**LAPORAN PEMBUATAN PROGRAM**

**GAME FAST CODING**



**NAMA KELOMPOK : KELOMPOK 10**

**Rhizma Elliza Putri** **20082010120**

**Achmad Yusuf Al Ma’ruf**  **20082010148**

**PROGRAM STUDI S1 SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UPN “VETERAN” JAWA TIMUR**

**SURABAYA**

**2022**

**DAFTAR ISI**

[**BAB I** 3](#_Toc100530575)

[**PENDAHULUAN** 3](#_Toc100530576)

[**1.1 Latar Belakang Masalah** 3](#_Toc100530577)

[**1.2 Identifikasi Masalah** 4](#_Toc100530578)

[**1.3 Pembatasan Masalah** 4](#_Toc100530579)

[**1.4 Rumusan Masalah** 4](#_Toc100530580)

[**1.5 Manfaat Pengembangan** 4](#_Toc100530581)

[**BAB II** 6](#_Toc100530582)

[**ANALISA DAN PERANCANGAN PROGRAM** 6](#_Toc100530583)

[**2.1 Diskripsi Program** 6](#_Toc100530584)

[**2.2 Rancang Bangun Program** 6](#_Toc100530585)

[**2.3 Alur Kerja Program** 16](#_Toc100530586)

[**BAB III** 17](#_Toc100530587)

[**PENGUJIAN , SCRIPT, DAN TAMPILAN PROGRAM** 17](#_Toc100530588)

[**3.1 Tampilan Program Dan Pengujian** 17](#_Toc100530589)

[**3.2 SOURCE CODE** 19](#_Toc100530590)

[**BAB IV** 35](#_Toc100530591)

[**PENUTUP** 35](#_Toc100530592)

[**4.1 KESIMPULAN** 35](#_Toc100530593)

[**4.2 SARAN** 35](#_Toc100530594)

# **BAB I**

# **PENDAHULUAN**

## **1.1 Latar Belakang Masalah**

Pesatnya perkembangan Ilmu Pengetahuan dan Teknologi sudah tidak diragukan lagi. Generasi saat ini merupakan generasi milenial yang memiliki pengaruh besar dalam membuat dunia menuju ke era yang lebih maju. Pesatnya perkembangan IPTEK juga dipengaruhi oleh adanya internet yang membuat segala informasi menyebar dengan cepat. Banyak informasi disediakan di berbagai jenis platform seperti aplikasi maupun website. Dalam membuat keduanya, diperlukan kemampuan yang mumpuni dalam meng-*coding*.

Dalam membuat suatu web atau laman diperlukan suatu bahasa yang berisi perintah yang dipahami oleh komputer sehingga harus menggunakan bahasa yang tepat, salah satunya yaitu **HTML.** **HTML** merupakan singkatan dari **Hypertext Markup Language** yang merupakan bahasa format dari tag XML atau Extended Markup Language dan dijadikan sebagai standar umum dari suatu halaman web. HTML memiliki standar elemen yang terdiri atas 3 bagian, yakni : 1) **DTD atau Document Type Declaration**, yang digunakan sebagai representasi dari tiap elemen pada docWeb, 2) **Head**, yang digunakan untuk memberi informasi tentang dokumen, 3) **Body**, yang berisi konten dari web. Selain HTML ada Javascript dan CSS (Cascading Style Sheet). Javascript tidak berbeda jauh dengan HTML yang kita kenal, bahasa pemrograman ini sering digunakan untuk membuat tampilan website menjadi lebih menarik dan biasanya digunakan untuk membuat aplikai mobile, desktop, dll. Sedangkan CSS berguna untuk memformat dokumen yang ditulis menggunakan HTML dengan tujuan memisahkan konten dari tampilan visualnya di situs laman web.

Dalam proses mengkoding sendiri seorang programmer membutuhakan sebuah ketrampilan. Salah satu ketrampilan tersebut adalah mengetik. Programmer harus sebisa mungkin memiliki ketrampilan mengetik dengan akurasi yang baik, WPM (Word per Minute) yang tinggi, CPM (Character per Minute) yang tinggi serta tingkat kesalahan / *typo* yang sedikit

Oleh karena itu pembuatan program *fast typing* bernama Fast Coding dengan struktur HTML, Javascript, dan CSS yang edukatif ini bertujuan untuk membantu programmer melatih kemampuan mengetiknya. Dengan harapan semakin sering seorang programmer melatih kemampuanya dengan menggunkan program ini maka kemampuan mengetiknya semakin terasah.

## **1.2 Identifikasi Masalah**

Berdasarkan latar belakang masalah yang telah dikemukakan, maka dapat diidentifikasi masalah-masalah, yaitu :

1. Kemampuan mengetik kode oleh programmer yang masih kurang maksimal
2. Tidak adanya platform khusus untuk berlatih kemampuan mengetik khususnya dalam mengetik kode program

## **1.3 Pembatasan Masalah**

Terdapat banyak permasalahan yang dapat ditemui dalam pengembangan pembuatan program game soal. Oleh karena itu, perlu dilakukan batasan terhadap masalah yang menjadi ruang lingkup dalam penelitian ini yaitu :

1. Penelitian ini difokuskan pada pembuatan program game fast typing beserta cara memainkanya.
2. Pilihan bahasa pemrograman yang tersedia terbatas yaitu HTML, Javascript, dan CSS saja.
3. Quotes pada game sudah di defeniskan pada file javascript.
4. Rata-rata character pada setiap kata dibulatkan menjadi 7 untuk perhitungan kata per menit

## **1.4 Rumusan Masalah**

Berdasarkan latar belakang masalah di atas, maka rumusan masalah pada penelitian ini adalah.

1. Bagaimana deskripsi dan manfaat dari game FastCoding?
2. Bagaimana cara pembuatan program game FastCoding?
3. Bagaimana tampilan program serta cara memainkan game FastCoding?

## **1.5 Manfaat Pengembangan**

Pembuatan program ini dibuat untuk mengembangkan kemampuan pemrograman bagi penulis sekaligus untuk memberikan manfaat bagi pengguna yang memainkannya. Manfaat dari program ini diantaranya:

1. Sebagai bahan dasar atau media bagi seseorang yang ingin belajar mengetik secara cepat dan tepat
2. Sebagai bahan dasar atau media bagi seseorang yang ingin belajar bahasa pemrograman
3. Sebagai bahan dasar atau media bagi seseorang untuk melatih tingkat kefokusan mereka
4. Sebagai bahan dasar atau media bagi seseorang untuk melatih tingkat kecekatan/spontanitas mereka
5. Sebagai bahan dasar atau media bagi seseorang untuk menghilangkan rasa jenuh

# **BAB II**

# **ANALISA DAN PERANCANGAN PROGRAM**

## **2.1 Diskripsi Program**

1. Program ini dibuat untuk melatih mahasiswa atau pengguna agar dapat mengetik code dengan menggunakan 10 jari dengan cepat dan tepat dalam waktu satu menit
2. Program tersebut dibuat dengan menggunakan html, css dan javascript.
3. Quotes bahasa (tulisan yang harus di tulis) dalam bahasa HTML, CSS, dan Javascript dimana pengguna nantinya memilih salah satu dari ketiga bahasa tersebut.
4. Tampilan awal program menampilkan beberapa item, yakni:
5. Judul dan deskripsi program
6. Waktu
7. Pemilihan bahasa
8. Quotes (tempat dimana kalimat harus di tulis)
9. Textarea input (tempat dimana user menginputan kata sesuai dengan quotes)
10. Tampilan saat dimainkan menunjukkan detik berjalan mundur beserta teks yang harus kita tulis
11. Tampilan akhir menunjukkan hasil dari permainan yang kita lakukan seperti:
12. Ketepatan (Accuracy)
13. WPM (Word per Minute)
14. CPM (Character per Minute)
15. Kesalahan (Errors)
16. Tombol restart (untuk memulai kembali permainan)

## **2.2 Rancang Bangun Program**

1. **Persiapan alat dan bahan**

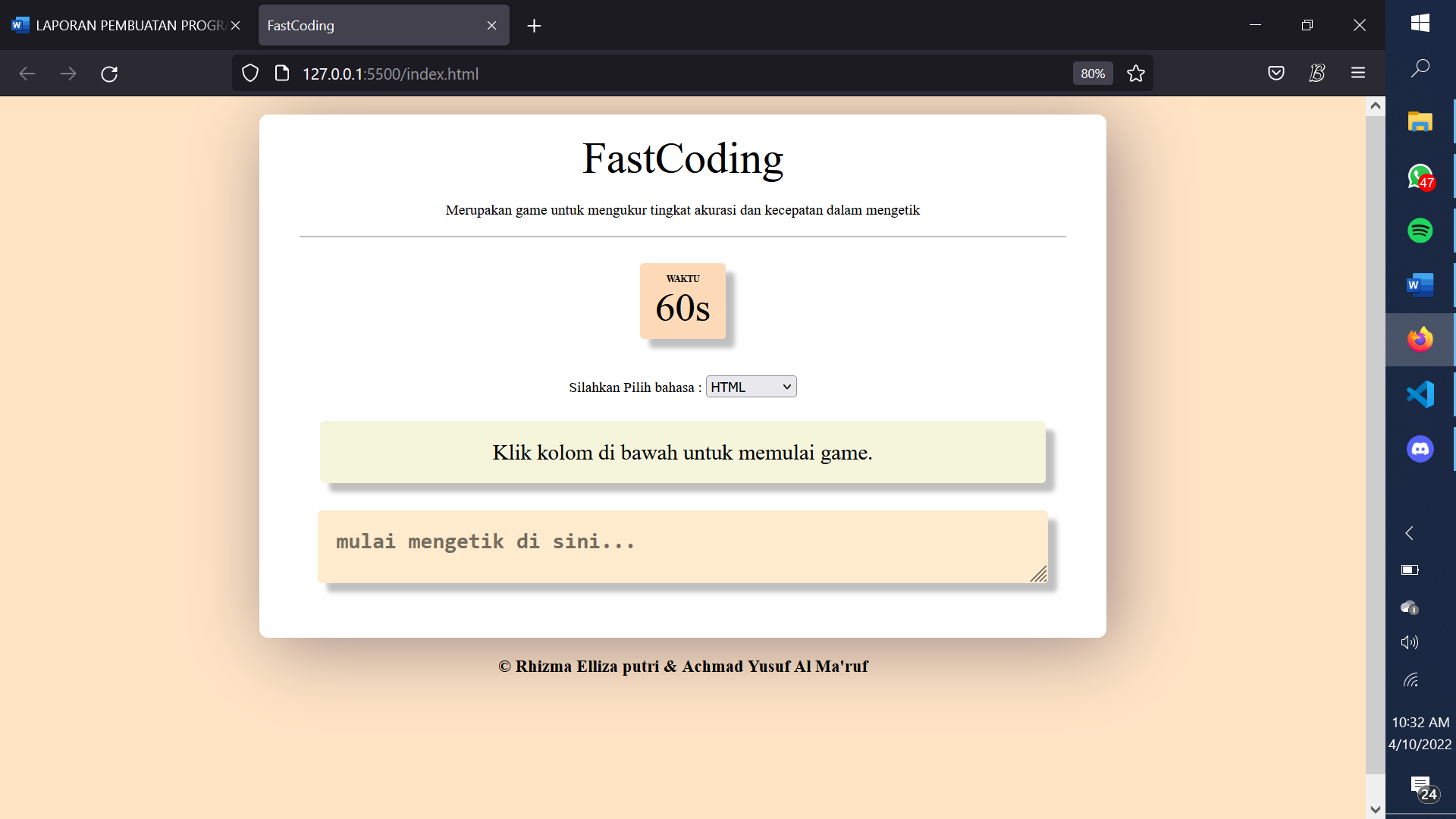
Pertama kita persiapan alat dan bahan diantaranya sebagai berikut :

* 1. Visual studio code
  2. Xampp
  3. Internet
  4. Laptop
  5. Keyboard
  6. Mouse

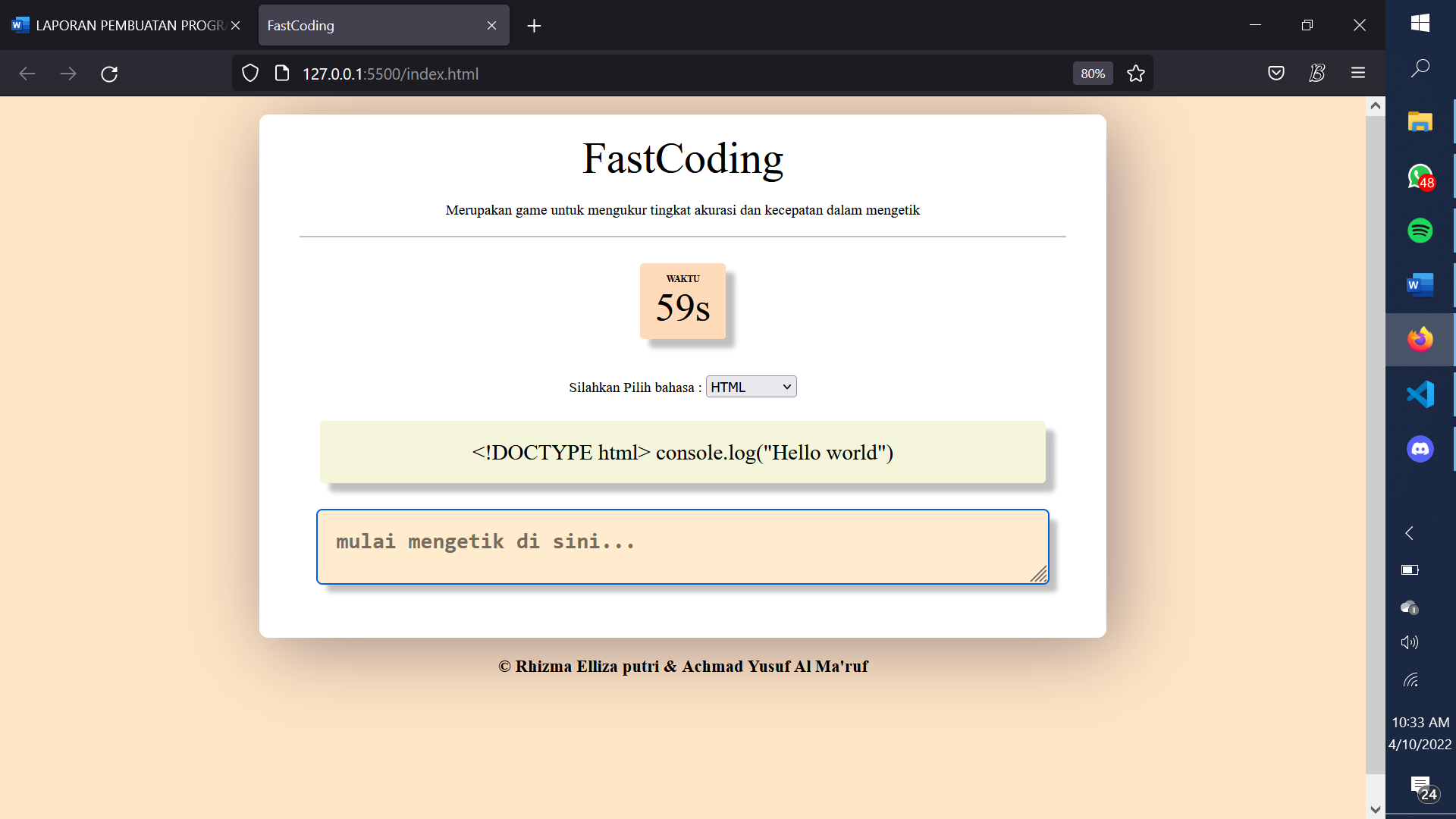
1. **Perancangan User Inteface**

Selanjutnya kita membuat gambaran awal dari program yang akan kita buat sebagai berikut :

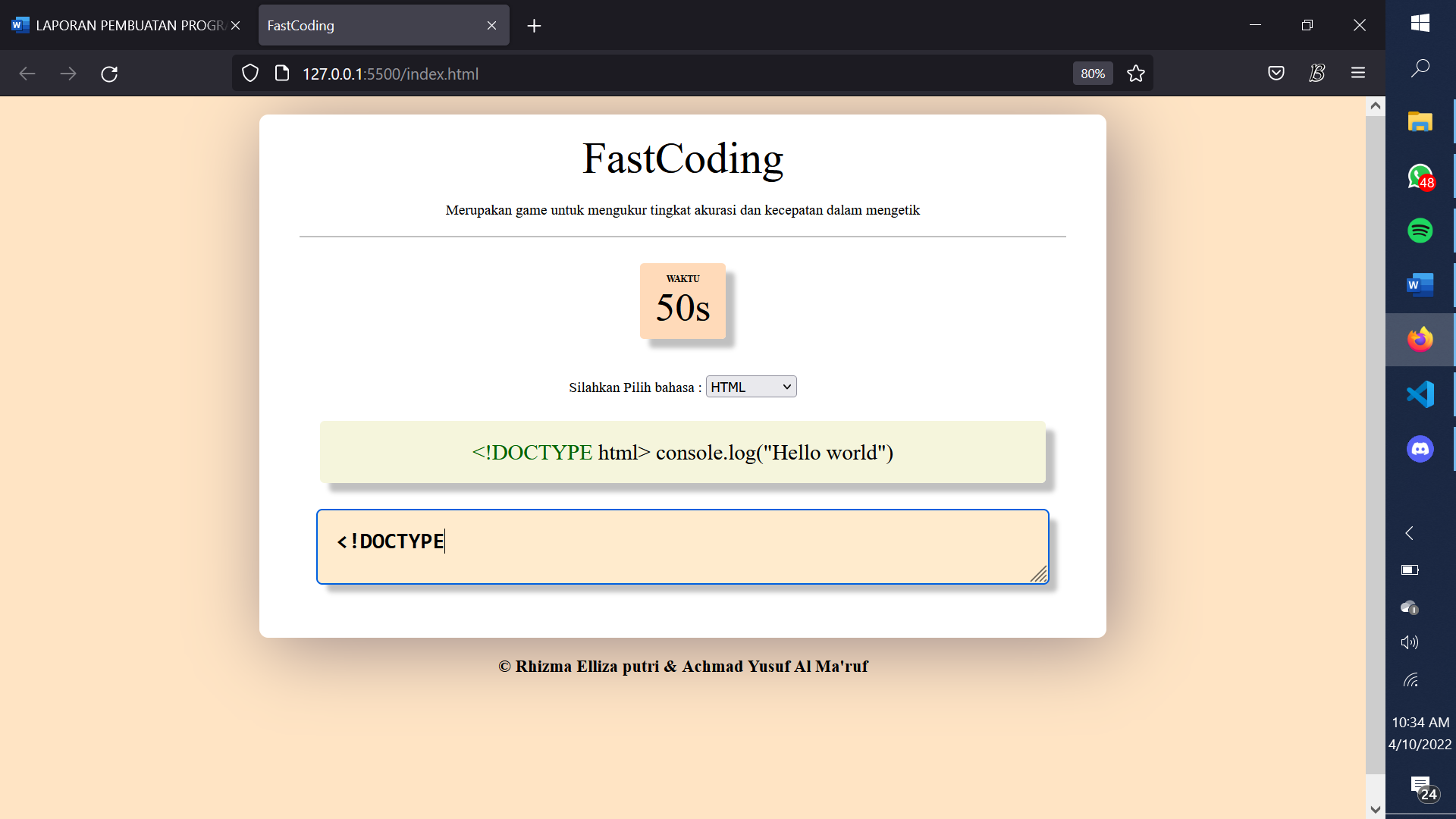
Tampilan awal program sebelum dimulai :



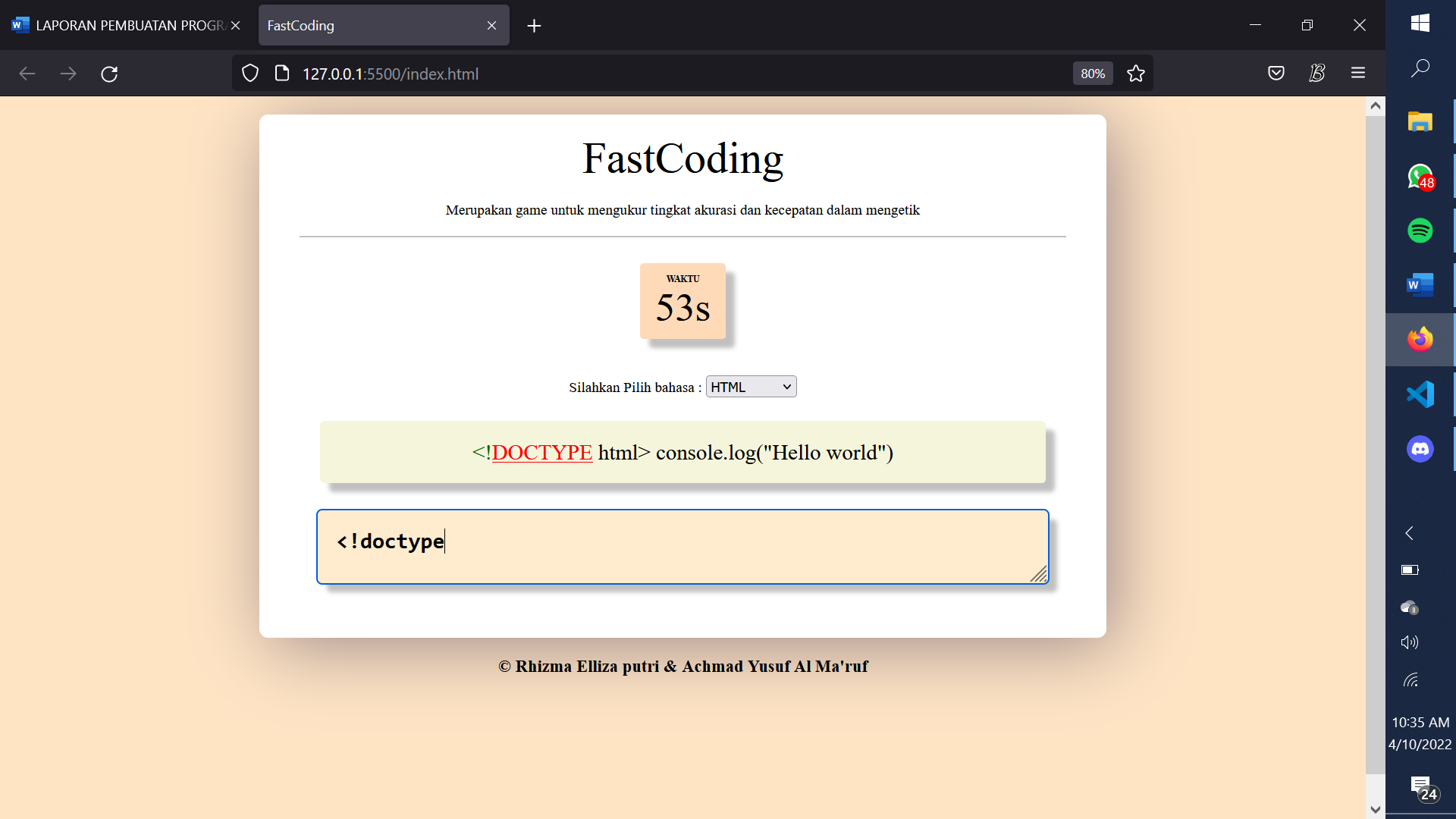
Tampilan awal ketika mulai mengetik :



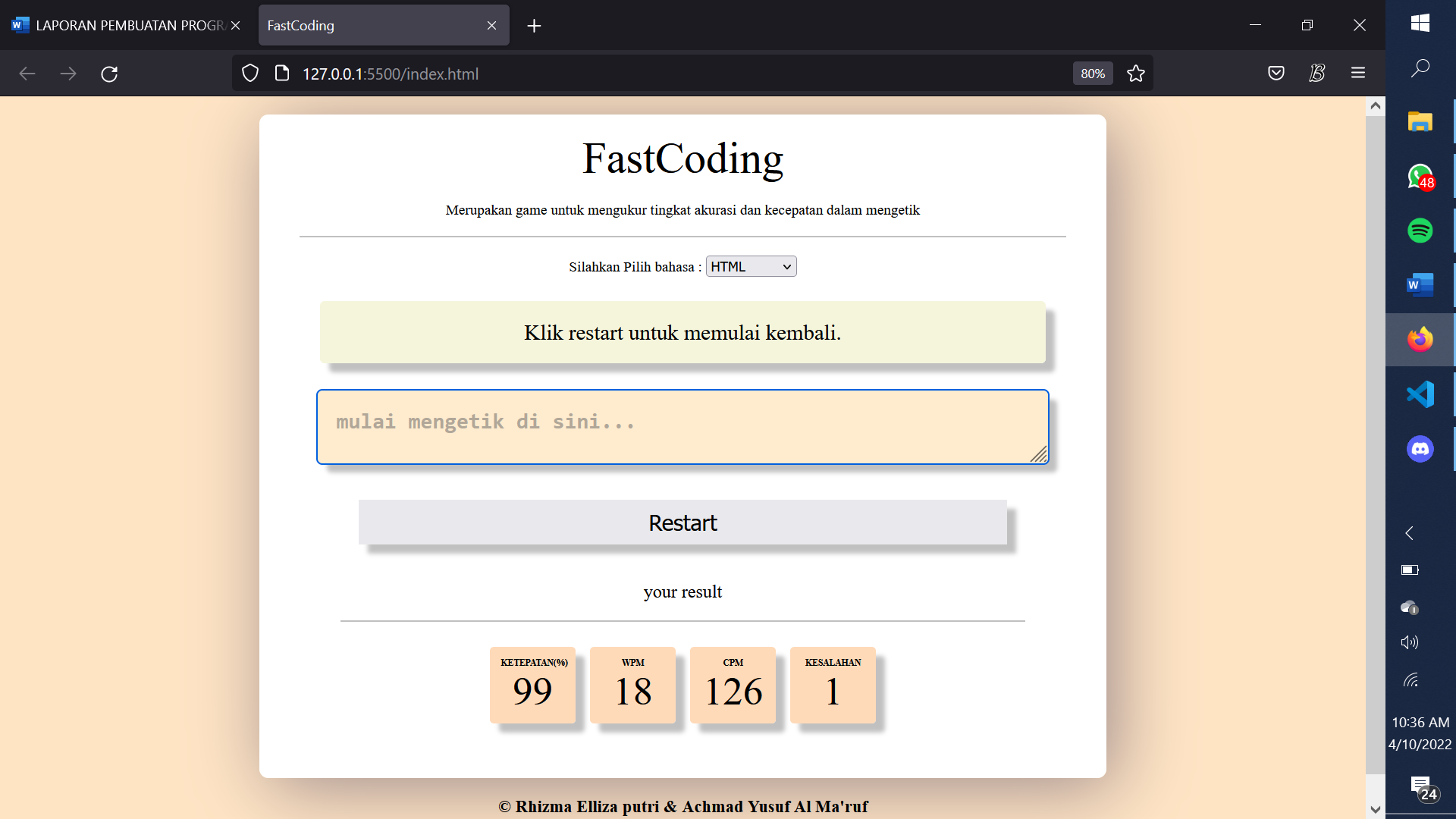
Tampilan ketika mengetik dengan benar :



Tampilan ketika mengetik dengan salah :



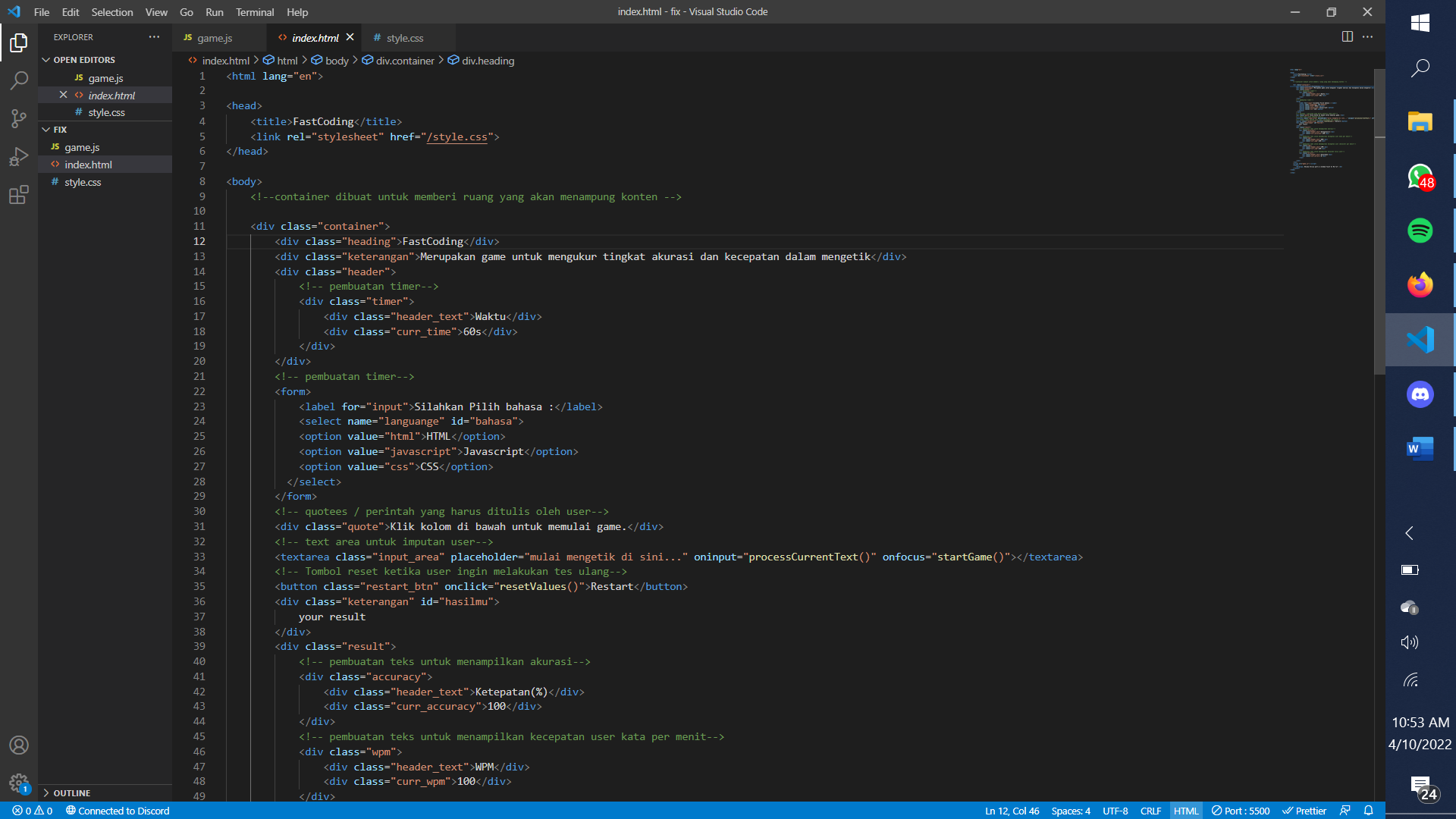
Tampilan akhir ketika program telah selesai :



1. **Pembuatan Program**
2. Pembuatan file HTML

Pembuatan file html digunakan untuk mendefinisikan struktur elemen yang akan ditampilkan pada halaman sebagai berikut :

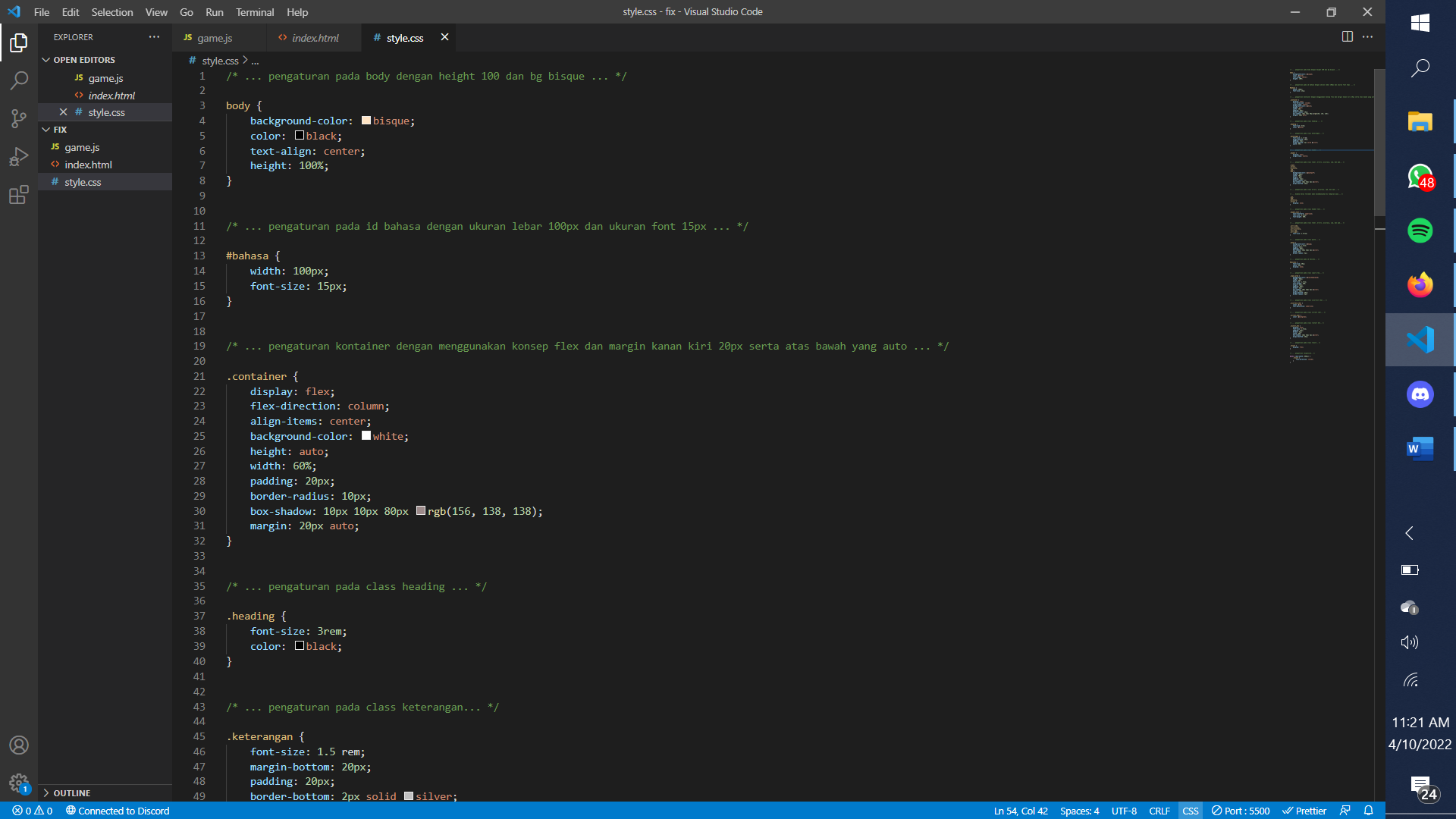
1. **Heading :** bagian iniberisi judul dan dikrispi program serta *timer* untuk menunjukkan waktu yang tersisa saat mengetik
2. **Form language:** bagian ini berisi *select option* untuk pengguna memilih bahasa apa yang akan dipilih
3. **Quotes :** bagian ini menunjukan teks saat ini yang harus diketik di area input
4. **Area input :** bagian ini berisi inputan dari *user*.
5. **Tombol restart :**  adalah tombol yang akan ditampilkan setelah waktu permainan habis
6. **Result :** bagian ini berisi WPM, CPM. ketepatan, dan kesalahan yang akan ditampilkan pada halaman ketika waktu telah selesai



1. Pembuatan file CSS

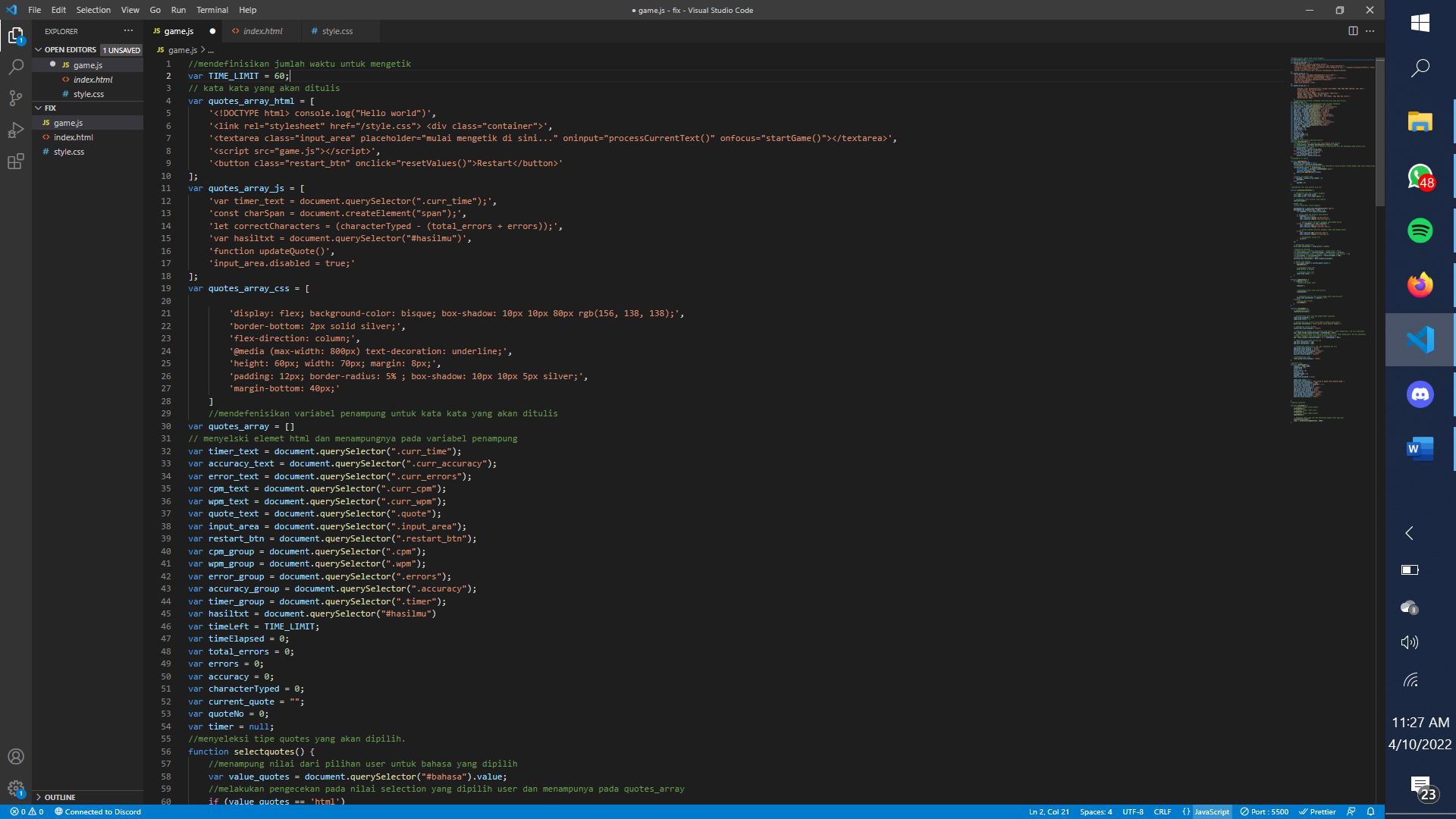
CSS digunakan untuk menata bagian yang berbeda dan membuatnya lebih menarik secara visual

* + - * 1. *Padding* dan *margin* diatur untuk memberikan jarak yang tepat untuk setiap element
        2. Ukuran teks setiap element diatur sedemikian rupa sehingga mudah dibaca oleh pengguna saat bermain game
        3. Dua kelas tambahan didefinisikan untuk menunjukan huruf yang diketik dengan benar atau salah. Kelas kelas ini akan ditambahkan atau dihapus secara dinamis bila diperlukan.
        4. Kelas result dibuat hide terlebih dahulu sehingga tidak ditampilkan ketika game pertama dimainkan selanjutnya ketika game telah selesai maka ditampilkan

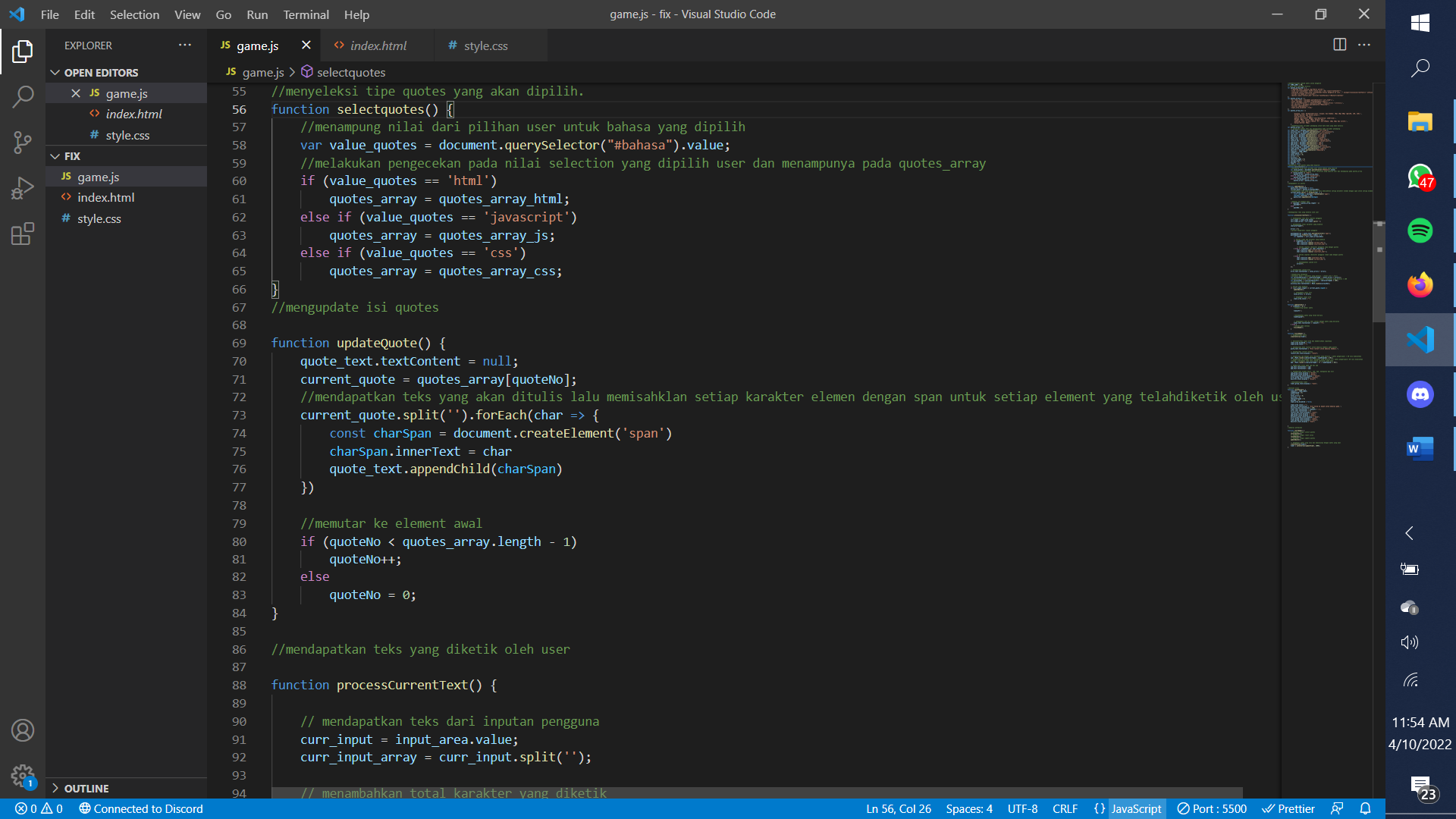


1. Pembuatan file JS

Memilih semua elemen dan mendefinisikan variable Elemen yang diperlukan dalam file HTML yang dipilih menggunakan metode querySelector. Selain itu variabel lain yang akan diakses sepanjang program juga ditentukan di awal.



Selanjutnya mendefiniskan isi quotes dari hasil pemilihan yang dilakukan oleh pengguna



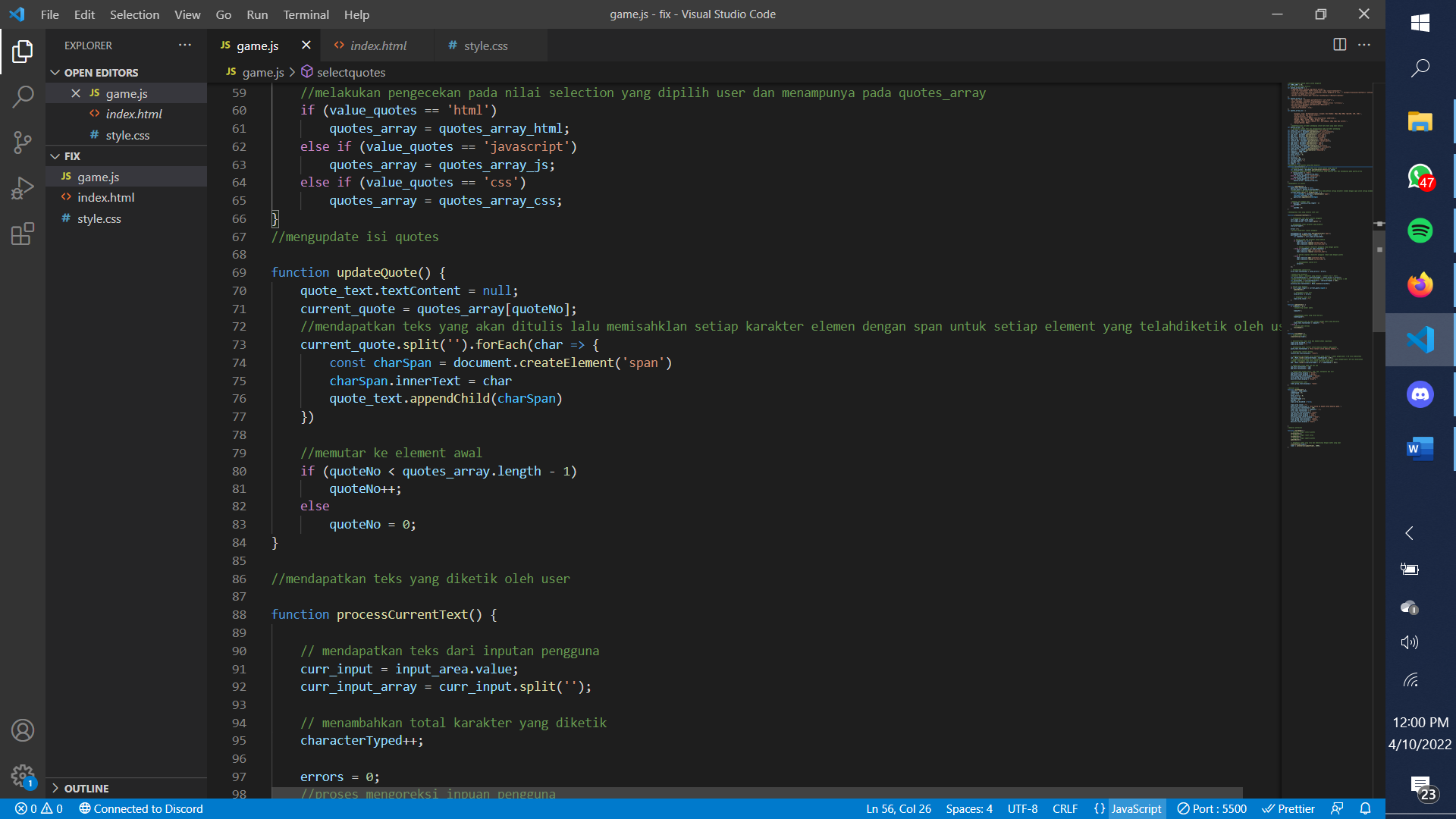
Menampilkan teks dengan fungsi quotes dengan cara sperti berikut:

1. **Mendapatkan teks**

Setiap kutipan diambil satu persatu dari array yang sudah ditentukan. Dimana terdapat sebuah variable untuk melacak indeks kutipan saat ini dan menambahkan setiap kali character baru diminta

1. **Memisahkan karakter**

Setiap karakter dalam teks dipisahkan menjadi sebuah elemen span. Hal ini bertujuan untuk mengubah warna setiap karakter tergantung apakah karakter yang telah diketik benar atau salah. Elemn tersebuh ditambahkan ke variable quote\_text



1. **Mendapatkan teks yang saat ini diketik pengguna**

Fungsi processCurrentText() didefinisikan yang akan dipanggil setiap kali pengguna mengetik atau mengubah apa pun di kotak input.

1. **Mendapatkan nilai saat ini dari kotak input**

Nilai dari area input digunakan untuk mendapatkan teks saat ini yang diketik oleh pengguna. Yang dibagi menjadi serangkaian karakter untuk dibandingkan dengan teks quotes. Dan disimpan di curr\_input\_array

1. **Mewarnai karakter teks kutipan**

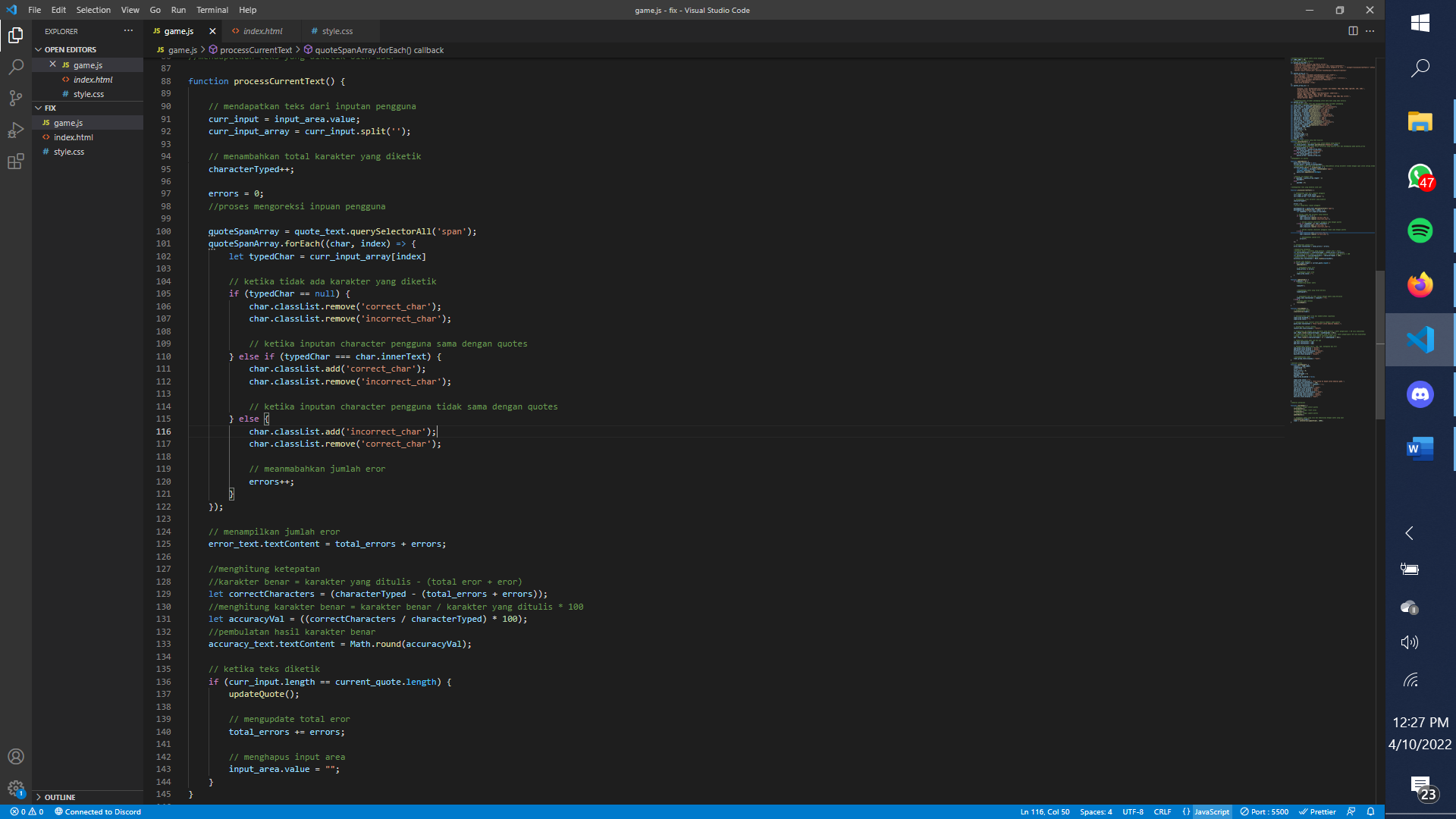
Karakter kutipan yang ditampilkan berwarna merah atau hijau tergantung apakah karakter diketik dengan benar. Hal tersebut dilakukan dengan memilih elemen span dari kutipan yang telah dibuat sebelumnya dan mengulanginya. Elemen tersebut kemudian menerapkan kelas yang dibuat tergantung pada apakah itu sama dengan teks yang diketik.

1. **Menghitung kesalahan dan akurasi**

Setiap kali pengguna membuat kesalahan saat mengetik, variabel eror bertambah. Ini digunakan untuk menghitung nilai akurasi dengan membagi jumlah karakter yang diketik dengan benar dengan jumlah total karakter yang diketik oleh pengguna.

1. **Pindah ke kutipan berikutnya**

Ketika panjang teks input cocok dengan panjang teks quotes, fungsi updateQuote() dipanggil yang mengubah kutipan dan mengosongkan area input. Jumlah total eror juga diperbarui untuk digunakan pada kutipan berikutnya.

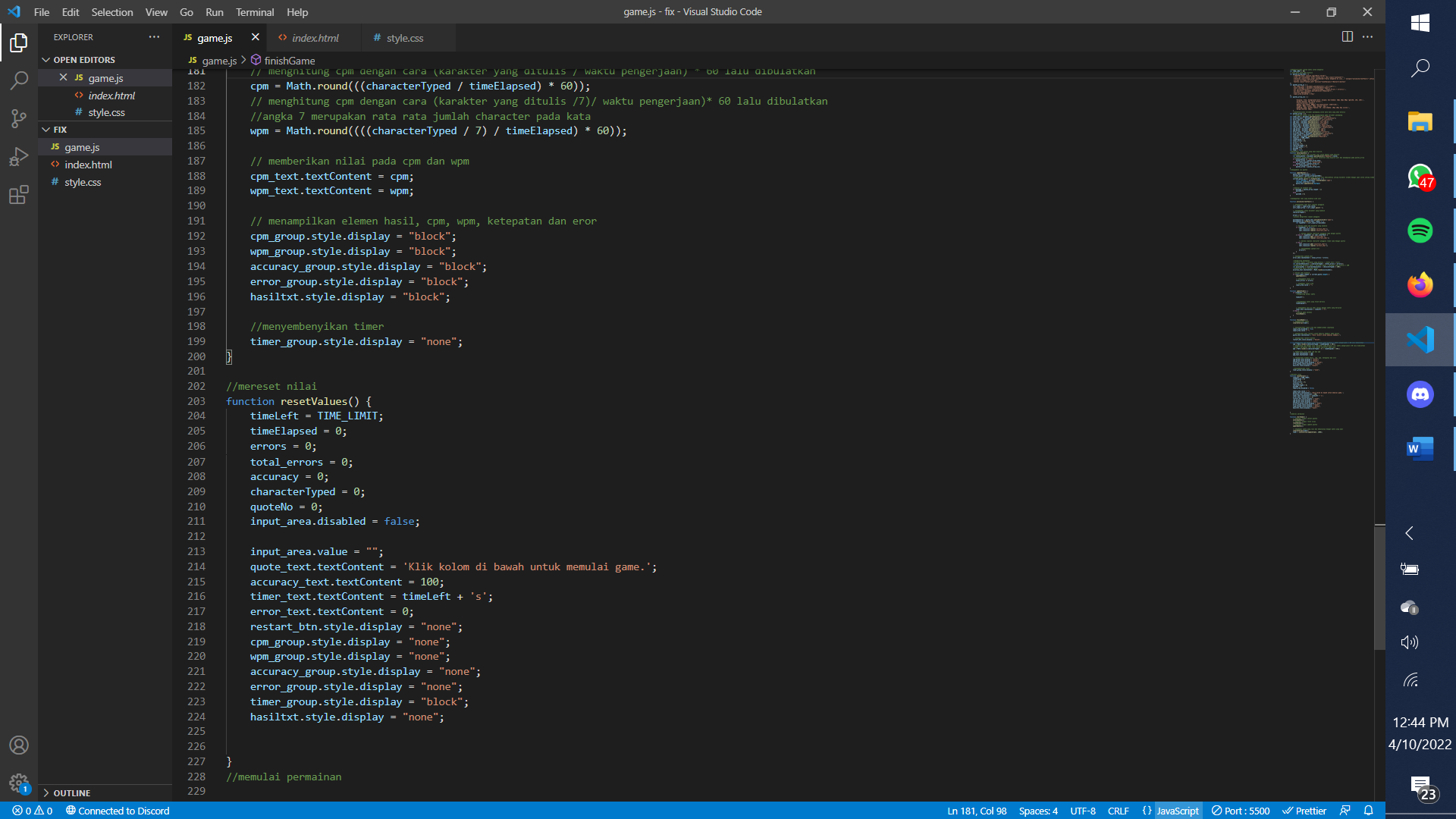


1. **Memulai permainan baru**

Fungsi startGame() didefinisikan yang akan dipanggil ketika pengguna mengklik pada kotak input.

1. **Setel ulang semua nilai**

Semua nilai disetel ulang ke nilai default sebelum memulai permainan baru menggunakan fungsi resetValues()

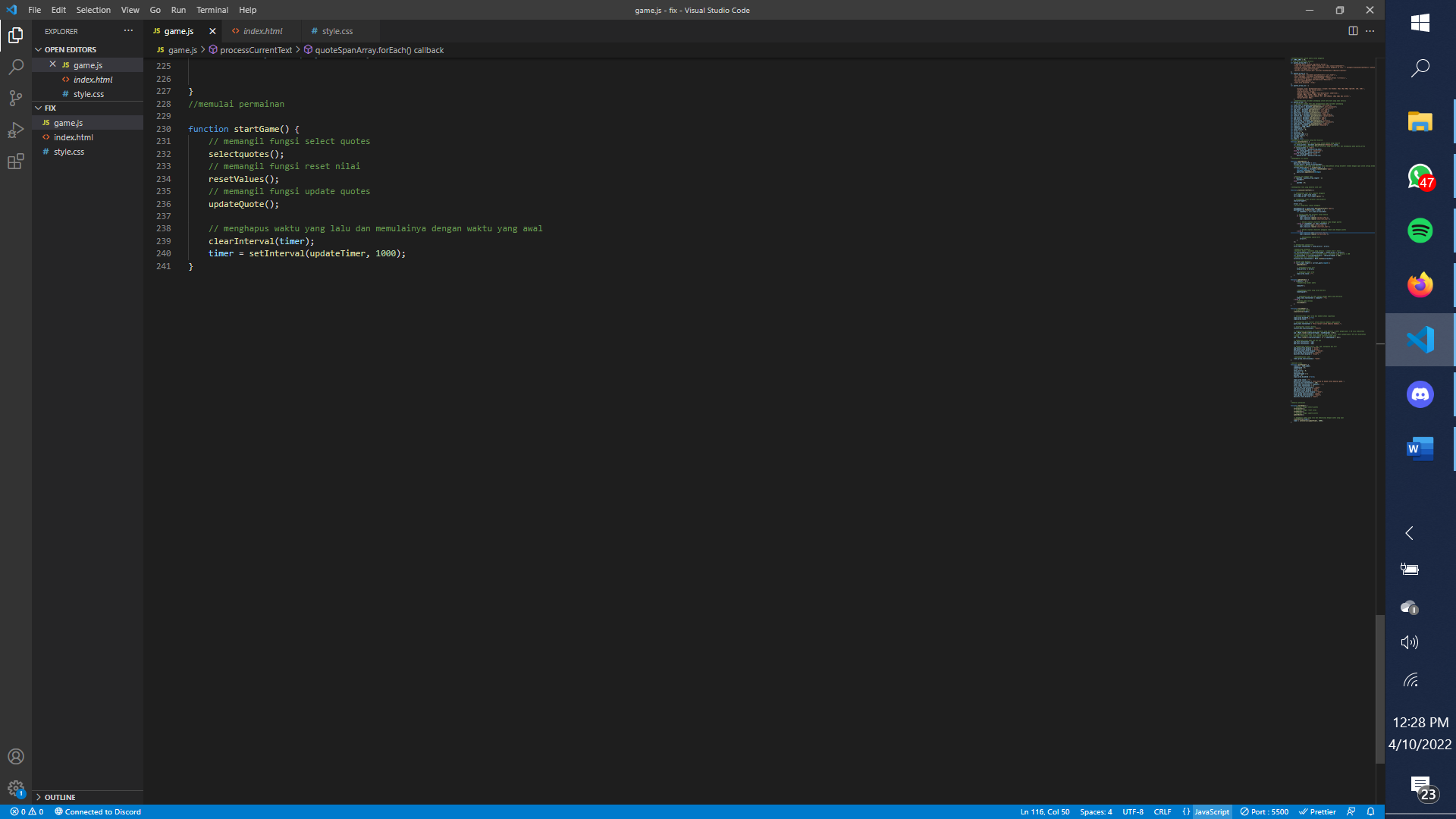


1. **Perbarui teks kutipan Teks**

Kutipan baru ditampilkan dengan memanggil fungsi updateQuote().

1. **Membuat timer baru**

**M**elacak jumlah detik yang tersisa dan menampilkannya kepada pengguna dibuat menggunakan metode setInterval() yang berulang kali memanggil fungsi updateTimer(). Sebelum membuat timer baru, instance timer sebelumnya dibersihkan menggunakan clearInterval().



1. **Memperbarui timer**

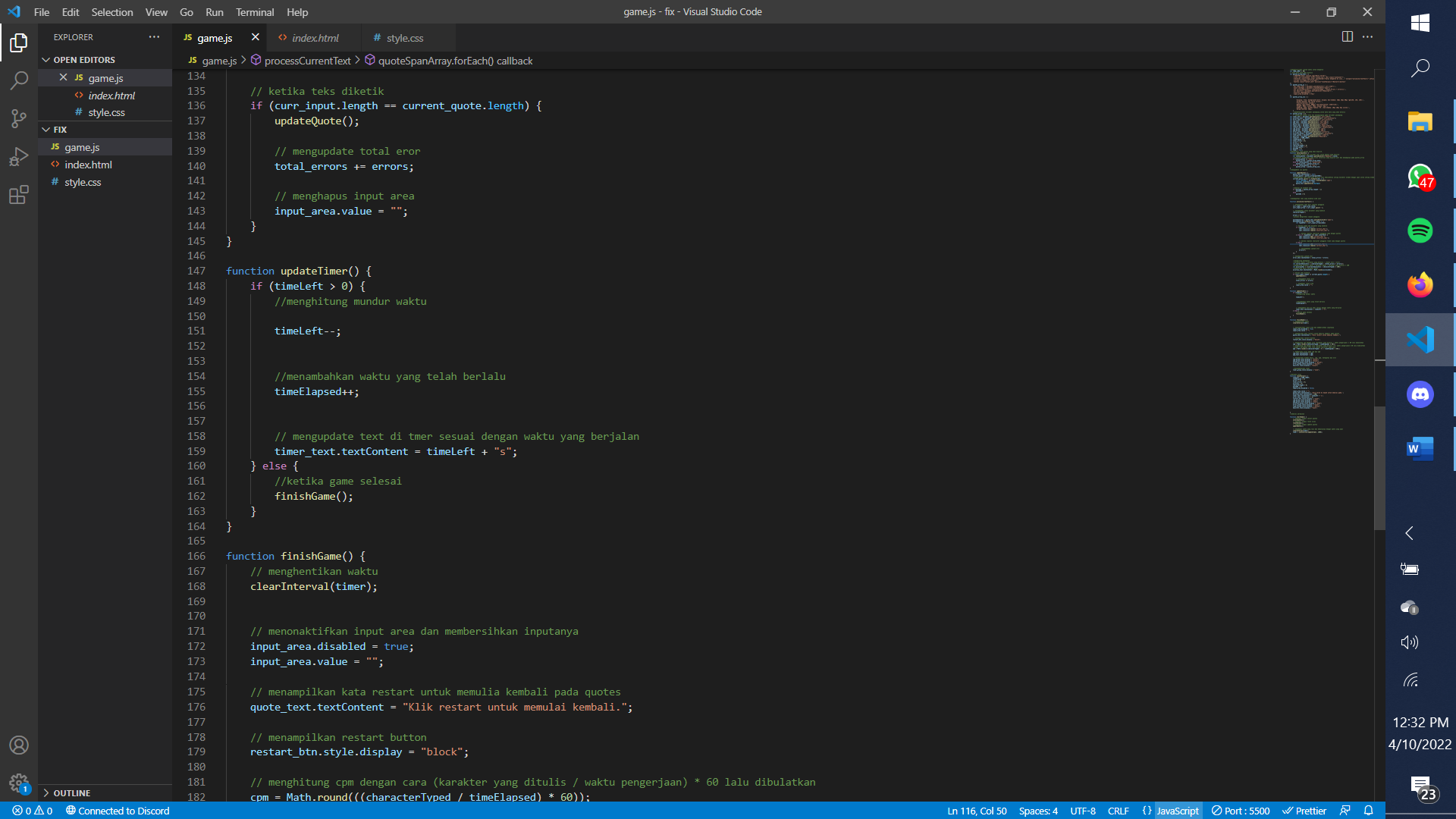
Fungsi updateTimer() didefinisikan yang akan dipanggil setiap detik untuk melacak waktu. Fungsi ini menangani hal-hal berikut:

1. **Perbarui nilai waktu**

Semua variabel yang melacak waktu diperbarui. Nilai timeLeft dikurangi, nilai timeElapsed bertambah, dan teks pengatur waktu diperbarui ke waktu tersisa.

1. **Menyelesaikan Permainan**

Memanggil fungsi finishGame() yang ditentukan di bawah yang menyelesaikan permainan.



1. **Menyelesaikan permainan**

Fungsi finishGame() didefinisikan yang akan dipanggil ketika permainan harus diselesaikan. Fungsi ini menangani hal-hal berikut:

1. **Menghapus**

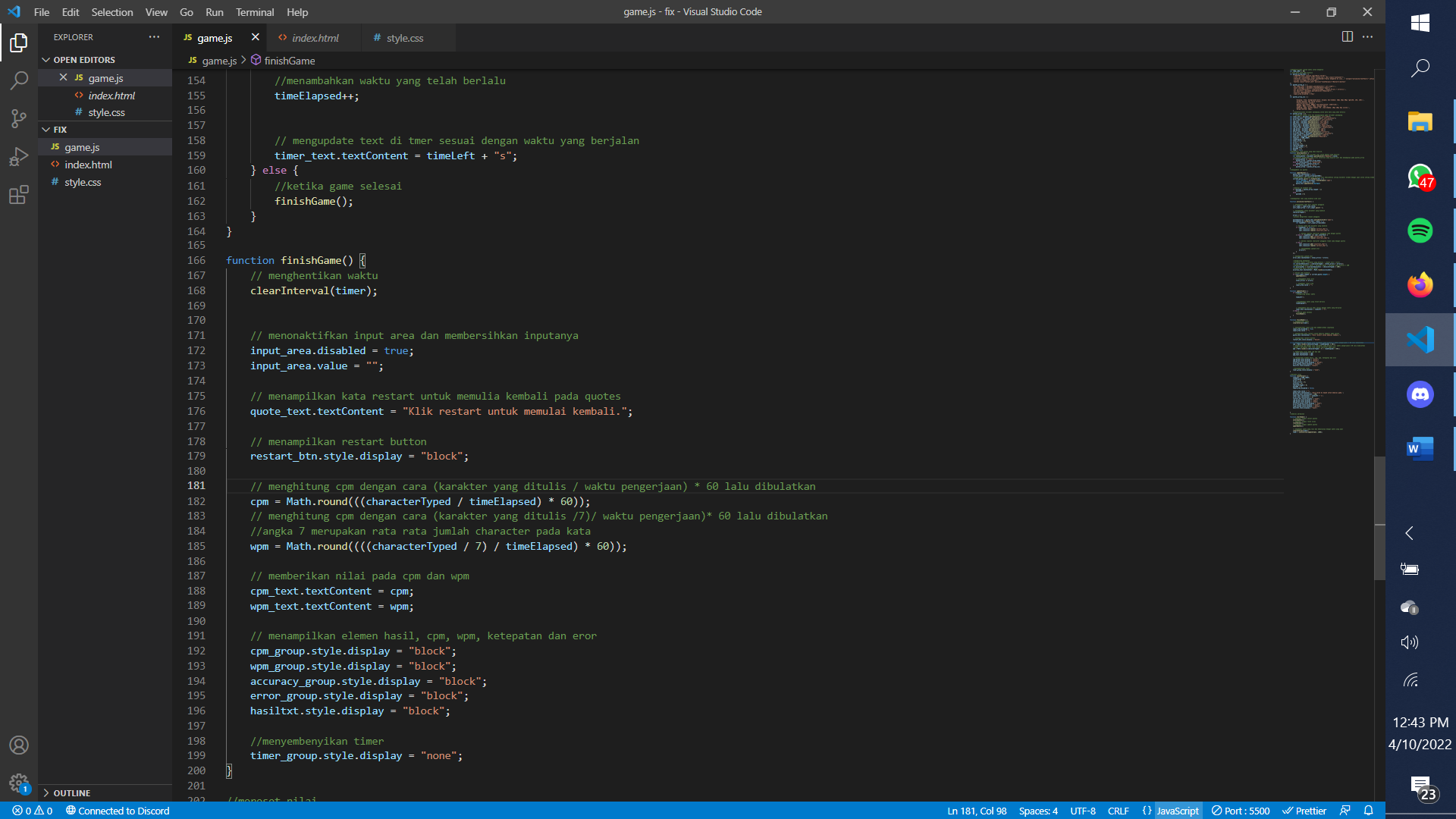
Timer Instance timer yang dibuat sebelumnya akan dihapus.

1. **Menampilkan teks dan tombol mulai ulang permainan Teks**

Kutipan yang ditampilkan kepada pengguna diubah menjadi teks yang menunjukkan bahwa permainan telah berakhir. Tombol 'Restart' juga ditampilkan dengan mengatur properti tampilan ke 'block'.

1. **Menghitung CPM dan WPM**:

Karakter Per Menit (CPM) dihitung dengan membagi jumlah karakter yang diketik dengan waktu yang berlalu dan kemudian mengalikan hasilnya dengan 60. Dibulatkan untuk mencegah titik desimal sedangkan Words Per Minute (WPM) dihitung dengan membagi CPM dengan 7 dan kemudian mengalikan hasilnya dengan 60. Angka7 menunjukkan jumlah rata-rata karakter per kata. Ini dibulatkan untuk mencegah titik desimal



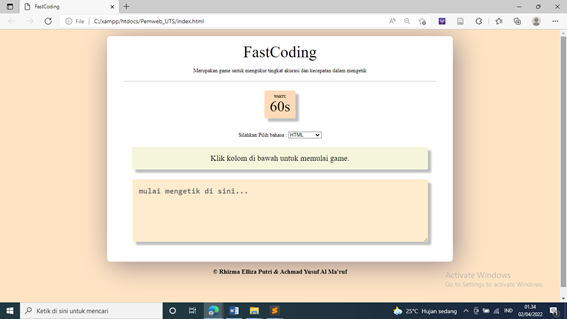
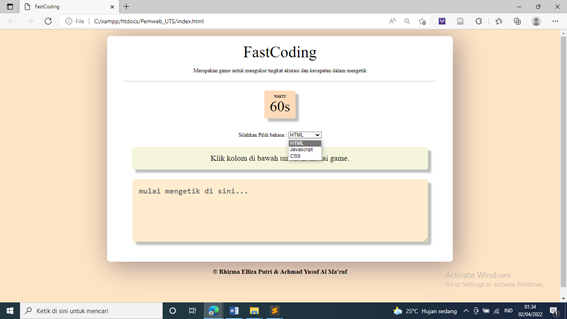
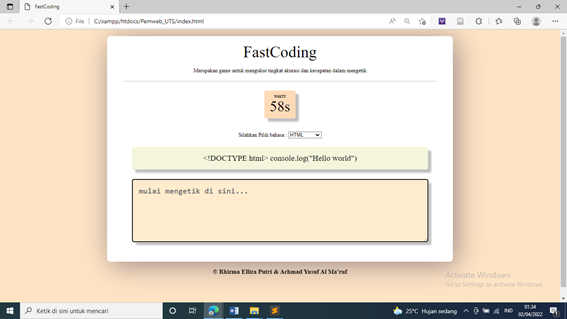
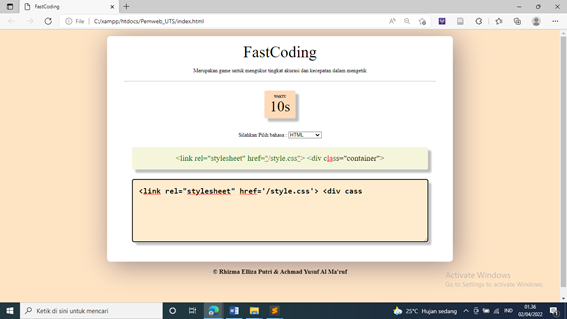
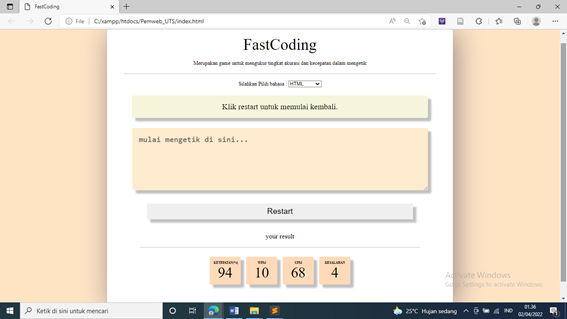
## **2.3 Alur Kerja Program**

1. Tampilan awal program terdapat menu yang mengharuskan kita untuk memilih bahasa pemrograman yang akan kita ketik. Menu tersebut berupa select option yang terdapat pilihan HTML, Javascript, dan CSS. Defaultnya yaitu HTML.
2. Ketika kita sudah memilih bahasa pemrograman maka kita akan beralih ke kolom yang bertuliskan “mulai mengetik di sini...” dengan cara mengkliknya.
3. Ketika kolom tersebut diklik maka waktu yang terdapat di bagian atas akan berjalan mundur secara otomatis. Pada kolom tersebut, kita harus mengetik kalimat yang sesuai dengan apa yang muncul tepat di bagian atas kolom tersebut.
4. Apabila ketikan yang kita ketik sesuai dengan apa yang diperintahkan maka tulisan perintah yang terdapat pada atas kolom akan berwarna hijau sedangkan apabila ketikan yang kita ketik tidak sesuai maka akan berwarna merah.
5. Kita diberi kesempatan untuk mengetik selama waktu masih tersedia dan ketika waktu habis maka akan muncul tombol restart jika ingin memulai ulang disertai hasil atau nilai dari yang kita mainkan seperti:
6. Ketepatan
7. WPM
8. CPM
9. Kesalahan

# **BAB III**

# **PENGUJIAN , SCRIPT, DAN TAMPILAN PROGRAM**

## **3.1 Tampilan Program Dan Pengujian**

1. Tampilan Awal Program
2. Tampilan Saat Memilih Bahasa
3. Tampilan Saat Game Dimulai (Input area diklik)
4. Tampilan Ketika Semua Huruf Diketik Secara Tepat
5. Tampilan Ketika Terdapat Huruf Yang Diketik Dengan Salah
6. Tampilan Saat Game Selesai Dimainkan

## **3.2 SOURCE CODE**

1. HTML (index.html)
2. <html lang="en">
3. <head>
4. <title>FastCoding</title>
5. <link rel="stylesheet" href="style.css">
6. </head>
7. <body>
8. <!--container dibuat untuk memberi ruang yang akan menampung konten -->
9. <div class="container">
10. <div class="heading">FastCoding</div>
11. <div class="keterangan">Merupakan game untuk mengukur tingkat akurasi dan kecepatan dalam mengetik</div>
12. <div class="header">
13. <!-- pembuatan timer-->
14. <div class="timer">
15. <div class="header\_text">Waktu</div>
16. <div class="curr\_time">60s</div>
17. </div>
18. </div>
19. <!-- pembuatan timer-->
20. <form>
21. <label for="input">Silahkan Pilih bahasa :</label>
22. <select name="languange" id="bahasa">
23. <option value="html">HTML</option>
24. <option value="javascript">Javascript</option>
25. <option value="css">CSS</option>
26. </select>
27. </form>
28. <!-- quotees / perintah yang harus ditulis oleh user-->
29. <div class="quote">Klik kolom di bawah untuk memulai game.</div>
30. <!-- text area untuk imputan user-->
31. <textarea class="input\_area" placeholder="mulai mengetik di sini..." oninput="processCurrentText()" onfocus="startGame()"></textarea>
32. <!-- Tombol reset ketika user ingin melakukan tes ulang-->
33. <button class="restart\_btn" onclick="resetValues()">Restart</button>
34. <div class="keterangan" id="hasilmu">
35. your result
36. </div>
37. <div class="result">
38. <!-- pembuatan teks untuk menampilkan akurasi-->
39. <div class="accuracy">
40. <div class="header\_text">Ketepatan(%)</div>
41. <div class="curr\_accuracy">100</div>
42. </div>
43. <!-- pembuatan teks untuk menampilkan kecepatan user kata per menit-->
44. <div class="wpm">
45. <div class="header\_text">WPM</div>
46. <div class="curr\_wpm">100</div>
47. </div>
48. <!-- pembuatan teks untuk menampilkan kecepatan user character per menit-->
49. <div class="cpm">
50. <div class="header\_text">CPM</div>
51. <div class="curr\_cpm">100</div>
52. </div>
53. <!-- pembuatan teks untuk menampilkan kesalahn tulis user-->
54. <div class="errors">
55. <div class="header\_text">Kesalahan</div>
56. <div class="curr\_errors">0</div>
57. </div>
58. </div>
59. </div>
60. <script src="game.js"></script>
61. <footer>
62. <h3>&copy; Rhizma Elliza Putri & Achmad Yusuf Al Ma'ruf </h3>
63. </footer>
64. </body>
65. </html>
66. JAVASCRIPT (game.js)
67. //mendefinisikan jumlah waktu untuk mengetik
68. var TIME\_LIMIT = 60;
69. // kata kata yang akan ditulis
70. var quotes\_array\_html = [
71. '<!DOCTYPE html> console.log("Hello world")',
72. '<link rel="stylesheet" href="/style.css"> <div class="container">',
73. '<textarea class="input\_area" placeholder="mulai mengetik di sini..." oninput="processCurrentText()" onfocus="startGame()"></textarea>',
74. '<script src="game.js"></script>',
75. '<button class="restart\_btn" onclick="resetValues()">Restart</button>'
76. ];
77. var quotes\_array\_js = [
78. 'var timer\_text = document.querySelector(".curr\_time");',
79. 'const charSpan = document.createElement("span");',
80. 'let correctCharacters = (characterTyped - (total\_errors + errors));',
81. 'var hasiltxt = document.querySelector("#hasilmu")',
82. 'function updateQuote()',
83. 'input\_area.disabled = true;'
84. ];
85. var quotes\_array\_css = [
86. 'display: flex; background-color: bisque; box-shadow: 10px 10px 80px rgb(156, 138, 138);',
87. 'border-bottom: 2px solid silver;',
88. 'flex-direction: column;',
89. '@media (max-width: 800px) text-decoration: underline;',
90. 'height: 60px; width: 70px; margin: 8px;',
91. 'padding: 12px; border-radius: 5% ; box-shadow: 10px 10px 5px silver;',
92. 'margin-bottom: 40px;'
93. ]
94. //mendefenisikan variabel penampung untuk kata kata yang akan ditulis
95. var quotes\_array = []
96. // menyelski elemet html dan menampungnya pada variabel penampung
97. var timer\_text = document.querySelector(".curr\_time");
98. var accuracy\_text = document.querySelector(".curr\_accuracy");
99. var error\_text = document.querySelector(".curr\_errors");
100. var cpm\_text = document.querySelector(".curr\_cpm");
101. var wpm\_text = document.querySelector(".curr\_wpm");
102. var quote\_text = document.querySelector(".quote");
103. var input\_area = document.querySelector(".input\_area");
104. var restart\_btn = document.querySelector(".restart\_btn");
105. var cpm\_group = document.querySelector(".cpm");
106. var wpm\_group = document.querySelector(".wpm");
107. var error\_group = document.querySelector(".errors");
108. var accuracy\_group = document.querySelector(".accuracy");
109. var timer\_group = document.querySelector(".timer");
110. var hasiltxt = document.querySelector("#hasilmu")
111. var timeLeft = TIME\_LIMIT;
112. var timeElapsed = 0;
113. var total\_errors = 0;
114. var errors = 0;
115. var accuracy = 0;
116. var characterTyped = 0;
117. var current\_quote = "";
118. var quoteNo = 0;
119. var timer = null;
120. //menyeleksi tipe quotes yang akan dipilih.
121. function selectquotes() {
122. //menampung nilai dari pilihan user untuk bahasa yang dipilih
123. var value\_quotes = document.querySelector("#bahasa").value;
124. //melakukan pengecekan pada nilai selection yang dipilih user dan menampunya pada quotes\_array
125. if (value\_quotes == 'html')
126. quotes\_array = quotes\_array\_html;
127. else if (value\_quotes == 'javascript')
128. quotes\_array = quotes\_array\_js;
129. else if (value\_quotes == 'css')
130. quotes\_array = quotes\_array\_css;
131. }
132. //mengupdate isi quotes
133. function updateQuote() {
134. quote\_text.textContent = null;
135. current\_quote = quotes\_array[quoteNo];
136. //mendapatkan teks yang akan ditulis lalu memisahklan setiap karakter elemen dengan span untuk setiap element yang telahdiketik oleh user
137. current\_quote.split('').forEach(char => {
138. const charSpan = document.createElement('span')
139. charSpan.innerText = char
140. quote\_text.appendChild(charSpan)
141. })
142. //memutar ke element awal
143. if (quoteNo < quotes\_array.length - 1)
144. quoteNo++;
145. else
146. quoteNo = 0;
147. }
148. //mendapatkan teks yang diketik oleh user
149. function processCurrentText() {
150. // mendapatkan teks dari inputan pengguna
151. curr\_input = input\_area.value;
152. curr\_input\_array = curr\_input.split('');
153. // menambahkan total karakter yang diketik
154. characterTyped++;
155. errors = 0;
156. //proses mengoreksi inpuan pengguna
157. quoteSpanArray = quote\_text.querySelectorAll('span');
158. quoteSpanArray.forEach((char, index) => {
159. let typedChar = curr\_input\_array[index]
160. // ketika tidak ada karakter yang diketik
161. if (typedChar == null) {
162. char.classList.remove('correct\_char');
163. char.classList.remove('incorrect\_char');
164. // ketika inputan character pengguna sama dengan quotes
165. } else if (typedChar === char.innerText) {
166. char.classList.add('correct\_char');
167. char.classList.remove('incorrect\_char');
168. // ketika inputan character pengguna tidak sama dengan quotes
169. } else {
170. char.classList.add('incorrect\_char');
171. char.classList.remove('correct\_char');
172. // meanmabahkan jumlah eror
173. errors++;
174. }
175. });
176. // menampilkan jumlah eror
177. error\_text.textContent = total\_errors + errors;
178. //menghitung ketepatan
179. //karakter benar = karakter yang ditulis - (total eror + eror)
180. let correctCharacters = (characterTyped - (total\_errors + errors));
181. //menghitung karakter benar = karakter benar / karakter yang ditulis \* 100
182. let accuracyVal = ((correctCharacters / characterTyped) \* 100);
183. //pembulatan hasil karakter benar
184. accuracy\_text.textContent = Math.round(accuracyVal);
185. // ketika teks diketik
186. if (curr\_input.length == current\_quote.length) {
187. updateQuote();
188. // mengupdate total eror
189. total\_errors += errors;
190. // menghapus input area
191. input\_area.value = "";
192. }
193. }
194. function updateTimer() {
195. if (timeLeft > 0) {
196. //menghitung mundur waktu
197. timeLeft--;
198. //menambahkan waktu yang telah berlalu
199. timeElapsed++;
200. // mengupdate text di tmer sesuai dengan waktu yang berjalan
201. timer\_text.textContent = timeLeft + "s";
202. } else {
203. //ketika game selesai
204. finishGame();
205. }
206. }
207. function finishGame() {
208. // menghentikan waktu
209. clearInterval(timer);
210. // menonaktifkan input area dan membersihkan inputanya
211. input\_area.disabled = true;
212. input\_area.value = "";
213. // menampilkan kata restart untuk memulia kembali pada quotes
214. quote\_text.textContent = "Klik restart untuk memulai kembali.";
215. // menampilkan restart button
216. restart\_btn.style.display = "block";
217. // menghitung cpm dengan cara (karakter yang ditulis / waktu pengerjaan) \* 60 lalu dibulatkan
218. cpm = Math.round(((characterTyped / timeElapsed) \* 60));
219. // menghitung wpm dengan cara (karakter yang ditulis /7)/ waktu pengerjaan)\* 60 lalu dibulatkan
220. //angka 7 merupakan rata rata jumlah character pada kata
221. wpm = Math.round((((characterTyped / 7) / timeElapsed) \* 60));
222. // memberikan nilai pada cpm dan wpm
223. cpm\_text.textContent = cpm;
224. wpm\_text.textContent = wpm;
225. // menampilkan elemen hasil, cpm, wpm, ketepatan dan eror
226. cpm\_group.style.display = "block";
227. wpm\_group.style.display = "block";
228. accuracy\_group.style.display = "block";
229. error\_group.style.display = "block";
230. hasiltxt.style.display = "block";
231. //menyembenyikan timer
232. timer\_group.style.display = "none";
233. }
234. //mereset nilai
235. function resetValues() {
236. timeLeft = TIME\_LIMIT;
237. timeElapsed = 0;
238. errors = 0;
239. total\_errors = 0;
240. accuracy = 0;
241. characterTyped = 0;
242. quoteNo = 0;
243. input\_area.disabled = false;
244. input\_area.value = "";
245. quote\_text.textContent = 'Klik kolom di bawah untuk memulai game.';
246. accuracy\_text.textContent = 100;
247. timer\_text.textContent = timeLeft + 's';
248. error\_text.textContent = 0;
249. restart\_btn.style.display = "none";
250. cpm\_group.style.display = "none";
251. wpm\_group.style.display = "none";
252. accuracy\_group.style.display = "none";
253. error\_group.style.display = "none";
254. timer\_group.style.display = "block";
255. hasiltxt.style.display = "none";
256. }
257. //memulai permainan
258. function startGame() {
259. // memangil fungsi select quotes
260. selectquotes();
261. // memangil fungsi reset nilai
262. resetValues();
263. // memangil fungsi update quotes
264. updateQuote();
265. // menghapus waktu yang lalu dan memulainya dengan waktu yang awal
266. clearInterval(timer);
267. timer = setInterval(updateTimer, 1000);
268. }
269. CSS (style.css)
270. /\* ... pengaturan pada body dengan height 100 dan bg bisque ... \*/
271. body {
272. background-color: bisque;
273. color: black;
274. text-align: center;
275. height: 100%;
276. }
277. /\* ... pengaturan pada id bahasa dengan ukuran lebar 100px dan ukuran font 15px ... \*/
278. #bahasa {
279. width: 100px;
280. font-size: 15px;
281. }
282. /\* ... pengaturan kontainer dengan menggunakan konsep flex dan margin kanan kiri 20px serta atas bawah yang auto ... \*/
283. .container {
284. display: flex;
285. flex-direction: column;
286. align-items: center;
287. background-color: white;
288. height: auto;
289. width: 60%;
290. padding: 20px;
291. border-radius: 10px;
292. box-shadow: 10px 10px 80px rgb(156, 138, 138);
293. margin: 20px auto;
294. }
295. /\* ... pengaturan pada class heading ... \*/
296. .heading {
297. font-size: 3rem;
298. color: black;
299. }
300. /\* ... pengaturan pada class keterangan... \*/
301. .keterangan {
302. font-size: 1.5 rem;
303. margin-bottom: 20px;
304. padding: 20px;
305. border-bottom: 2px solid silver;
306. width: 90%;
307. }
308. /\* ... pengaturan pada class header... \*/
309. .header {
310. display: flex;
311. align-items: center;
312. }
313. /\* ... pengaturan pada class timer, errors, accuracy, cpm, dan wpm... \*/
314. .timer,
315. .errors,
316. .accuracy,
317. .cpm,
318. .wpm {
319. background-color: peachpuff;
320. height: 60px;
321. width: 70px;
322. margin: 8px;
323. padding: 12px;
324. border-radius: 5%;
325. box-shadow: 10px 10px 5px silver;
326. margin-bottom: 40px;
327. }
328. /\* ... pengaturan pada class errors, accuracy, cpm, dan wpm... \*/
329. /\* ... dimana kelas tersebut akan disembunyikan di tampilan awal... \*/
330. .cpm,
331. .wpm,
332. .accuracy,
333. .errors {
334. display: none;
335. }
336. /\* ... pengaturan pada class header text... \*/
337. .header\_text {
338. text-transform: uppercase;
339. font-size: 0.6rem;
340. font-weight: 600;
341. }
342. /\* ... pengaturan pada class timer, errors, accuracy, cpm, dan wpm... \*/
343. .curr\_time,
344. .curr\_errors,
345. .curr\_accuracy,
346. .curr\_cpm,
347. .curr\_wpm {
348. font-size: 2.75rem;
349. }
350. /\* ... pengaturan pada class quote... \*/
351. .quote {
352. background-color: beige;
353. font-size: 1.5rem;
354. margin: 10px;
355. padding: 20px;
356. box-shadow: 10px 10px 5px silver;
357. width: 85%;
358. border-radius: 5px;
359. }
360. /\* ... pengaturan pada id hasilmu... \*/
361. #hasilmu {
362. font-size: 20px;
363. width: 80%;
364. display: none;
365. }
366. /\* ... pengaturan pada class input area... \*/
367. .input\_area {
368. background-color: blanchedalmond;
369. height: 80px;
370. width: 90%;
371. font-size: 1.5rem;
372. font-weight: 600;
373. margin: 15px;
374. padding: 20px;
375. border: 0px;
376. box-shadow: 10px 10px 5px silver;
377. margin: 20px;
378. margin-bottom: 40px;
379. border-radius: 5px;
380. }
381. /\* ... pengaturan pada class incorrect char... \*/
382. .incorrect\_char {
383. color: red;
384. text-decoration: underline;
385. }
386. /\* ... pengaturan pada class correct char... \*/
387. .correct\_char {
388. color: darkgreen;
389. }
390. /\* ... pengaturan pada class restart btn.. \*/
391. .restart\_btn {
392. display: none;
393. font-size: 1.5rem;
394. padding: 10px;
395. border: 0px;
396. width: 80%;
397. box-shadow: 10px 10px 5px silver;
398. margin-bottom: 20px;
399. }
400. /\* ... pengaturan pada class result.. \*/
401. .result {
402. display: flex;
403. }
404. /\* ... pengaturan responsive.. \*/
405. @media (max-width: 800px) {
406. .result {
407. flex-direction: column;
408. }}

**Link Github** : <https://github.com/yusufmaruf/FastTyping.git>

**Link demonstrasi :**

# **BAB IV**

# **PENUTUP**

## **4.1 Kesimpulan**

Perkembangan Ilmu Pengetahuan dan Teknologi yang pesat tidak jauh dari kata internet yang juga berpengaruh besar terhadap penyebaran informasi melalui berbagai jenis platform, salah satunya yaitu website. Dalam membuat suatu website atau laman diperlukan suatu bahasa yang berisi perintah dan dipahami oleh komputer sehingga harus menggunakan bahasa yang tepat, beberapa di antaranya yaitu HTML, Javascript, dan CSS. **HTML** merupakan singkatan dari **Hypertext Markup Language** yang merupakan bahasa format dari tag XML atau Extended Markup Language dan dijadikan sebagai standar umum dari suatu halaman web. Setiap baris perintah selalu memiliki fungsi sendiri dan semua itu dieksekusi dan ditampilkan sesuai dengan urutan perintah yang ditulis. Sedangkan Javascript digunakan untuk mengubah dan mengkreasikan tampilan website pada laman menjadi lebih menarik dengan beberapa fungsi dengan bantuan CSS yang digunakan untuk memformat dokumen HTML dengan memisahkannya dari tampilan visual. Tidak hanya perlu menguasai bahasa pemograman saja tetapi seorang programmer juga membutuhkan skill untuk mengetik dengan cepat dan tepat. Dengan membuat program Fast Coding diatas diharapkan mahasiswa dapat melatih skill meng-*coding* dimana mahasiswa dapat mengetik coding dengan menggunakan 10 jari dengan cepat dan tepat.

## **4.2 Saran**

Kemampuan mahasiswa dalam pemrograman seharusnya dimanfaatkan dengan membuat program yang bersifat edukatif dan memberi banyak manfaat untuk penggunanya. Dengan begitu, kita dapat berkontribusi untuk negara kita – Indonesia untuk bersaing secara global di bidang Ilmu Pengetahuan dan Teknologi sehingga kemampuan dalam berbahasa pemrograman harus terus dilatih untuk meningkatkan skill.