

**LAPORAN PEMBUATAN PROGRAM  
GAME FAST CODING**



**NAMA KELOMPOK : KELOMPOK 10**

|                               |                    |
|-------------------------------|--------------------|
| <b>Rhizma Elliza Putri</b>    | <b>20082010120</b> |
| <b>Achmad Yusuf Al Ma'ruf</b> | <b>20082010148</b> |

**PROGRAM STUDI S1 SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UPN "VETERAN" JAWA TIMUR  
SURABAYA  
2022**

## DAFTAR ISI

|   |           |
|---|-----------|
| <b>BAB I .....</b>                                    | <b>3</b>  |
| <b>PENDAHULUAN .....</b>                              | <b>3</b>  |
| <b>1.1 Latar Belakang Masalah .....</b>               | <b>3</b>  |
| <b>1.2 Identifikasi Masalah .....</b>                 | <b>4</b>  |
| <b>1.3 Pembatasan Masalah .....</b>                   | <b>4</b>  |
| <b>1.4 Rumusan Masalah .....</b>                      | <b>5</b>  |
| <b>1.5 Manfaat Pengembangan .....</b>                 | <b>5</b>  |
| <b>BAB II .....</b>                                   | <b>6</b>  |
| <b>ANALISA DAN PERANCANGAN PROGRAM .....</b>          | <b>6</b>  |
| <b>2.1 Diskripsi Program .....</b>                    | <b>6</b>  |
| <b>2.2 Rancang Bangun Program .....</b>               | <b>7</b>  |
| <b>2.3 Alur Kerja Program .....</b>                   | <b>17</b> |
| <b>BAB III .....</b>                                  | <b>19</b> |
| <b>PENGUJIAN , SCRIPT, DAN TAMPILAN PROGRAM .....</b> | <b>19</b> |
| <b>3.1 Tampilan Program Dan Pengujian .....</b>       | <b>19</b> |
| <b>3.2 SOURCE CODE .....</b>                          | <b>21</b> |
| <b>BAB IV .....</b>                                   | <b>41</b> |
| <b>PENUTUP .....</b>                                  | <b>41</b> |
| <b>4.1 KESIMPULAN .....</b>                           | <b>41</b> |
| <b>4.2 SARAN .....</b>                                | <b>41</b> |

# 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 aplikasi 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 membutuhkan 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 kemampuannya dengan menggunakan 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 memainkannya.
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 kefokusannya
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**

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

- a) Visual studio code
- b) Xampp
- c) Internet
- d) Laptop
- e) Keyboard
- f) Mouse

### 2. 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 :

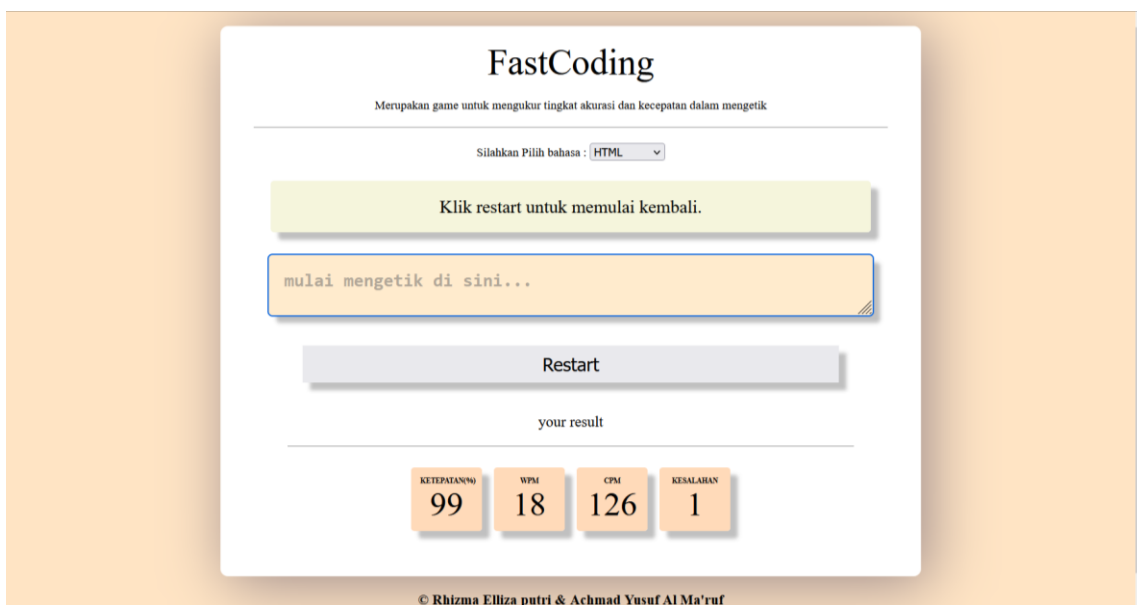


Tampilan ketika mengetik dengan salah :





Tampilan akhir ketika program telah selesai :



### 3. Pembuatan Program

#### a) Pembuatan file HTML

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

1. **Heading** : bagian ini berisi judul dan dikrispi program serta *timer* untuk menunjukkan waktu yang tersisa saat mengetik

- The screenshot shows a web browser window with a game interface. The interface includes a title 'Fast Coding', a description 'Memupayakan game untuk mengukur tingkat akurasi dan kecepatan dalam mengetik', and a 'Start Game' button. The browser's developer tools are open, showing the HTML structure of the page. The HTML structure is as follows:

```

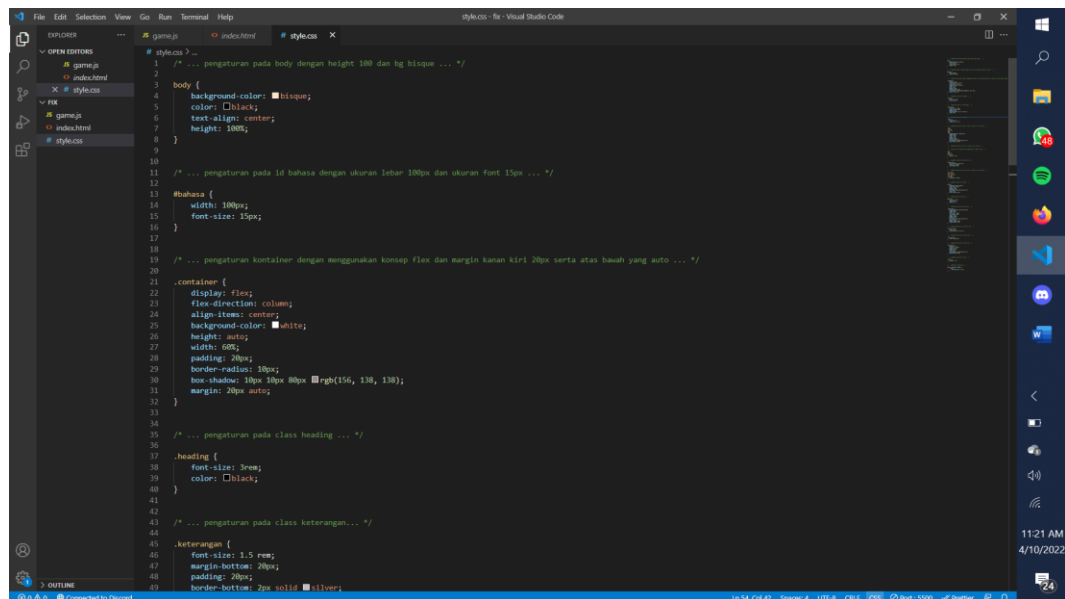
<html lang="en">
  <head>
    <title>Fast Coding</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <!-- container dibuat untuk memberi ruang yang akan menampung konten -->
    <div class="container">
      <div class="header">
        <div class="keterangan">Memupayakan game untuk mengukur tingkat akurasi dan kecepatan dalam mengetik</div>
        <div class="button">
          <button type="button">Start Game</button>
        </div>
      </div>
      <div class="form">
        <div class="input">
          <input type="text" value="Silahkan Pilih bahasa :"/>
        </div>
        <div class="select">
          <select name="language" id="bahasa">
            <option value="html">HTML</option>
            <option value="javascript">JavaScript</option>
            <option value="css">CSS</option>
          </select>
        </div>
        <div class="button">
          <button type="button">Start Game</button>
        </div>
      </div>
      <div class="result">
        <div class="accuracy">
          <div class="header">Akurasi</div>
          <div class="value">100</div>
        </div>
        <div class="speed">
          <div class="header">Kecepatan</div>
          <div class="value">100</div>
        </div>
      </div>
    </div>
  </body>
</html>

```

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 menunjukkan 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 /* ... pengaturan pada body dengan height 100 dan bg bilauque ... */
2
3 body {
4   background-color: #1a2b3c;
5   color: #ffffff;
6   text-align: center;
7   height: 100%;
8 }
9
10
11 /* ... pengaturan pada id bahasa dengan ukuran lebar 100px dan ukuran font 15px ... */
12
13 #bahasa {
14   width: 100px;
15   font-size: 15px;
16 }
17
18
19 /* ... pengaturan kontainer dengan menggunakan konsep flex dan margin kanan kiri 20px serta atas bawah yang auto ... */
20
21 .container {
22   display: flex;
23   flex-direction: column;
24   align-items: center;
25   background-color: #ffffff;
26   height: auto;
27   width: 60%;
28   padding: 20px;
29   border-radius: 10px;
30   box-shadow: 10px 10px 8px #1a2b3c;
31   margin: 20px auto;
32 }
33
34
35 /* ... pengaturan pada class heading ... */
36
37 .heading {
38   font-size: 3rem;
39   color: #1a2b3c;
40 }
41
42
43 /* ... pengaturan pada class keterangan... */
44
45 .keterangan {
46   font-size: 1.5 rem;
47   margin-bottom: 20px;
48   padding: 20px;
49   border-bottom: 2px solid #1a2b3c;
50 }
```

### c) 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.



tergantung apakah karakter yang telah diketik benar atau salah.

Elemen tersebut ditambahkan ke variable `quote_text`

```
68
69 function updateQuote() {
70   quote_text.textContent = null;
71   current_quote = quotes_array[quoteNo];
72   //mendapatkan teks yang akan ditulis lalu memisahkan setiap karakter elemen dengan span untuk setiap element yang telah diketik oleh u
73   current_quote.split('').forEach(char => {
74     const charSpan = document.createElement('span')
75     charSpan.innerText = char
76     quote_text.appendChild(charSpan)
77   })
78
79   //memutar ke element awal
80   if (quoteNo < quotes_array.length - 1)
81     quoteNo++;
82   else
83     quoteNo = 0;
84 }
85
```

### 3. Mendapatkan teks yang saat ini diketik pengguna

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

### 4. 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`

### 5. 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 mengulangnya. Elemen tersebut kemudian menerapkan kelas yang dibuat tergantung pada apakah itu sama dengan teks yang diketik.

### 6. Menghitung kesalahan dan akurasi

Setiap kali pengguna membuat kesalahan saat mengetik, variabel `error` bertambah. Ini digunakan untuk menghitung nilai akurasi

dengan membagi jumlah karakter yang diketik dengan benar dengan jumlah total karakter yang diketik oleh pengguna.

## 7. 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.

```
87
88 function processCurrentText() {
89
90     // mendapatkan teks dari inputan pengguna
91     curr_input = input_area.value;
92     curr_input_array = curr_input.split('');
93
94     // menambahkan total karakter yang diketik
95     characterTyped++;
96
97     errors = 0;
98     //proses mengoreksi inputan pengguna
99
100    quoteSpanArray = quote_text.querySelectorAll('span');
101    quoteSpanArray.forEach((char, index) => {
102        let typedChar = curr_input_array[index]
103
104        // ketika tidak ada karakter yang diketik
105        if (typedChar == null) {
106            char.classList.remove('correct_char');
107            char.classList.remove('incorrect_char');
108
109            // ketika inputan character pengguna sama dengan quotes
110        } else if (typedChar === char.innerText) {
111            char.classList.add('correct_char');
112            char.classList.remove('incorrect_char');
113
114            // ketika inputan character pengguna tidak sama dengan quotes
115        } else {
116            char.classList.add('incorrect_char');
117            char.classList.remove('correct_char');
118
119            // menambahkan jumlah eror
120            errors++;
121        }
122    });
123
124    // menampilkan jumlah eror
125    error_text.textContent = total_errors + errors;
126
127    //menghitung ketepatan
128    //karakter benar = karakter yang ditulis - (total eror + eror)
129    let correctCharacters = (characterTyped - (total_errors + errors));
130    //menghitung karakter benar = karakter benar / karakter yang ditulis * 100
131    let accuracyVal = ((correctCharacters / characterTyped) * 100);
132    //pembulatan hasil karakter benar
133    accuracy_text.textContent = Math.round(accuracyVal);
134
135    // ketika teks diketik
136    if (curr_input.length == current_quote.length) {
137        updateQuote();
138
139        // mengupdate total eror
140        total_errors += errors;
141
142        // menghapus input area
143        input_area.value = "";
144    }
145 }
```

## 8. Memulai permainan baru

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

## 9. Setel ulang semua nilai

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

```
201
202 //mereset nilai
203 function resetValues() {
204     timeLeft = TIME_LIMIT;
205     timeElapsed = 0;
206     errors = 0;
207     total_errors = 0;
208     accuracy = 0;
209     characterTyped = 0;
210     quoteNo = 0;
211     input_area.disabled = false;
212
213     input_area.value = "";
214     quote_text.textContent = 'Klik kolom di bawah untuk memulai game.';
215     accuracy_text.textContent = 100;
216     timer_text.textContent = timeLeft + 's';
217     error_text.textContent = 0;
218     restart_btn.style.display = "none";
219     cpm_group.style.display = "none";
220     wpm_group.style.display = "none";
221     accuracy_group.style.display = "none";
222     error_group.style.display = "none";
223     timer_group.style.display = "block";
224     hasilTxt.style.display = "none";
225
226 }
227
```

## 10. Perbarui teks kutipan Teks

Kutipan baru ditampilkan dengan memanggil fungsi `updateQuote()`.

## 11. Membuat timer baru

Melacak 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()`.

```
228
229
230 function startGame() {
231     // memanggil fungsi select quotes
232     selectQuotes();
233     // memanggil fungsi reset nilai
234     resetValues();
235     // memanggil fungsi update quotes
236     updateQuote();
237
238     // menghapus waktu yang lalu dan memulainya dengan waktu yang awal
239     clearInterval(timer);
240     timer = setInterval(updateTimer, 1000);
241 }
```

## 12. Memperbarui timer

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

## 13. Perbarui nilai waktu

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

#### 14. Menyelesaikan Permainan

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

```
146
147 function updateTimer() {
148     if (timeLeft > 0) {
149         //menghitung mundur waktu
150
151         timeLeft--;
152
153
154         //menambahkan waktu yang telah berlalu
155         timeElapsed++;
156
157
158         // mengupdate text di tmer sesuai dengan waktu yang berjalan
159         timer_text.textContent = timeLeft + "s";
160     } else {
161         //ketika game selesai
162         finishGame();
163     }
164 }
165
```

#### 15. Menyelesaikan permainan

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

#### 16. Menghapus

Timer Instance timer yang dibuat sebelumnya akan dihapus.

#### 17. 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'.

#### 18. Menghitung CPM dan WPM:

Karakter Per Menit (CPM) dihitung dengan membagi jumlah karakter yang diketik dengan waktu yang berlalu dan kemudian mengalikannya 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

```
166 function finishGame() {
167   // menghentikan waktu
168   clearInterval(timer);
169
170   // menonaktifkan input area dan membersihkan inputnya
171   input_area.disabled = true;
172   input_area.value = "";
173
174   // menampilkan kata restart untuk memulai kembali pada quotes
175   quote_text.textContent = "Klik restart untuk memulai kembali.";
176
177   // menampilkan restart button
178   restart_btn.style.display = "block";
179
180   // menghitung cpm dengan cara (karakter yang ditulis / waktu pengerjaan) * 60 lalu dibulatkan
181   cpm = Math.round(((characterTyped / timeElapsed) * 60));
182   // menghitung cpm dengan cara (karakter yang ditulis // waktu pengerjaan) * 60 lalu dibulatkan
183   // angka 7 menunjukan rata rata jumlah character pada kata
184   wpm = Math.round(((characterTyped / 7) / timeElapsed) * 60);
185
186   // memberikan nilai pada cpm dan wpm
187   cpm_text.textContent = cpm;
188   wpm_text.textContent = wpm;
189
190   // menampilkan elemen hasil, cpm, wpm, ketepatan dan error
191   cpm_group.style.display = "block";
192   wpm_group.style.display = "block";
193   accuracy_group.style.display = "block";
194   error_group.style.display = "block";
195   hasilText.style.display = "block";
196
197   // menyembunyikan timer
198   timer_group.style.display = "none";
199 }
```

## 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:

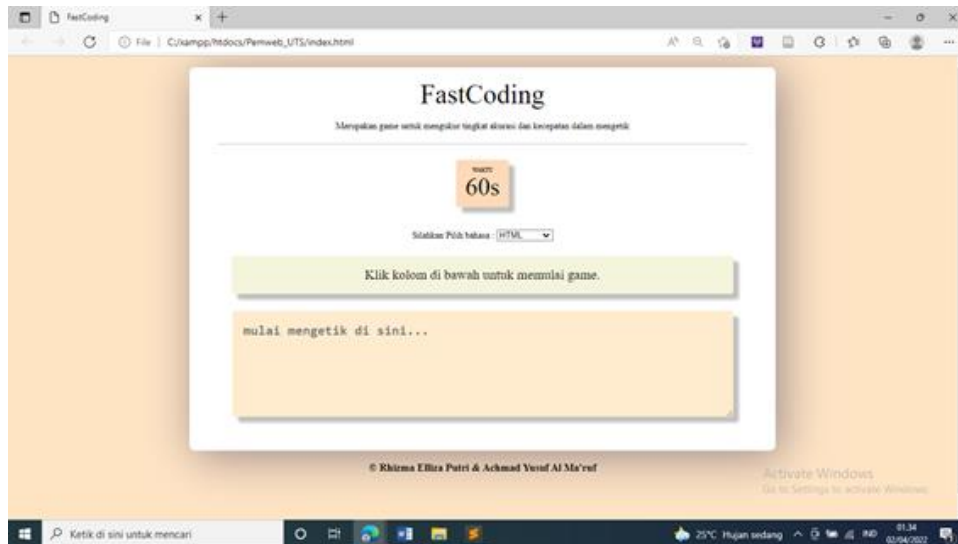
- a) Ketepatan
- b) WPM
- c) CPM
- d) Kesalahan

## BAB III

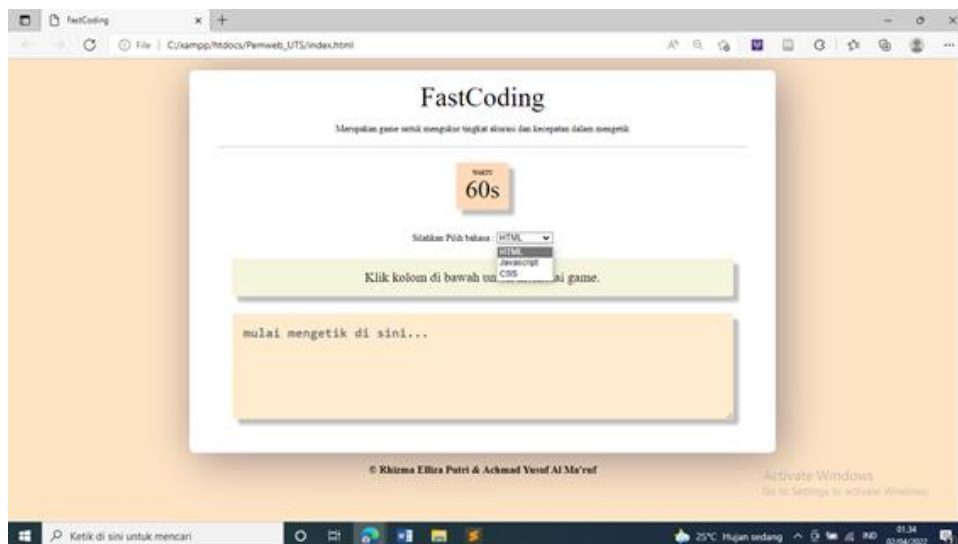
### PENGUJIAN , SCRIPT, DAN TAMPILAN PROGRAM

#### 3.1 Tampilan Program Dan Pengujian

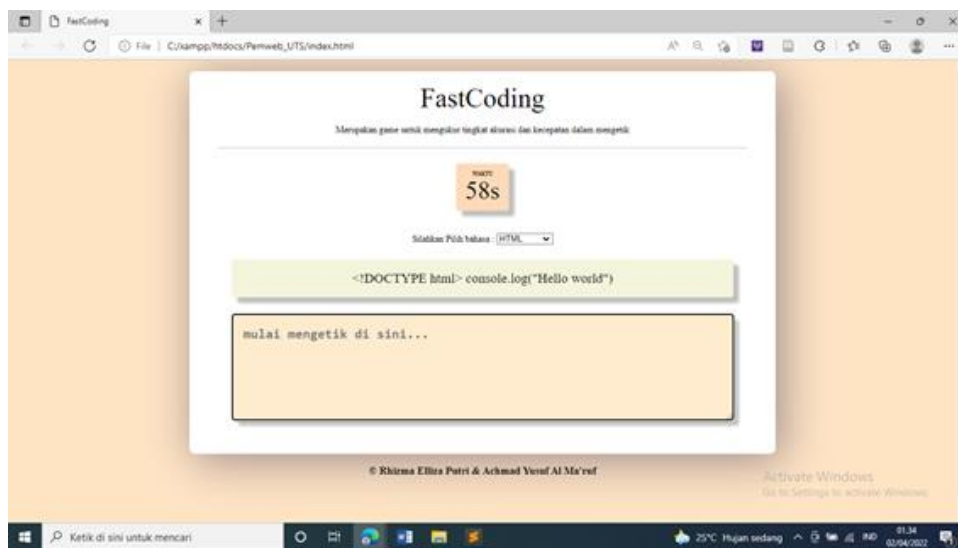
##### A. Tampilan Awal Program



##### B. Tampilan Saat Memilih Bahasa



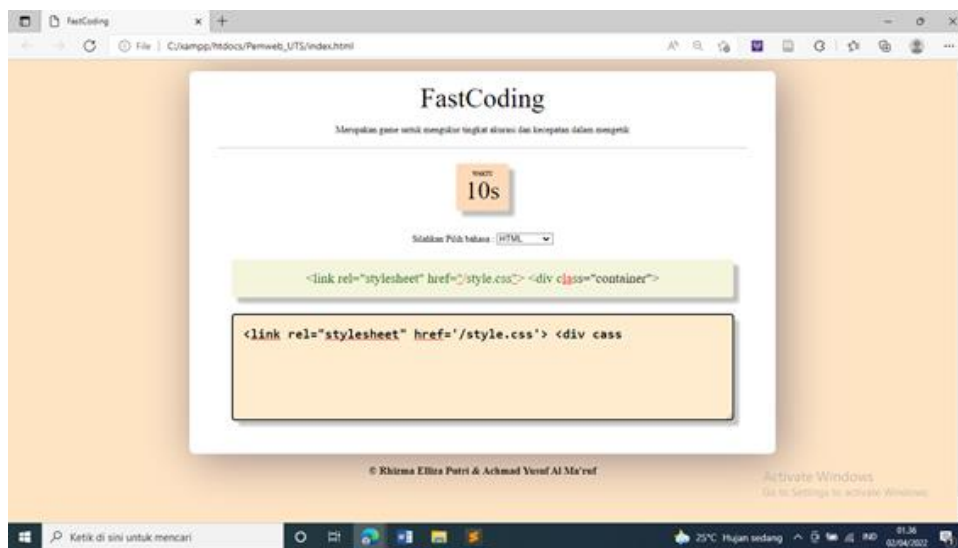
### C. Tampilan Saat Game Dimulai (Input area diklik)



### D. Tampilan Ketika Semua Huruf Diketik Secara Tepat



## E. Tampilan Ketika Terdapat Huruf Yang Diketik Dengan Salah



## F. Tampilan Saat Game Selesai Dimainkan



## 3.2 SOURCE CODE

### A. HTML (index.html)

1. `<html lang="en">`
2. `<head>`
3. `<title>FastCoding</title>`
4. `<link rel="stylesheet" href="style.css">`
5. `</head>`
- 6.

```
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.
20.     <!-- pembuatan timer-->
21.     <form>
22.       <label for="input">Silahkan Pilih bahasa :</label>
23.       <select name="languange" id="bahasa">
24.         <option value="html">HTML</option>
25.         <option value="javascript">Javascript</option>
26.         <option value="css">CSS</option>
27.       </select>
28.     </form>
29.
30.     <!-- quotees / perintah yang harus ditulis oleh user-->
31.     <div class="quote">Klik kolom di bawah untuk memulai game.</div>
```

```
32.     <!-- text area untuk imputan user-->
33.     <textarea class="input_area" placeholder="mulai mengetik di sini..."
        oninput="processCurrentText()" onfocus="startGame()"></textarea>
34.
35.     <!-- Tombol reset ketika user ingin melakukan tes ulang-->
36.     <button class="restart_btn" onclick="resetValues()">Restart</button>
37.     <div class="keterangan" id="hasilmu">
38.         your result
39.     </div>
40.     <div class="result">
41.         <!-- pembuatan teks untuk menampilkan akurasi-->
42.         <div class="accuracy">
43.             <div class="header_text">Ketepatan(%)</div>
44.             <div class="curr_accuracy">100</div>
45.         </div>
46.
47.         <!-- pembuatan teks untuk menampilkan kecepatan user kata per
            menit-->
48.         <div class="wpm">
49.             <div class="header_text">WPM</div>
50.             <div class="curr_wpm">100</div>
51.         </div>
52.
53.         <!-- pembuatan teks untuk menampilkan kecepatan user character
            per menit-->
54.         <div class="cpm">
55.             <div class="header_text">CPM</div>
```

```

56.     <div class="curr_cpm">100</div>
57.     </div>
58.
59.     <!-- pembuatan teks untuk menampilkan kesalahn tulis user-->
60.     <div class="errors">
61.         <div class="header_text">Kesalahan</div>
62.         <div class="curr_errors">0</div>
63.     </div>
64. </div>
65. </div>
66. <script src="game.js"></script>
67. <footer>
68.     <h3>&copy; Rhizma Elliza Putri & Achmad Yusuf Al Ma'ruf </h3>
69. </footer>
70.</body>
71.</html>

```

## B. JAVASCRIPT (game.js)

```

1. //mendefinisikan jumlah waktu untuk mengetik
2. var TIME_LIMIT = 60;
3.
4. // kata kata yang akan ditulis
5. var quotes_array_html = [
6.     '<!DOCTYPE html> console.log("Hello world")',
7.     '<link rel="stylesheet" href="/style.css"> <div class="container">',
8.     '<textarea class="input_area" placeholder="mulai mengetik di sini..."
    oninput="processCurrentText()" onfocus="startGame()"></textarea>',

```



```
9.   '<script src="game.js"></script>',
10.   '<button class="restart_btn" onclick="resetValues()">Restart</button>'
11.];
12.var quotes_array_js = [
13.   'var timer_text = document.querySelector(".curr_time");',
14.   'const charSpan = document.createElement("span");',
15.   'let correctCharacters = (characterTyped - (total_errors + errors));',
16.   'var hasiltxt = document.querySelector("#hasilmu");',
17.   'function updateQuote()',
18.   'input_area.disabled = true;',
19.
20.];
21.var quotes_array_css = [
22.
23.   'display: flex; background-color: bisque; box-shadow: 10px 10px 80px
    rgb(156, 138, 138);',
24.   'border-bottom: 2px solid silver;',
25.   'flex-direction: column;',
26.   '@media (max-width: 800px) text-decoration: underline;',
27.   'height: 60px; width: 70px; margin: 8px;',
28.   'padding: 12px; border-radius: 5% ; box-shadow: 10px 10px 5px
    silver;',
29.   'margin-bottom: 40px;'
30.
31. ]
32. //mendefenisikan variabel penampung untuk kata kata yang akan ditulis
33.var quotes_array = []
```

```
34. //menyeleksi tipe quotes yang akan dipilih.
35.function selectquotes() {
36. //menampung nilai dari pilihan user untuk bahasa yang dipilih
37. var value_quotes = document.querySelector("#bahasa").value;
38. //melakukan pengecekan pada nilai selection yang dipilih user dan
    menampungnya pada quotes_array
39. if (value_quotes == 'html')
40.     quotes_array = quotes_array_html;
41. else if (value_quotes == 'javascript')
42.     quotes_array = quotes_array_js;
43. else if (value_quotes == 'css')
44.     quotes_array = quotes_array_css;
45.}
46.
47.// menyelski elemet html dan menampungnya pada variabel penampung
48.var timer_text = document.querySelector(".curr_time");
49.var accuracy_text = document.querySelector(".curr_accuracy");
50.var error_text = document.querySelector(".curr_errors");
51.var cpm_text = document.querySelector(".curr_cpm");
52.var wpm_text = document.querySelector(".curr_wpm");
53.var quote_text = document.querySelector(".quote");
54.var input_area = document.querySelector(".input_area");
55.var restart_btn = document.querySelector(".restart_btn");
56.var cpm_group = document.querySelector(".cpm");
57.var wpm_group = document.querySelector(".wpm");
58.var error_group = document.querySelector(".errors");
59.var accuracy_group = document.querySelector(".accuracy");
```

```
60.var timer_group = document.querySelector(".timer");
61.var hasiltxt = document.querySelector("#hasilmu")
62.
63.var timeLeft = TIME_LIMIT;
64.var timeElapsed = 0;
65.var total_errors = 0;
66.var errors = 0;
67.var accuracy = 0;
68.var characterTyped = 0;
69.var current_quote = "";
70.var quoteNo = 0;
71.var timer = null;
72.
73.//mengupdate isi quotes
74.
75.function updateQuote() {
76.  quote_text.textContent = null;
77.  current_quote = quotes_array[quoteNo];
78.  //mendapatkan teks yang akan ditulis lalu memisahkan setiap karakter
    elemen dengan span untuk setiap element yang telah diketik oleh user
79.  current_quote.split("").forEach(char => {
80.    const charSpan = document.createElement('span')
81.    charSpan.innerText = char
82.    quote_text.appendChild(charSpan)
83.  })
84.
85.  //memutar ke element awal
```

```
86. if (quoteNo < quotes_array.length - 1)
87.     quoteNo++;
88. else
89.     quoteNo = 0;
90.}
91.
92.//mendapatkan teks yang diketik oleh user
93.
94.function processCurrentText() {
95.
96.    // mendapatkan teks dari inputan pengguna
97.    curr_input = input_area.value;
98.    curr_input_array = curr_input.split("");
99.
100.    // menambahkan total karakter yang diketik
101.    characterTyped++;
102.
103.    errors = 0;
104.    //proses mengoreksi inpuan pengguna
105.
106.    quoteSpanArray = quote_text.querySelectorAll('span');
107.    quoteSpanArray.forEach((char, index) => {
108.        let typedChar = curr_input_array[index]
109.
110.        // ketika tidak ada karakter yang diketik
111.        if (typedChar == null) {
112.            char.classList.remove('correct_char');
```

```
113.     char.classList.remove('incorrect_char');
114.
115.     // ketika inputan character pengguna sama dengan quotes
116. } else if (typedChar === char.innerText) {
117.     char.classList.add('correct_char');
118.     char.classList.remove('incorrect_char');
119.
120.     // ketika inputan character pengguna tidak sama dengan quotes
121. } else {
122.     char.classList.add('incorrect_char');
123.     char.classList.remove('correct_char');
124.
125.     // menambahkan jumlah eror
126.     errors++;
127. }
128. });
129.
130. // menampilkan jumlah eror
131. error_text.textContent = total_errors + errors;
132.
133. //menghitung ketepatan
134. //karakter benar = karakter yang ditulis - (total eror + eror)
135. let correctCharacters = (characterTyped - (total_errors + errors));
136. //menghitung karakter benar = karakter benar / karakter yang ditulis *
    100
137. let accuracyVal = ((correctCharacters / characterTyped) * 100);
138. //pembulatan hasil karakter benar
```

```
139. accuracy_text.textContent = Math.round(accuracyVal);
140.
141. // ketika teks diketik
142. if (curr_input.length == current_quote.length) {
143.     updateQuote();
144.
145.     // mengupdate total eror
146.     total_errors += errors;
147.
148.     // menghapus input area
149.     input_area.value = "";
150. }
151. }
152.
153. function updateTimer() {
154.     if (timeLeft > 0) {
155.         //menghitung mundur waktu
156.
157.         timeLeft--;
158.
159.         //menambahkan waktu yang telah berlalu
160.         timeElapsed++;
161.
162.         // mengupdate text di tmer sesuai dengan waktu yang berjalan
163.         timer_text.textContent = timeLeft + "s";
164.     } else {
165.         //ketika game selesai
```

```
166.     finishGame();
167. }
168. }
169.
170. function finishGame() {
171.     // menghentikan waktu
172.     clearInterval(timer);
173.
174.     // menonaktifkan input area dan membersihkan inputanya
175.     input_area.disabled = true;
176.     input_area.value = "";
177.
178.     // menampilkan kata restart untuk memulia kembali pada quotes
179.     quote_text.textContent = "Klik restart untuk memulai kembali.";
180.
181.     // menampilkan restart button
182.     restart_btn.style.display = "block";
183.
184.     // menghitung cpm dengan cara (karakter yang ditulis / waktu
        pengerjaan) * 60 lalu dibulatkan
185.     cpm = Math.round(((characterTyped / timeElapsed) * 60));
186.     // menghitung cpm dengan cara (karakter yang ditulis /7)/ waktu
        pengerjaan)* 60 lalu dibulatkan
187.     //angka 7 merupakan rata rata jumlah character pada kata
188.     wpm = Math.round((((characterTyped / 7) / timeElapsed) * 60));
189.
190.     // memberikan nilai pada cpm dan wpm
```

```
191.   cpm_text.textContent = cpm;
192.   wpm_text.textContent = wpm;
193.
194.   // menampilkan elemen hasil, cpm, wpm, ketepatan dan eror
195.   cpm_group.style.display = "block";
196.   wpm_group.style.display = "block";
197.   accuracy_group.style.display = "block";
198.   error_group.style.display = "block";
199.   hasiltxt.style.display = "block";
200.
201.   //menyembenyikan timer
202.   timer_group.style.display = "none";
203. }
204.
205. //mereset nilai
206. function resetValues() {
207.   timeLeft = TIME_LIMIT;
208.   timeElapsed = 0;
209.   errors = 0;
210.   total_errors = 0;
211.   accuracy = 0;
212.   characterTyped = 0;
213.   quoteNo = 0;
214.   input_area.disabled = false;
215.
216.   input_area.value = "";
217.   quote_text.textContent = 'Klik kolom di bawah untuk memulai game.';
```



```
218. accuracy_text.textContent = 100;
219. timer_text.textContent = timeLeft + 's';
220. error_text.textContent = 0;
221. restart_btn.style.display = "none";
222. cpm_group.style.display = "none";
223. wpm_group.style.display = "none";
224. accuracy_group.style.display = "none";
225. error_group.style.display = "none";
226. timer_group.style.display = "block";
227. hasiltxt.style.display = "none";
228.
229. }
230. //memulai permainan
231.
232. function startGame() {
233.     // memanggil fungsi select quotes
234.     selectquotes();
235.     // memanggil fungsi reset nilai
236.     resetValues();
237.     // memanggil fungsi update quotes
238.     updateQuote();
239.
240.     // menghapus waktu yang lalu dan memulainya dengan waktu yang
        awal
241.     clearInterval(timer);
242.     timer = setInterval(updateTimer, 1000);}
```

### C. CSS (style.css)

```
1. /* ... pengaturan pada body dengan height 100 dan bg bisque ... */
2.
3. body {
4.     background-color: bisque;
5.     color: black;
6.     text-align: center;
7.     height: 100%;
8. }
9.
10./* ... pengaturan pada id bahasa dengan ukuran lebar 100px dan ukuran
    font 15px ... */
11.
12.#bahasa {
13.    width: 100px;
14.    font-size: 15px;
15.}
16.
17./* ... pengaturan kontainer dengan menggunakan konsep flex dan margin
    kanan kiri 20px serta atas bawah yang auto ... */
18.
19..container {
20.    display: flex;
21.    flex-direction: column;
22.    align-items: center;
23.    background-color: white;
24.    height: auto;
```

```
25. width: 60%;
26. padding: 20px;
27. border-radius: 10px;
28. box-shadow: 10px 10px 80px rgb(156, 138, 138);
29. margin: 20px auto;
30.}
31.
32./* ... pengaturan pada class heading ... */
33.
34..heading {
35. font-size: 3rem;
36. color: black;
37.}
38.
39./* ... pengaturan pada class keterangan... */
40.
41..keterangan {
42. font-size: 1.5 rem;
43. margin-bottom: 20px;
44. padding: 20px;
45. border-bottom: 2px solid silver;
46. width: 90%;
47.}
48.
49./* ... pengaturan pada class header... */
50.
51..header {
```

```
52. display: flex;
53. align-items: center;
54.}
55.
56./* ... pengaturan pada class timer, errors, accuracy, cpm, dan wpm... */
57.
58..timer,
59..errors,
60..accuracy,
61..cpm,
62..wpm {
63. background-color: peachpuff;
64. height: 60px;
65. width: 70px;
66. margin: 8px;
67. padding: 12px;
68. border-radius: 5%;
69. box-shadow: 10px 10px 5px silver;
70. margin-bottom: 40px;
71.}
72.
73./* ... pengaturan pada class errors, accuracy, cpm, dan wpm... */
74.
75./* ... dimana kelas tersebut akan disembunyikan di tampilan awal... */
76.
77..cpm,
78..wpm,
```

```
79..accuracy,
80..errors {
81.  display: none;
82.}
83.
84./* ... pengaturan pada class header text... */
85.
86..header_text {
87.  text-transform: uppercase;
88.  font-size: 0.6rem;
89.  font-weight: 600;
90.}
91.
92./* ... pengaturan pada class timer, errors, accuracy, cpm, dan wpm... */
93.
94..curr_time,
95..curr_errors,
96..curr_accuracy,
97..curr_cpm,
98..curr_wpm {
99.  font-size: 2.75rem;
100. }
101.
102. /* ... pengaturan pada class quote... */
103.
104. .quote {
105.  background-color: beige;
```

```
106.   font-size: 1.5rem;
107.   margin: 10px;
108.   padding: 20px;
109.   box-shadow: 10px 10px 5px silver;
110.   width: 85%;
111.   border-radius: 5px;
112. }
113.
114. /* ... pengaturan pada id hasilmu... */
115.
116. #hasilmu {
117.   font-size: 20px;
118.   width: 80%;
119.   display: none;
120. }
121.
122. /* ... pengaturan pada class input area... */
123.
124. .input_area {
125.   background-color: blanchedalmond;
126.   height: 80px;
127.   width: 90%;
128.   font-size: 1.5rem;
129.   font-weight: 600;
130.   margin: 15px;
131.   padding: 20px;
132.   border: 0px;
```

```
133.   box-shadow: 10px 10px 5px silver;
134.   margin: 20px;
135.   margin-bottom: 40px;
136.   border-radius: 5px;
137. }
138.
139. /* ... pengaturan pada class incorrect char... */
140.
141. .incorrect_char {
142.   color: red;
143.   text-decoration: underline;
144. }
145.
146. /* ... pengaturan pada class correct char... */
147.
148. .correct_char {
149.   color: darkgreen;
150. }
151.
152. /* ... pengaturan pada class restart btn.. */
153.
154. .restart_btn {
155.   display: none;
156.   font-size: 1.5rem;
157.   padding: 10px;
158.   border: 0px;
159.   width: 80%;
```

```
160.   box-shadow: 10px 10px 5px silver;
161.   margin-bottom: 20px;
162. }
163.
164. /* ... pengaturan pada class result.. */
165.
166. .result {
167.   display: flex;
168. }
169.
170. /* ... pengaturan responsive.. */
171.
172. @media (max-width: 800px) {
173.   .result {
174.     flex-direction: column;
175.   }
```

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



## 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 pemrograman 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 penggunaanya. 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.