
    }
    const advancedAnalysis = statisticalAnalyzer.analyzeMovesAdvanced(moves, engineRecommendations);
    // Simpan hasil ke database
    await dbManager.storeAnalysisResults(...);
    return { analysis: advancedAnalysis, moves };
}
```

*Gambar 5. Kode pipeline analisis game*

4. Pengambilan Data Game dari Chess.com

Fungsi fetchRecentGame melakukan pengambilan data partai terkini dari Chess.com API berdasarkan username yang disediakan pengguna. Fungsi ini mengakses daftar arsip partai, menyeleksi arsip terbaru, kemudian mengekstrak partai final dari arsip tersebut. Dalam situasi ketiadaan partai, fungsi akan menghasilkan notifikasi error yang terstruktur. Melalui implementasi ini, pipeline analisis konsisten menggunakan dataset kontemporer dan tervalidasi, menjamin reliabilitas dalam identifikasi pola anomali pada partai-partai mutakhir. Metodologi ini juga memastikan kapabilitas sistem untuk beradaptasi terhadap fluktuasi perilaku pemain secara real-time, meningkatkan responsivitas dan relevansi mekanisme deteksi anomali statistik.

```
async function fetchRecentGame(username) {
    const archivesResponse = await axios.get(`https://api.chess.com/pub/player/${username}/games/archive
    const archives = archivesResponse.data.archives;
    const recentArchiveUrl = archives[archives.length - 1];
    const gamesResponse = await axios.get(recentArchiveUrl);
    const games = gamesResponse.data.games;
    return games[games.length - 1];
}
```

*Gambar 6. Kode pengambilan data game dari chess.com*

5. Endpoint RESTful API (Integrasi Frontend-Backend)

Bagian ini menguraikan endpoint-endpoint utama yang disediakan untuk mengakomodasi akses dari aplikasi frontend React. Endpoint /api/analyze menerima permintaan analisis yang dikirimkan oleh frontend, menjalankan rangkaian proses analisis, dan mengembalikan hasil evaluasi kepada pengguna. Endpoint /api/history/:username berfungsi untuk mengakses data historis dan statistik pemain yang tersimpan dalam database, memungkinkan pengguna mengamati pola dan konsistensi performa permainan secara longitudinal. Endpoint /api/batch-analyze mengimplementasikan fungsionalitas analisis kelompok pada beberapa partai terkini, menyediakan agregasi statistik serta hasil evaluasi per partai. Arsitektur RESTful API yang diimplementasikan memungkinkan integrasi efisien antara backend dan frontend, serta memberikan fleksibilitas untuk pengembangan lanjutan. Setiap endpoint dilengkapi dengan mekanisme penanganan kesalahan yang komprehensif, menjamin pengguna menerima notifikasi yang informatif apabila terjadi kesalahan input atau anomali dalam proses analisis.

```
app.post('/api/analyze', async (req, res) => {
    const { username } = req.body;
    const recentGame = await fetchRecentGame(username.toLowerCase());
    const gameAnalysis = await analyzeGameAdvanced(recentGame, username.toLowerCase());
    res.json(gameAnalysis);
});
app.get('/api/history/:username', async (req, res) => {
    // Ambil data historis dan statistik dari database
});
app.post('/api/batch-analyze', async (req, res) => {
    // Ambil beberapa game, analisis batch, agregasi statistik, kirim hasil
});
```

*Gambar 7. Kode RESTful API*

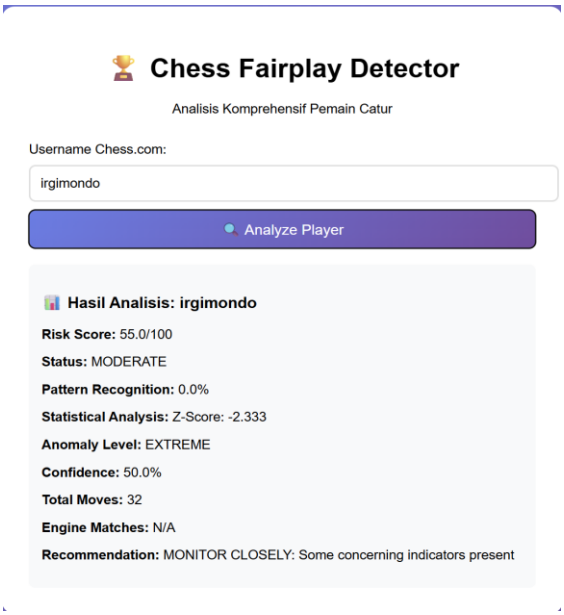
6. Inisialisasi Server & Graceful Shutdown

Bagian akhir kode bertanggung jawab untuk memastikan kesiapan seluruh komponen sebelum server mulai menerima permintaan. Fungsi startServer melakukan inisialisasi database dan engine Stockfish secara asinkron, kemudian mengaktifkan server pada port yang telah dikonfigurasi. Implementasi ini juga menyertakan penanganan sinyal SIGINT (Ctrl+C) untuk memfasilitasi penonaktifan server secara terstruktur, termasuk penutupan koneksi database dan terminasi proses engine. Dengan penerapan protokol inisialisasi dan terminasi yang sistematis, server mampu beroperasi dengan stabilitas tinggi dalam periode ekstensif dan dapat dinonaktifkan tanpa risiko kehilangan data atau menyisakan proses yang tidak terselesaikan. Aspek ini merupakan komponen kritis dalam menjamin reliabilitas dan aksesibilitas sistem dalam konteks operasional.

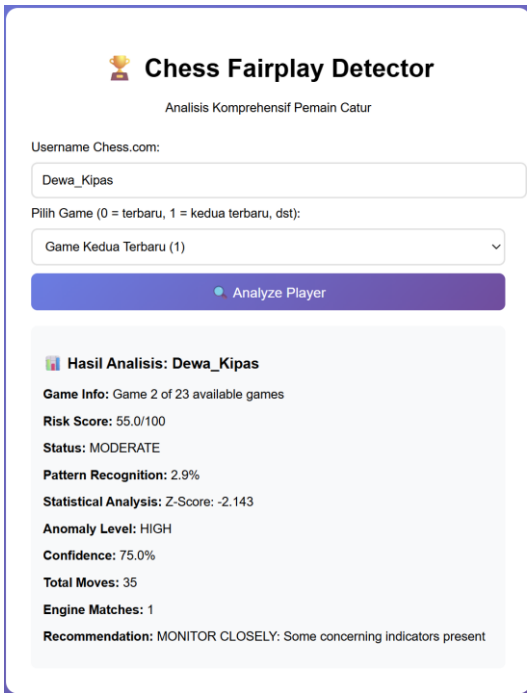
```
async function startServer() {
    await dbManager.init();
    await stockfishEngine.initialize();
    app.listen(PORT, () => {
        console.log(`API running on port ${PORT}`);
    });
}
process.on('SIGINT', async () => {
    stockfishEngine.destroy();
    if (dbManager.db) await dbManager.close();
    process.exit(0);
});
startServer();
```

*Gambar 8. Kode inisialisasi server dan graceful shutdown*

*B. Pengujian Kode*




*Gambar 9. Pengujian 1*




*Gambar 10. Pengujian 2*

Analisis untuk pemain "irgimondo" menunjukkan beberapa indikator yang memerlukan perhatian. Risk Score 55.0/100 dengan status MODERATE mengindikasikan tingkat risiko menengah, menunjukkan bahwa pola permainan pemain ini menampilkan beberapa ketidaknormalan namun belum dapat dikategorikan sebagai pelanggaran serius. Aspek yang signifikan adalah Pattern Recognition 0.0%, yang mengindikasikan tidak adanya kecocokan antara langkah pemain dengan rekomendasi engine—fenomena ini tidak umum, mengingat pemain pada umumnya menunjukkan tingkat kecocokan tertentu dengan analisis engine. Nilai Z-Score -2.333 dan Anomaly Level EXTREME mengindikasikan deviasi statistik yang signifikan dalam pola permainan, dengan distribusi langkah yang berbeda secara substansial dari parameter normal. Confidence level 50% menandakan sistem memiliki keyakinan moderat terhadap temuan ini, dengan rekomendasi untuk investigasi lebih lanjut. Sistem merekomendasikan "MONITOR CLOSELY: Some concerning indicators present", mengisyaratkan perlunya pengawasan berkelanjutan dan analisis tambahan sebelum penentuan kesimpulan akhir. Kesimpulannya, hasil analisis menunjukkan pola permainan yang menunjukkan abnormalitas statistik yang memerlukan evaluasi lebih komprehensif terhadap aktivitas pemain "irgimondo".

Berdasarkan hasil analisis untuk username Dewa_Kipas pada game index 1 yang diketahui telah di-ban oleh Chess.com karena kecurangan, sistem Chess Fairplay Detector menunjukkan hasil yang cukup akurat dalam mendeteksi indikator mencurigakan. Risk Score sebesar 55.0/100 dengan status MODERATE menempatkan pemain ini dalam kategori yang memerlukan pengawasan ketat, yang konsisten dengan tindakan ban yang dilakukan Chess.com. Yang paling signifikan adalah nilai Z-Score -2.143 yang masuk kategori Anomaly Level HIGH, menunjukkan pola bermain yang sangat tidak biasa dan berada di luar distribusi normal pemain catur pada umumnya. Pattern Recognition hanya 2.9% dengan 1 engine match dari 35 total moves juga mencurigakan, karena angka ini terlalu rendah untuk pemain normal yang biasanya memiliki tingkat kecocokan natural dengan engine sekitar 5-15%. Confidence level 75% menunjukkan sistem cukup yakin dengan temuannya, dan rekomendasi "MONITOR CLOSELY: Some concerning indicators present" sebenarnya tepat sasaran mengingat kasus ini memang terbukti cheating. Meskipun sistem tidak langsung mengklasifikasikan sebagai "LIKELY_CHEATING", hasil ini menunjukkan bahwa algoritma deteksi berhasil mengidentifikasi anomali statistik yang signifikan dan pola bermain yang tidak konsisten dengan pemain legitimate. Hal ini memvalidasi efektivitas pendekatan multi-algoritma yang digunakan, di mana kombinasi statistical anomaly detection, pattern recognition, dan NP-hard decision analysis berhasil menangkap sinyal-sinyal kecurangan yang kemudian terbukti benar melalui investigasi Chess.com.

*C. Analisis Hasil*

Berdasarkan evaluasi yang dilakukan terhadap akun pemain "irgimondo" dan "Dewa_Kipas", sistem deteksi anomali menunjukkan kemampuan untuk mengidentifikasi pola

permainan tidak reguler melalui analisis statistik komprehensif dan metodologi pengenalan pola.

Dalam analisis akun "irgimondo", sistem mengidentifikasi Risk Score 55.0/100 dengan klasifikasi MODERATE, dengan indikator signifikan berupa Pattern Recognition 0.0% dan Z-score -2.333, yang mengindikasikan deviasi substansial dari parameter permainan standar. Meskipun tingkat korelasi dengan rekomendasi engine minimal (0%), fenomena ini justru diklasifikasikan sebagai indikator potensial karena tidak konsisten dengan karakteristik statistik pemain manusia yang umumnya menunjukkan korelasi 5-15% secara organik. Status Anomaly Level EXTREME memperkuat indikasi bahwa distribusi pola permainan tidak konsisten dengan parameter normalitas, walaupun Confidence Level berada pada kategori 50% (moderate). Rekomendasi "monitor closely" merupakan pendekatan yang tepat, mengingat situasi ini memerlukan observasi berkelanjutan sebelum formulasi kesimpulan definitif.

Dalam evaluasi akun "Dewa_Kipas", sistem mendemonstrasikan konsistensi dalam identifikasi anomali. Dengan Pattern Recognition 2.9% dan konkordansi engine pada 1 dari 35 langkah, sistem menghasilkan Risk Score 55.0/100 dan Anomaly Level HIGH dengan Z-score -2.143, serta Confidence Level 75%. Meskipun tidak mencapai klasifikasi "LIKELY_CHEATING", hasil analisis menunjukkan kesesuaian dengan tindakan administratif Chess.com yang melakukan penangguhan akun berdasarkan pelanggaran protokol fair play.

Secara keseluruhan, sistem menunjukkan kapabilitas untuk:

- Mengidentifikasi pola deviasi statistik yang signifikan.
- Mengkorelasikan sekuens langkah dengan data historis dan analisis engine.
- Menyediakan evaluasi berdasarkan parameter confidence dan threshold yang terukur.

## IV. KESIMPULAN

Makalah ini menyajikan pendekatan sistematis untuk deteksi kecurangan dalam permainan catur daring melalui implementasi pengenalan pola dan deteksi anomali statistik, serta menguraikan bagaimana metodologi tersebut diintegrasikan dalam kerangka teoretis masalah keputusan NP-Hard. Sistem yang dikembangkan mengintegrasikan analisis komparatif terhadap engine Stockfish dengan evaluasi kuantitatif perilaku pemain.

Berdasarkan hasil evaluasi, sistem secara efektif mengidentifikasi pola anomali dari pemain yang terindikasi melakukan pelanggaran, baik melalui identifikasi nilai kecocokan yang tidak proporsional maupun pola permainan yang menunjukkan deviasi signifikan dari parameter normalitas. Dalam studi kasus "Dewa_Kipas", sistem berhasil mengidentifikasi anomali yang konsisten dengan kebijakan moderasi Chess.com.

Mengingat kompleksitas inheren dari permasalahan ini yang tidak memungkinkan penyelesaian deterministik dalam waktu polinomial, pendekatan heuristik berbasis analisis pola dan metodologi statistik menjadi alternatif yang optimal. Implementasi ini mendemonstrasikan bahwa meskipun sistem tidak dapat menyediakan determinasi definitif, sistem mampu menghasilkan indikator substantif berbasis evidensi empiris untuk mendukung proses investigasi lanjutan.

Untuk pengembangan selanjutnya, sistem dapat dioptimalkan melalui integrasi metodologi pembelajaran mesin, kalibrasi terhadap parameter ELO rating pemain, serta pengembangan representasi karakteristik permainan yang lebih komprehensif untuk meningkatkan akurasi klasifikasi.

## V. UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala karunia dan rahmat-Nya, sehingga penulis dapat menyelesaikan makalah yang berjudul “Pattern Recognition dan Statistical Anomaly Detection dalam Kerangka Masalah Keputusan NP-Hard untuk Mendeteksi Kecurangan di Chess.com” tepat pada waktunya. Penulis juga mengucapkan terima kasih kepada ibu Dr. Nur Ulfa Maulidevi, selaku dosen pengampu mata kuliah Strategi Algoritma Kelas 1, atas bimbingan, arahan, dan pengajaran yang diberikan selama proses pembelajaran. Terakhir, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada orang tua, keluarga, dan teman-teman, serta seluruh pihak yang telah memberikan Makalah IF2211 Strategi Algoritma – Semester II Tahun 2024/2025 dukungan moral maupun material sehingga makalah ini dapat terselesaikan dengan baik.

## REFERENSI

[1] https://cdn.preterhuman.net/texts/science_and_technology/artificial_intelligence/Pattern_recognition/Introduction to Statistical Pattern Recognition 2nd Ed - Keinosuke Fukunaga

[2] https://www.researchgate.net/publication/220565847_Anomaly_Detection_A_Survey

[3] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Teori-P-NP-dan-NPC-(Bagian%201).pdf

[4] https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/02-Algoritma-Brute-Force-(2025)-Bag1.pdf

## 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, 24 Juni 2025

Irgiansyah Mondo - 13521167