

Penerapan Web Speech API pada Game Catur Berbasis Suara

I Gede Hendra Parwata, I Ketut Gede Darma Putra, Ni Putu Sutramiani

Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Udayana
Bukit Jimbaran, Bali, Indonesia, telp. (0361) 701806

e-mail: hendrape@gmail.com, darma.putra@unud.ac.id, sutramiani@unud.ac.id

Abstrak

Catur adalah sebuah permainan strategi papan yang dimainkan pada petak 8x8. Catur awalnya dimainkan langsung secara manual oleh dua orang pemain (manusia), namun sekarang permainan sudah bisa dilakukan secara digital antara manusia melawan komputer (AI). Komputer catur saat ini bahkan sudah bisa mengungguli juara dunia catur, namun pengembangan teknologi catur masih terus terfokus untuk menciptakan AI yang semakin cerdas. Aplikasi yang dikembangkan pada penelitian ini berfokus pada teknologi input dari aplikasi catur yang sudah ada, yang mana memungkinkan pengguna untuk memindahkan bidak pada permainan digital hanya dengan menggunakan perintah suara baik dalam Bahasa Indonesia atau Bahasa Inggris dengan memanfaatkan Web Speech API. Pengguna dapat menggunakan perintah suara pada permainan melawan AI atau secara online antar pengguna. Perintah yang digunakan bisa berupa notasi catur baku atau kalimat biasa, misalnya "Gxe5" atau "Gajah makan pion" sesuai dengan kemungkinan langkah yang tersedia.

Keywords: speech to text, web speech API, chess, AI, web worker

Abstract

Chess is a two-player strategy game played on an 8x8 squares chessboard. Chess was played manually by two persons, but now it's can be played using computer, against another person or AI (Artificial Intelligence). Chess engine has far surpassed even world chess champion, but the development of digital chess still focusing on creating an even stronger engine. The application developed on this research is focusing more on input method, enable user to play digital chess using voice command, moving any legal pieces on the board simply using voice command in Bahasa or English using Web Speech API technologies. Player can use voice input method in a user vs computer (AI) game provided by chess engine or user vs user online. The command used to move the piece can be in standard notation or simple sentence, for example "Bxe5" or "Bishop take pawn" according to the possibilities.

Keywords: speech to text, web speech API, chess, AI, web worker

1. Pendahuluan

Catur adalah permainan yang sudah ada sejak abad ke-6 sebelum masehi dan saat ini menjadi permainan papan paling populer di dunia dengan lebih dari 700 juta pemain, dimainkan oleh dua pemain pada papan dengan 64 petak yang diatur 8x8 (1). Catur juga mengalami perkembangan seiring dengan berkembangnya ilmu komputasi, pada Tahun 1912 Leonardo Torres y Quevedo membangun sebuah mesin yang bisa memainkan pertandingan babak akhir catur antara Raja dan Benteng melawan Raja secara otomatis dengan tingkat keberhasilan pasti (2). Penemuan mesin catur otomatis menjadi catatan penting dalam sejarah perkembangan komputer catur meski belum bisa melakukan perhitungan untuk melakukan permainan secara penuh dari awal dengan semua bidak sesuai susunan permainan catur asli. Kemungkinan kombinasi langkah yang terdapat pada catur sangat banyak sehingga kekuatan komputasi dan teknologi pada waktu itu masih belum memungkinkan untuk melakukan permainan secara penuh (3).

Perkembangan pesat dunia komputasi pada 1950-an membuat penggiat catur, teknisi komputer, dan ilmuwan komputer membangun mesin dan program komputer catur dengan peningkatan tingkat keseriusan dan keberhasilan. Analisis komputer berkontribusi signifikan terhadap teori catur, terutama di babak akhir. Penyelesaian akan permasalahan terkait dijelaskan

oleh Shanon pada jurnal berjudul "Programming a Computer for Playing Chess" yang diterbitkan Tahun 1950 (4).

Dua program catur yang benar-benar sudah bisa digunakan untuk bermain secara penuh dikembangkan pada Tahun 1957, yang pertama oleh Alex Bernstein dan satu lagi oleh kelompok programmer Rusia menggunakan BESM. Algoritma komputer catur terus dikembangkan selama beberapa dekade, hingga pada Tahun 1997, untuk pertama kalinya komputer catur bisa mengalahkan juara dunia catur pada waktu itu yaitu Garry Kasparov dalam pertandingan 6 babak (5).

Komputer catur terus dikembangkan, namun sayang sebagian besar pengembangan aplikasi catur terfokus pada algoritma untuk mencapai kemenangan seperti yang sudah dibahas pada penelitian-penelitian sebelumnya.

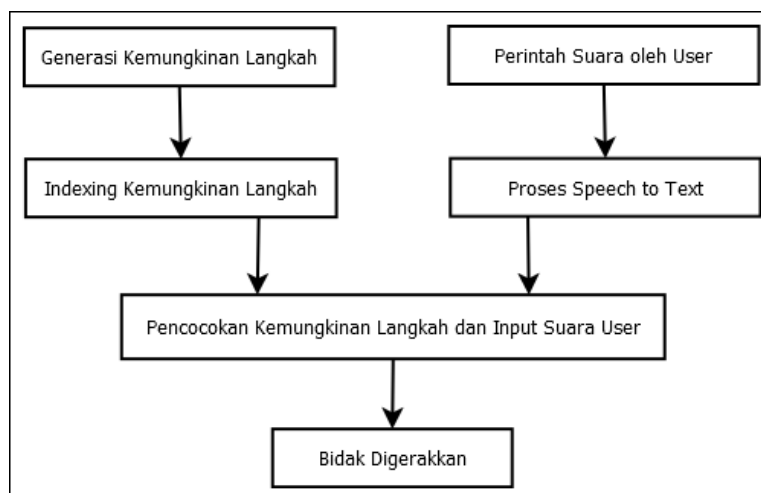
Beberapa perkembangan di bidang komputer catur selain algoritma yaitu perkembangan tampilan dan metode *input*. Tampilan permainan catur pada komputer dimulai dari 2D dan berlanjut sampai 3D. Metode *input* pada aplikasi catur dapat menggunakan *mouse* atau *keyboard*, sampai gestur dengan teknologi Kinect (6). Metode input game catur lain bisa menggunakan kontroler Wii, yaitu perangkat nirkabel yang bisa mengirim dan menerima data menggunakan media Bluetooth (7). Metode input lain yaitu menggunakan AR (Augmented Reality), yaitu teknologi yang mampu mensimulasikan suatu lingkungan virtual sehingga dapat berinteraksi dengan lingkungan virtual yang disimulasikan oleh computer (8). Metode input yang saat ini masih belum banyak dikembangkan yaitu metode input menggunakan suara. Pendekatan yang digunakan dalam penelitian ini yaitu pendeteksi suara menggunakan Web Speech API yang dapat mendeteksi kalimat dalam Bahasa Indonesia sehingga dapat diproses ke dalam perintah untuk menggerakkan bidak catur pada aplikasi. Pengguna yang tidak terbiasa dengan notasi catur dapat menggunakan perintah sederhana seperti kuda makan kuda, ataupun tetap dengan menggunakan notasi catur misalnya kuda makan f3.

2. Metodologi Penelitian

Metodologi Penelitian yang digunakan pada proses pembuatan aplikasi catur berbasis suara menggunakan Web Speech API yaitu pencarian studi literatur, perancangan gambaran umum sistem, proses pengembangan aplikasi, pengujian aplikasi dengan menggunakan web, serta perbaikan kembali sesuai hasil yang ditemukan pada pengujian.

2.1 Gambaran Umum Sistem

Aplikasi catur yang dikembangkan dalam penelitian ini terbagi menjadi dua tipe permainan yaitu *single player* (melawan AI) dan *multi player (online)*. AI yang digunakan yaitu mesin catur standar yang bersifat *open source* yaitu Stock Fish. Permainan *online* dapat dilakukan oleh dua pemain dari dua perangkat yang berbeda dengan mengakses aplikasi yang sudah diunggah ke *web*. Kedua pemain hanya perlu membuat kesepakatan dengan menentukan kode papan tempat mereka ingin bermain, kemudian keduanya bisa saling memilih posisi sebagai pemain putih atau hitam dan melakukan permainan secara *online*. Permainan *single player* dan *multi player* sama-sama bisa dilakukan dengan menggunakan perintah suara.



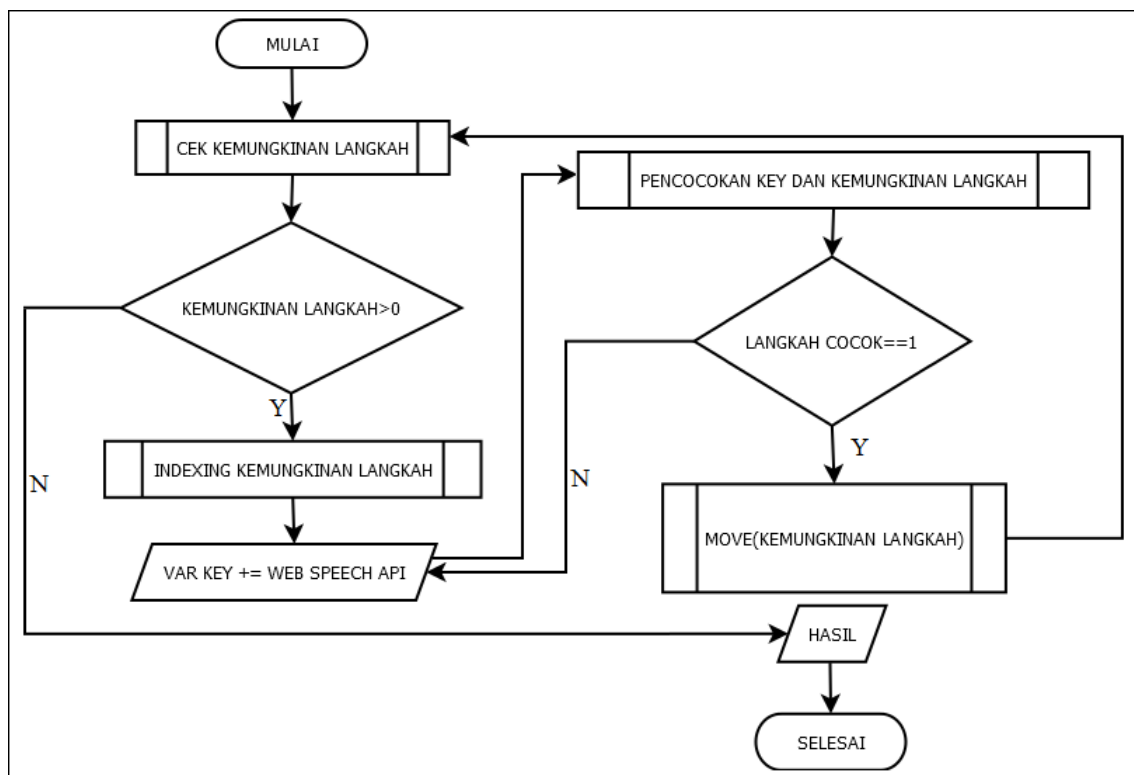
Gambar 1. Gambaran Umum Sistem

Sistem melakukan generasi kemungkinan langkah untuk memperoleh semua langkah yang valid untuk dilakukan pada giliran saat proses generasi dilakukan menggunakan fungsi dari Chess.js. Permainan terus berlanjut selama masih terdapat kemungkinan langkah yang bisa digenerasi. Kemungkinan langkah yang dihasilkan masih berupa notasi sehingga harus dilakukan proses *indexing* untuk mengubah notasi-notasi hasil menjadi dasar pencarian yang dapat dicocokkan pada perintah suara.

Perintah diberikan menggunakan suara yang kemudian diubah menjadi teks pada proses *speech to text* yang dilakukan oleh Web Speech API. Hasil perintah yang sudah berupa teks dicocokkan pada kemungkinan langkah yang sudah melalui proses *indexing*. Bidak digerakkan ketika hanya terdapat satu kemungkinan langkah yang cocok terhadap perintah suara. Proses pencocokan terus diulangi apabila belum ada kemungkinan langkah yang ditemukan atau kemungkinan langkah yang ditemukan berjumlah lebih dari satu. Kata kunci dari teks hasil perintah suara dihapus dan menunggu perintah ulang apabila tidak ada kemungkinan langkah yang ditemukan dan menunggu kata kunci baru apabila kemungkinan yang ditemukan masih berjumlah lebih dari satu. Perintah dengan kata "pion" pada awal permainan menghasilkan 16 kemungkinan langkah yang cocok, jadi sistem masih menunggu kata berikutnya. Perintah dengan kata "pion a" pada awal permainan menghasilkan 2 kemungkinan langkah yang cocok, jadi sistem masih menunggu kata berikutnya, sampai akhirnya ketika pengguna menggunakan perintah "pion a 4" bidak baru dijalankan karena hanya tersisa satu kemungkinan langkah.

2.2 Algoritma Sistem

Algoritma sistem aplikasi catur berbasis suara menggunakan Web Speech API secara umum dapat dilihat pada diagram alir yang terdapat pada Gambar 2.



Gambar 2. Diagram Alir Umum Sistem

Proses pertama yang dilakukan ketika permainan dimulai adalah pengecekan kemungkinan langkah menggunakan fungsi `.moves()` pada Library Chess.js, sekaligus dicek apakah masih terdapat kemungkinan langkah atau tidak. Permainan terus dilanjutkan selama kemungkinan langkah masih tersedia dan sebaliknya berhenti apabila sudah tidak ada lagi kemungkinan langkah yang dihasilkan. Kemungkinan langkah bisa tidak tersedia lagi apabila salah satu pihak sudah berada pada posisi skakmat atau remis.

Setiap kemungkinan langkah yang diperoleh melalui fungsi `.moves()` diindeks pada variabel indeks yang menjadi acuan pada saat dilakukan proses perintah menggunakan suara. Perintah suara diterima menggunakan Web Speech API, hasil suara diproses kemudian dicocokkan dengan kemungkinan-kemungkinan langkah yang terdapat pada indeks. Proses input suara diulangi kembali apabila kemungkinan langkah yang ditemukan tidak bernilai 1 sampai hanya terdapat 1 kemungkinan langkah. Misalnya ketika pemain menyebut kata kuda, dan di papan masih terdapat 4 kemungkinan langkah kuda, maka selanjutnya pemain dapat menyebut kata tambahan misalnya maju, jadi kalimat perintah yang diproses yaitu kuda maju. Kemungkinan langkah hasil digerakkan menggunakan fungsi `.move(langkah)` ketika kemungkinan yang cocok bernilai 1. Giliran melangkah diganti dan proses kembali diulang dari pengecekan kemungkinan langkah.

3. Kajian Pustaka

Kajian Pustaka berisi penunjang yang menjadi landasan teori untuk mendukung pembuatan aplikasi catur berbasis suara menggunakan Web Speech API.

3.1 Web Speech API

Web Speech API adalah JavaScript IP yang dikeluarkan oleh Google bersamaan dengan peluncuran Google Chrome versi 25 pada Februari 2013. Web Speech API memungkinkan pengguna menambahkan fitur pengenalan suara pada aplikasi *web* mereka. Teknologi Web Speech oleh Google diterapkan pada Android dan aplikasi *desktop* untuk melakukan pencarian menggunakan suara, navigasi pada GPS, dan pengembangan aplikasi-aplikasi Android. Penerapan teknologi pembelajaran mendalam *neural network* pada 2015 membuat teknologi Web Speech menjadi semakin akurat. Google mengklaim bahwa mereka memiliki investasi terbaik dalam *machine learning* dan tingkat kesalahan deteksi suara menurun dari 23 persen menjadi 8 persen (9).

3.2 Chess.js

Chess.js adalah sebuah JavaScript *library* yang dapat menghasilkan kemungkinan langkah catur dan mengelola permainan catur berdasarkan input teks dan juga memprosesnya sebagai teks tanpa visual. Hasil yang diperoleh dari *library* ini berupa FEN yang bisa dikelola dengan aplikasi-aplikasi yang menerapkan tatapmuka standar aplikasi catur.

3.3 Chessboard.js

Chessboard.js adalah sebuah JavaScript *library* yang dapat menampilkan dan mengelola *output* visual posisi permainan catur digital dalam notasi FEN. Forsyth–Edwards Notation (FEN) adalah sebuah notasi standar untuk menggambarkan posisi papan dalam permainan catur. FEN bertujuan untuk menyediakan semua informasi yang diperlukan untuk memulai kembali permainan pada posisi tertentu. FEN didasari oleh sebuah sistem yang dikembangkan oleh seorang jurnalis Skotlandia, David Forsyth. Sistem Forsyth menjadi terkenal pada abad ke-19, kemudian Steven J. Edwards mengembangkannya hingga bisa diterapkan pada komputer. FEN bisa disebut sebagai bagian dari Portable Game Notation (PGN) untuk permainan catur karena FEN digunakan untuk mendefinisikan posisi tertentu, bukan langkah secara keseluruhan. FEN terdiri dari 6 bagian yang dipisahkan oleh spasi.

3.4 StockFish Engine

Stockfish adalah salah satu UCI *chess engine* gratis dan bersifat *open source*, yang dapat dijalankan di berbagai aplikasi *desktop* dan *mobile*. Stockfish dikembangkan Marco Costalba, Joona Kiiski, Gary Linscott dan Tord Romstad, serta berbagai kalangan yang turut berkontribusi dalam komunitas pengembang *open source*. *Engine* catur pada dasarnya menerapkan algoritma alpha-beta pruning. Algoritma Alpha-beta Pruning adalah algoritma untuk menentukan hasil terbaik dari langkah masing-masing pemain secara bergantian yang merupakan pengembangan dari algoritma MiniMax. Algoritma Alpha-beta Pruning lebih cepat dari MiniMax karena memotong cabang-cabang kemungkinan yang dianggap tidak perlu untuk dijelajahi (10).

3.5 Ajax

Ajax adalah teknologi yang memungkinkan *browser* untuk memuat data secara realtime tanpa perlu memuat halaman. Teknologi Ajax bisa dimanfaatkan untuk melakukan permainan *online* pada dua perangkat yaitu dengan saling mengirim dan membaca data dari satu *resource* dan kemudian memuatnya pada perangkat masing-masing (11). Data dari sumber disimpan dalam format JSON (JavaScript Object Notation) yaitu format pertukaran data ringan yang membuat proses membaca dan menulis pengguna menjadi lebih mudah serta dapat digenerasi secara otomatis oleh computer (12).

4. Hasil dan Pembahasan

Hasil dan Pembahasan memuat hasil dari aplikasi catur berbasis suara menggunakan Web Speech API baik pada aplikasi *single player* maupun *multi player*.

4.1 Deteksi Perintah Suara dan Cek Kemungkinan Langkah

Deteksi perintah suara dilakukan menggunakan fungsi `listen()`, dari `AudioListener()` yang diperoleh dari Web Speech API. Variabel `lan` adalah variabel tempat menyimpan bahasa deteksi, yang pada aplikasi ini diatur berupa "id" (Bahasa Indonesia) atau "en" (English) pada *form* tersembunyi.

Hasil deteksi dikembalikan sebagai variabel `text`, yang diubah menjadi huruf kecil untuk mempermudah proses pengolahan kata. Variabel `text` ditampilkan dan disimpan pada kolom teks, kemudian diproses pada fungsi `cekKemungkinan()`.

Fungsi `cekKemungkinan()` berjalan menggunakan data perintah dari variabel teks yang terdapat pada kolom dengan `id` teks. Variabel kemungkinan berupa *array* yang diperoleh dari fungsi bawaan `Chess.js` yaitu `game.moves()`. Variabel cocok untuk menentukan nilai kecocokan mula-mula diset 0 sebelum proses pencocokan dimulai.

Proses perulangan dilakukan sesuai jumlah kemungkinan yang diperoleh dari fungsi `game.moves()`. Setiap kemungkinan yang diperoleh dari fungsi `game.moves()` diubah terlebih dahulu menjadi kata pengganti notasi, misal "R" diganti menjadi "benteng rook". Notasi "Ra5" berubah menjadi "benteng rook a 5". Kalimat pasca perubahan yang dihasilkan dicocokkan dengan kata-kata yang diperoleh dari variabel teks.

4.2 Pencocokan Kata Kunci dan Kemungkinan Langkah

Proses pencocokan dilakukan per kata, jadi dilakukan perulangan sejumlah kata pada variabel teks. Tiap kata dicocokkan dengan kalimat kunci pada kemungkinan dalam perulangan pertama. Nilai cocok ditambah 1 apabila kata terdapat pada kalimat kunci, sebaliknya nilai kecocokan dikurang dengan nilai dalam jumlah yang besar misalnya sembilan ribu jika kata tidak ditemukan pada kalimat kunci, dengan kata lain kemungkinan tidak bernilai cocok jika terdapat satu saja kata yang tidak terdapat pada kalimat kunci.

Jumlah kemungkinan yang bernilai cocok ditampilkan sebagai pemberitahuan pada kolom dengan `id` kecocokan. Variabel teks dikosongkan apabila jumlah kemungkinan yang cocok bernilai 0. Perintah berikutnya ditambahkan pada perintah sebelumnya apabila jumlah kemungkinan yang cocok lebih dari 1. Fungsi `game.move()`, `board.move()`, dan *worker* untuk engine apabila mode permainan solo dijalankan apabila jumlah kemungkinan yang cocok tepat bernilai 1.

4.3 Proses Menghapus, Menyimpan, dan Membaca Posisi Multi Player

Proses menghapus, menyimpan, dan membaca posisi dilakukan pada permainan *multi player* secara *online* untuk melakukan sinkronisasi posisi pada kedua perangkat pemain.

Proses penghapusan langkah dilakukan ketika seorang pemain menarik kembali langkahnya. Data FEN yang telah tersimpan pada file `papan.txt` dibaca dari bawah ke atas menggunakan fungsi `array_reverse()` untuk mencari kode papan sesuai data yang diperoleh dari POST. Dua baris yaitu kode papan dan informasi FEN terbaru dari kode yang diinginkan dihapus sehingga informasi langkah kembali menggunakan FEN pada posisi sebelumnya.

Proses pengeditan posisi papan pada file tidak dilakukan dengan mengubah FEN pada posisi sebelumnya, melainkan menambahkan FEN baru dan tetap membiarkan FEN sebelumnya sebagai riwayat yang disimpan dan bisa digunakan ketika pemain ingin mengulang langkahnya.

File papan.txt dibuat dengan *default* FEN posisi mula-mula permainan catur ketika permainan baru dimulai.

```

Papan150
rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1
Papan150
r1bqkbnr/pppppppp/2n5/8/4P3/8/PPPP1PPP/RNBQKBNR w KQkq - 1 2
Papan150
r1bqkbnr/pppppppp/2n5/8/4P3/5N2/PPPP1PPP/RNBQKB1R b KQkq - 2 2
Papan150
r1bqkbnr/pppp1ppp/2n5/4p3/4P3/5N2/PPPP1PPP/RNBQKB1R w KQkq e6 0 3
Papan150
r1bqkbnr/pppp1ppp/2n5/4p3/2B1P3/5N2/PPPP1PPP/RNBQK2R b KQkq - 1 3
Papan150
r1bqkb1r/pppp1ppp/2n2n2/4p3/2B1P3/5N2/PPPP1PPP/RNBQK2R w KQkq - 2 4
Papan150
r1bqkb1r/pppp1ppp/2n2n2/4p1N1/2B1P3/8/PPPP1PPP/RNBQK2R b KQkq - 3 4
Papan150
r1bqk2r/pppp1ppp/2n2n2/2b1p1N1/2B1P3/8/PPPP1PPP/RNBQK2R w KQkq - 4 5
Papan150
r1bqk2r/pppp1Npp/2n2n2/2b1p3/2B1P3/8/PPPP1PPP/RNBQK2R b KQkq - 0 5
Papan150
r1bqk2r/pppp1Npp/2n2n2/4p3/2B1P3/8/PPPP1bPP/RNBQK2R w KQkq - 0 6
Papan150
r1bqk2r/pppp1Npp/2n2n2/4p3/2B1P3/8/PPPP1KPP/RNBQ3R b kq - 0 6
Papan150
r1bqk2r/pppp1Npp/2n5/4p3/2B1n3/8/PPPP1KPP/RNBQ3R w kq - 0 7
Papan150
r1bqk2r/pppp1Npp/2n5/4p3/2B1n3/8/PPPP2PP/RNBQ2KR b kq - 1 7
Papan150
r1b1k2r/pppp1Npp/2n5/4p3/2B1n2q/8/PPPP2PP/RNBQ2KR w kq - 2 8
Papan150
r1b1k2r/pppp1Npp/2n5/4p3/2B1n2q/6P1/PPPP3P/RNBQ2KR b kq - 0 8
Papan150
r1b1k2r/pppp1Npp/2n5/4p3/2B4q/6n1/PPPP3P/RNBQ2KR w kq - 0 9
Papan150
r1b1k2r/pppp1Npp/2n5/4p3/2B4q/6P1/PPPP4/RNBQ2KR b kq - 0 9
Papan150
r1b1k2r/pppp1Npp/2n5/4p3/2B5/6q1/PPPP4/RNBQ2KR w kq - 0 10
Papan150
r1b1k2r/pppp1Npp/2n5/4p3/2B5/6q1/PPPP4/RNBQ1K1R b kq - 1 10
Papan150
r1b2rk1/pppp1Npp/2n5/4p3/2B5/6q1/PPPP4/RNBQ1K1R w - - 2 11
Papan150
r1b2rk1/pppp1Npp/2n5/4p3/2B5/5Qq1/PPPP4/RNB2K1R b - - 3 11

```

Gambar 3. FEN pada File txt

FEN pada file papan.txt disimpan dua baris setiap giliran. Baris pertama berisi kode papan, baris kedua berisi FEN, jadi setiap kode papan mewakili FEN yang ada di bawahnya. FEN baru ditambah pada baris terakhir, jadi semakin ke bawah FEN semakin baru. Proses pencarian dilakukan terbalik dari bawah untuk memperoleh FEN terbaru dan mempercepat proses pencarian. Contoh pada Gambar 3 adalah hasil yang memuat satu kode papan yaitu Papan150. Ketika terdapat lebih dari satu kode papan di saat yang bersamaan, kode ditulis berbaur, bergantian tiap langkah.

4.4 Hasil Uji Coba Deteksi Suara

Deteksi perintah suara yang dilakukan menggunakan Web Speech API difokuskan pada dua pilihan bahasa yaitu Bahasa Indonesia dan Inggris, kata-kata yang diperlukan untuk dilakukan pendeteksian terbatas pada kata-kata yang digunakan dalam permainan catur.

Web Speech API mampu dengan baik mendeteksi nama-nama bidak yang diperlukan dalam permainan catur, namun kadang diperlukan beberapa kali percobaan sampai sistem bisa mendeteksi kata seperti pion karena Web Speech API dapat membaca pion sebagai beberapa kata lain misal "ion" atau "beyond". Kesalahan deteksi kata dapat diatasi dengan memetakan secara manual kata-kata lain yang terdeteksi saat diberikan perintah pion misal "ion" atau "dion" menjadi "pion" pada fungsi deteksi suara.

Percobaan dilakukan dengan menyebutkan kata atau perintah yang sama sebanyak 10 kali untuk setiap kata atau perintah yang mungkin digunakan pada permainan, hasilnya dapat dilihat pada Tabel 1.

Tabel 1 Akurasi Deteksi pada 10 Kali Percobaan

No	Kata Diperlukan	Akurasi	Kata Lain Terdeteksi
1	Pion	80%	ion, iya
2	Benteng	90%	ben 10
3	Kuda	90%	budha
4	Gajah	100%	-
5	Menteri	100%	-
6	Raja	90%	radja
7	Pawn	60%	phone, fall, farm, found
8	Rook	40%	book, (kata tidak terdeteksi)
9	Knight	80%	night, my
10	Bishop	80%	biso, pizza
11	Queen	80%	wait, weed
12	King	100%	-
13	Satu	100%	-
14	Dua	100%	-
15	Tiga	100%	-
16	Empat	70%	tempat, sempat, e empat
17	Lima	100%	-
18	Enam	100%	-
19	Tujuh	100%	-
20	Delapan	100%	-
21	One	90%	want
22	Two	100%	-
23	Three	100%	-
24	Four	50%	full, pool, tidak terdeteksi
25	Five	100%	-
26	Six	90%	sick
27	seven	100%	-
28	eight	50%	ate, tidak terdeteksi

Beberapa kata sama sekali tidak terdeteksi saat dilakukan pengecekan perintah suara, biasanya terjadi ketika perintah yang diberikan terlalu sering sehingga proses yang dijalankan menjadi semakin banyak dan koneksi internet tidak cukup cepat untuk menyelesaikan semua antrian yang ada.

Perintah selain kata yang juga diperlukan dalam permainan yaitu huruf untuk menandai petak mulai dari a sampai h lebih sulit dideteksi dibanding kata. Beberapa huruf memiliki tingkat akurasi di bawah 50%, huruf a sering diabaikan dan hanya dianggap sebagai noise sehingga sulit untuk dideteksi, begitu juga huruf e dan f. Huruf d sering salah dideteksi sebagai huruf b.

Perintah dapat diberikan langsung beberapa kata maupun satu per satu setiap kata. Perintah yang diberikan secara langsung beberapa kata memiliki risiko akurasi deteksi yang lebih rendah, perintah "kuda ke c 3" dapat dideteksi sebagai "kuda kecil 3" atau petak "e 3" dapat dideteksisi ebagai "ertiga".

Kata-kata yang terdeteksi sebagai kata yang salah dapat diatasi dengan memetakan kemungkinan kesalahan yang bisa terjadi dan menjadikannya sesuai dengankata yang diinginkan, masalah deteksi sulit untuk ditangani ketika API sama sekali tidak mendeteksi kata seperti yang terjadi ketika pengguna memberikan perintah "a" atau "rook". Kata salah juga tidak bisa dipetakan ketika kesalahan yang diterima juga terdapat pada kemungkinan perintah yang bisa digunakan dalam permainan, misal ketika "b" dideteksi sebagai "d".

5. Kesimpulan

Suara yang diterima game catur diubah menjadi teks menggunakan Web Speech API yang kemudian hasilnya diproses menjadi kata kunci untuk dicocokkan dengan kemungkinan langkah yang dihasilkan oleh Chess.js dan diproses sesuai notasi catur. Kemungkinan langkah digenerasi menggunakan Chess.js sesuai dengan kemungkinan langkah pada permainan catur asli kemudian hasil yang berupa notasi diubah menjadi kata-kata yang bisa dicocokkan dengan kata kunci baik dalam Bahasa Indonesia atau Inggris. Permainan *single player* menggunakan AI

dari *engine* yang memang sudah tersedia yaitu Stockfish dengan menjalankannya di belakang halaman menggunakan *worker*. Permainan *multi player* dilakukan secara *online* dengan menyimpan posisi pemain putih dan hitam sesuai kode papan yang ditentukan oleh kedua pemain pada file txt.

Daftar Pustaka

1. Norizan Mat Diah, Nathan Nossal, Nor Azan Mat Zin, Tadaki Higuchi HI. A Game Informatical Comparison of Chess and Association Football ("Soccer"). *Adv Comput Sci an Int J*. 2014;3(4):89–94.
2. Prost F. On the Impact of Information Technologies on Society: an Historical Perspective through the Game of Chess. Turing-100 Alan Turing Centen [Internet]. 2012;10(1):1–12. Available from: <http://arxiv.org/abs/1203.3434>
3. Steinerberger S. On the number of positions in chess without promotion. *Int J Game Theory*. 2015;44(3):761–7.
4. George Kovács, Alexander Petunin, Jevgenij Ivanko NYU. From the First Chess-Automaton to the Mars Pathfinder. *Acta Polytech Hungarica*. 2016;13(1):61–81.
5. Xiangran Du, Min Zahang XW. Self-Optimizing Evaluation Function for Chinese-Chess. *Int J Hybrid Inf Technol* [Internet]. 2014;7(4):163–72. Available from: http://www.sersc.org/journals/IJHIT/vol7_no4_2014/14.pdf
6. Vasiljevic GAM, De Miranda LC, De Miranda EEC. A Case Study of MasterMind Chess: Comparing Mouse/Keyboard Interaction with Kinect-Based Gestural Interface. *Adv Human-Computer Interact*. 2016;2016.
7. Wandana Narayana Putra, I AA Ketut Agung Cahyawan NP. Design and Built Interactive Whiteboard Application as Learning Support Using Game Controlling. *Int J Comput Sci Issues*. 2013;10(2):480–5.
8. Bhaskara SGA. Permainan Edukasi Labirin Virtual Reality Dengan Metode Collision Detection Dan Stereoscopic. *Lontar Komput* [Internet]. 2017;Volume 8(Nomor 2):Halaman 65-76. Available from: <https://ojs.unud.ac.id/index.php/lontar/article/view/31106/19848>
9. Këpuska V. Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx). *Int J Eng Res Appl* [Internet]. 2017;07(03):20–4. Available from: http://www.ijera.com/papers/Vol7_issue3/Part-2/D0703022024.pdf
10. Akanksha Kumari, Shreya Singh, Shailja Dalmia G V. Parallelization of Alpha-beta Pruning Algorithm for Enhancing The Two Player Games. *Int J Adv Electron Comput Sci*. 2017;4(2):74–81.
11. Agung Jodi Pratama, A.A. KOMPIANG Oka Sudana INP. Rancang Bangun Game Kartu Spirit Berbasis Android dengan Fitur Online Multiplayer. *Lontar Komput*. 2015;6(1):49–60.
12. Ni L.P. Pravina Utpatadevi, A. A. K. Oka Sudana AAKAC. Implementation of MVC (Model-View-Controller) Architectural to Academic Management Information System with Android Platform Base. *Int J Comput Appl*. 2012;57(8):1–6.