Jumadi

1. Typing Tutor Game

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Typing Tutor Advanced</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <h1>Typing Tutor Advanced</h1>
  <p>Latih kecepatan dan ketepatan mengetik kamu!</p>

  <label for="difficulty">Pilih tingkat kesulitan:</label>
  <select id="difficulty">
    <option value="easy">Mudah</option>
    <option value="medium" selected>Sedang</option>
    <option value="hard">Susah</option>
  </select>

  <div id="wordDisplay" class="word-display"></div>
  <input type="text" id="inputBox" placeholder="Ketik kata di sini" autocomplete="off"
disabled />

  <div class="info">
    <p>Waktu: <span id="time">60</span> detik</p>
    <p>Skor: <span id="score">0</span></p>
    <p>Kecepatan: <span id="wpm">0</span> WPM</p>
  </div>

  <button id="startBtn">Mulai</button>
  <button id="resetBtn" disabled>Reset</button>

  <script src="script.js"></script>
</body>
</html>
```

style.css

```
body {
  font-family: Arial, sans-serif;
  text-align: center;
  padding: 30px;
  background-color: #eef6fc;
}

h1 {
  margin-bottom: 10px;
}

label {
  font-size: 18px;
}

select {
  font-size: 16px;
```

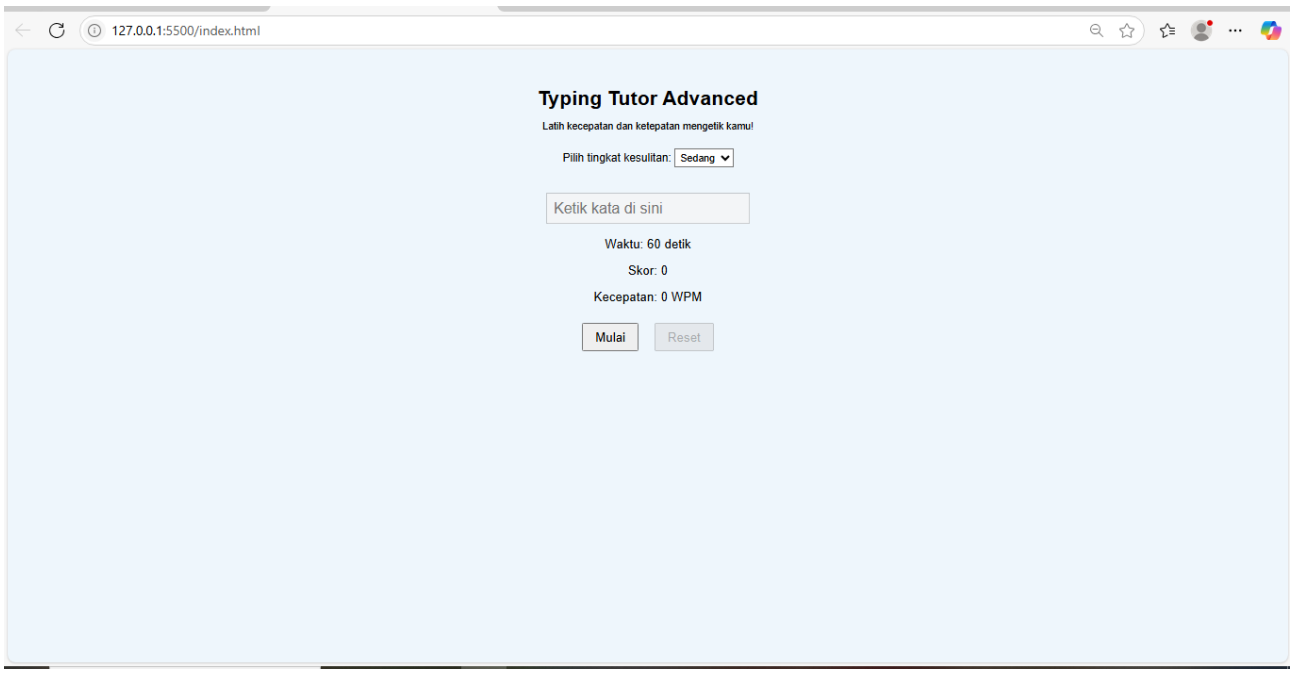
```
  padding: 5px;
  margin: 10px 0 20px 0;
}

.word-display {
  font-size: 36px;
  font-weight: bold;
  margin: 20px 0;
  color: #333;
  letter-spacing: 2px;
}

.word-display .correct {
  color: green;
}

.word-display .incorrect {
  color: red;
  text-decoration: underline;
}

#inputBox {
  font-size: 24px;
  padding: 10px;
  width: 300px;
}

.info {
  margin: 15px 0;
  font-size: 20px;
}

button {
  font-size: 20px;
  padding: 10px 20px;
  cursor: pointer;
  margin: 10px;
}
```

script.js

```
const words = {
  easy: ["cat", "dog", "sun", "tree", "book", "car", "pen", "cup", "fish", "hat"],
  medium: ["javascript", "function", "variable", "object", "array", "programming",
"keyboard", "developer", "browser", "syntax"],
  hard: ["asynchronous", "polymorphism", "encapsulation", "inheritance", "abstraction",
"synchronization", "initialization", "declaration", "optimization", "concurrency"]
};

const wordDisplay = document.getElementById("wordDisplay");
const inputBox = document.getElementById("inputBox");
const timeDisplay = document.getElementById("time");
const scoreDisplay = document.getElementById("score");
const wpmDisplay = document.getElementById("wpm");
const startBtn = document.getElementById("startBtn");
const resetBtn = document.getElementById("resetBtn");
const difficultySelect = document.getElementById("difficulty");

let currentWord = "";
let score = 0;
let time = 60;
let timerInterval;
```

```
let startTime;
let totalTypedWords = 0;

function getRandomWord(difficulty) {
  const list = words[difficulty];
  return list[Math.floor(Math.random() * list.length)];
}

function renderWord(word, typed) {
  // Highlight tiap huruf benar/ salah
  wordDisplay.innerHTML = "";
  for (let i = 0; i < word.length; i++) {
    const span = document.createElement("span");
    span.textContent = word[i];
    if (typed[i] == null) {
      // belum diketik
    } else if (typed[i] === word[i]) {
      span.classList.add("correct");
    } else {
      span.classList.add("incorrect");
    }
    wordDisplay.appendChild(span);
  }
}

function showNewWord() {
  const difficulty = difficultySelect.value;
  currentWord = getRandomWord(difficulty);
  renderWord(currentWord, "");
  inputBox.value = "";
  inputBox.focus();
}

function startGame() {
  score = 0;
  time = 60;
  totalTypedWords = 0;
  scoreDisplay.textContent = score;
  timeDisplay.textContent = time;
  wpmDisplay.textContent = 0;
  inputBox.disabled = false;
  startBtn.disabled = true;
  resetBtn.disabled = false;
  difficultySelect.disabled = true;

  showNewWord();
  startTime = Date.now();

  timerInterval = setInterval(() => {
    const elapsed = Math.floor((Date.now() - startTime) / 1000);
    time = 60 - elapsed;
    timeDisplay.textContent = time;

    if (time <= 0) {
      endGame();
    }
  }, 200);
}

function endGame() {
  clearInterval(timerInterval);
  inputBox.disabled = true;
```

```
  startBtn.disabled = false;
  resetBtn.disabled = true;
  difficultySelect.disabled = false;
  wordDisplay.textContent = `Waktu habis! Skor: ${score}, Kecepatan: ${calculateWPM()}
WPM`;
}

function calculateWPM() {
  // WPM = (kata benar / menit)
  const minutes = (60 - time) / 60;
  if (minutes === 0) return 0;
  return Math.round(score / minutes);
}

inputBox.addEventListener("input", () => {
  const typed = inputBox.value;
  renderWord(currentWord, typed);

  if (typed.toLowerCase() === currentWord.toLowerCase()) {
    score++;
    totalTypedWords++;
    scoreDisplay.textContent = score;
    wpmDisplay.textContent = calculateWPM();
    showNewWord();
  }
});

resetBtn.addEventListener("click", () => {
  clearInterval(timerInterval);
  inputBox.disabled = true;
  startBtn.disabled = false;
  resetBtn.disabled = true;
  difficultySelect.disabled = false;
  timeDisplay.textContent = 60;
  scoreDisplay.textContent = 0;
  wpmDisplay.textContent = 0;
  wordDisplay.textContent = "";
  inputBox.value = "";
});

startBtn.addEventListener("click", startGame);
```



2. Game Tebak Angka

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Typing Tutor Advanced</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <h1>Typing Tutor Advanced</h1>
  <p>Latih kecepatan dan ketepatan mengetik kamu!</p>

  <label for="difficulty">Pilih tingkat kesulitan:</label>
  <select id="difficulty">
    <option value="easy">Mudah</option>
    <option value="medium" selected>Sedang</option>
    <option value="hard">Susah</option>
  </select>

  <div id="wordDisplay" class="word-display"></div>
  <input type="text" id="inputBox" placeholder="Ketik kata di sini" autocomplete="off"
disabled />

  <div class="info">
    <p>Waktu: <span id="time">60</span> detik</p>
    <p>Skor: <span id="score">0</span></p>
    <p>Kecepatan: <span id="wpm">0</span> WPM</p>
  </div>

  <button id="startBtn">Mulai</button>
  <button id="resetBtn" disabled>Reset</button>

  <script src="script.js"></script>
</body>
</html>
```

style.css

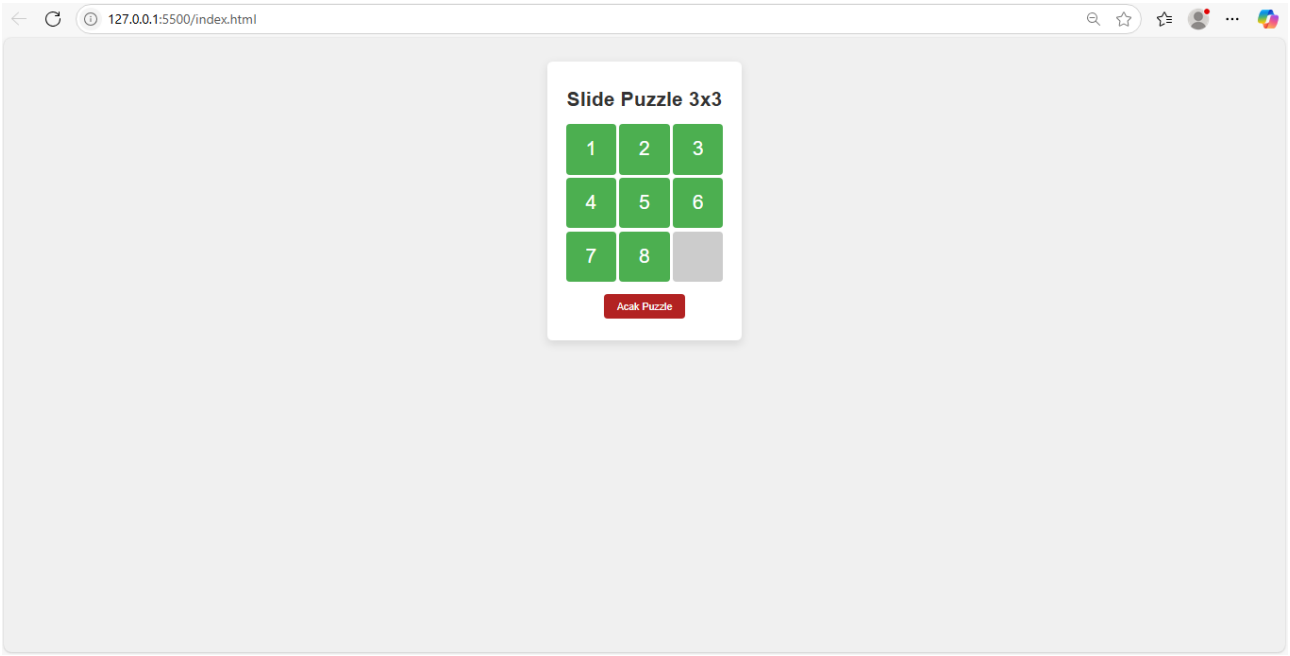
```
body {
  font-family: Arial, sans-serif;
  text-align: center;
  padding: 30px;
  background-color: #eef6fc;
}

h1 {
  margin-bottom: 10px;
}

label {
  font-size: 18px;
}

select {
  font-size: 16px;
  padding: 5px;
  margin: 10px 0 20px 0;
}

.word-display {
  font-size: 36px;
  font-weight: bold;
  margin: 20px 0;
  color: #333;
  letter-spacing: 2px;
}

.word-display .correct {
  color: green;
}

.word-display .incorrect {
  color: red;
  text-decoration: underline;
}

#inputBox {
  font-size: 24px;
  padding: 10px;
  width: 300px;
}

.info {
  margin: 15px 0;
  font-size: 20px;
}

button {
  font-size: 20px;
  padding: 10px 20px;
  cursor: pointer;
  margin: 10px;
}
```

script.js

```
const words = {
  easy: ["cat", "dog", "sun", "tree", "book", "car", "pen", "cup", "fish", "hat"],
```

```
  medium: ["javascript", "function", "variable", "object", "array", "programming",
"keyboard", "developer", "browser", "syntax"],
  hard: ["asynchronous", "polymorphism", "encapsulation", "inheritance", "abstraction",
"synchronization", "initialization", "declaration", "optimization", "concurrency"]
};

const wordDisplay = document.getElementById("wordDisplay");
const inputBox = document.getElementById("inputBox");
const timeDisplay = document.getElementById("time");
const scoreDisplay = document.getElementById("score");
const wpmDisplay = document.getElementById("wpm");
const startBtn = document.getElementById("startBtn");
const resetBtn = document.getElementById("resetBtn");
const difficultySelect = document.getElementById("difficulty");

let currentWord = "";
let score = 0;
let time = 60;
let timerInterval;
let startTime;
let totalTypedWords = 0;

function getRandomWord(difficulty) {
  const list = words[difficulty];
  return list[Math.floor(Math.random() * list.length)];
}

function renderWord(word, typed) {
  // Highlight tiap huruf benar/ salah
  wordDisplay.innerHTML = "";
  for (let i = 0; i < word.length; i++) {
    const span = document.createElement("span");
    span.textContent = word[i];
    if (typed[i] == null) {
      // belum diketik
    } else if (typed[i] === word[i]) {
      span.classList.add("correct");
    } else {
      span.classList.add("incorrect");
    }
    wordDisplay.appendChild(span);
  }
}

function showNewWord() {
  const difficulty = difficultySelect.value;
  currentWord = getRandomWord(difficulty);
  renderWord(currentWord, "");
  inputBox.value = "";
  inputBox.focus();
}

function startGame() {
  score = 0;
  time = 60;
  totalTypedWords = 0;
  scoreDisplay.textContent = score;
  timeDisplay.textContent = time;
  wpmDisplay.textContent = 0;
  inputBox.disabled = false;
  startBtn.disabled = true;
  resetBtn.disabled = false;
```

```
  difficultySelect.disabled = true;

  showNewWord();
  startTime = Date.now();

  timerInterval = setInterval(() => {
    const elapsed = Math.floor((Date.now() - startTime) / 1000);
    time = 60 - elapsed;
    timeDisplay.textContent = time;

    if (time <= 0) {
      endGame();
    }
  }, 200);
}

function endGame() {
  clearInterval(timerInterval);
  inputBox.disabled = true;
  startBtn.disabled = false;
  resetBtn.disabled = true;
  difficultySelect.disabled = false;
  wordDisplay.textContent = `Waktu habis! Skor: ${score}, Kecepatan: ${calculateWPM()}
WPM`;
}

function calculateWPM() {
  // WPM = (kata benar / menit)
  const minutes = (60 - time) / 60;
  if (minutes === 0) return 0;
  return Math.round(score / minutes);
}

inputBox.addEventListener("input", () => {
  const typed = inputBox.value;
  renderWord(currentWord, typed);

  if (typed.toLowerCase() === currentWord.toLowerCase()) {
    score++;
    totalTypedWords++;
    scoreDisplay.textContent = score;
    wpmDisplay.textContent = calculateWPM();
    showNewWord();
  }
});

resetBtn.addEventListener("click", () => {
  clearInterval(timerInterval);
  inputBox.disabled = true;
  startBtn.disabled = false;
  resetBtn.disabled = true;
  difficultySelect.disabled = false;
  timeDisplay.textContent = 60;
  scoreDisplay.textContent = 0;
  wpmDisplay.textContent = 0;
  wordDisplay.textContent = "";
  inputBox.value = "";
});

startBtn.addEventListener("click", startGame);
```



3. Game Pazzel

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Slide Puzzle 3x3</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <div class="container">
    <h1>Slide Puzzle 3x3</h1>
    <div id="puzzle"></div>
    <button id="shuffle-btn">Acak Puzzle</button>
    <div id="message"></div>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

style.css

```
body {
    font-family: Arial, sans-serif;
    background: #f0f0f0;
    display: flex;
    justify-content: center;
    padding: 30px;
  }

  .container {
    text-align: center;
    background: white;
    padding: 20px 30px;
    border-radius: 8px;
```

```
  box-shadow: 0 5px 15px rgba(0,0,0,0.1);
  max-width: 320px;
}

h1 {
  margin-bottom: 20px;
  color: #333;
}

#puzzle {
  display: grid;
  grid-template-columns: repeat(3, 80px);
  grid-template-rows: repeat(3, 80px);
  gap: 5px;
  margin-bottom: 20px;
}

.tile {
  background: #4caf50;
  color: white;
  font-size: 2em;
  display: flex;
  justify-content: center;
  align-items: center;
  border-radius: 6px;
  cursor: pointer;
  user-select: none;
  transition: background 0.3s;
}

.tile.empty {
  background: #ccc;
  cursor: default;
}

.tile:hover:not(.empty) {
  background: #45a049;
}

#shuffle-btn {
  padding: 10px 20px;
  background: #d62828;
  border: none;
  color: white;
  border-radius: 6px;
  cursor: pointer;
  font-size: 1em;
}

#shuffle-btn:hover {
  background: #b22222;
}

#message {
  margin-top: 15px;
  font-weight: bold;
  color: #d62828;
}
```

script.js

```
const puzzleEl = document.getElementById("puzzle");
const shuffleBtn = document.getElementById("shuffle-btn");
const messageEl = document.getElementById("message");

let tiles = [];
let emptyIndex = 8; // posisi kotak kosong awal

function initPuzzle() {
  tiles = [...Array(8).keys()].map(i => i + 1);
  tiles.push(null);
  emptyIndex = 8;
  render();
  messageEl.textContent = "";
}

function render() {
  puzzleEl.innerHTML = "";
  tiles.forEach((num, idx) => {
    const tile = document.createElement("div");
    tile.classList.add("tile");
    if(num === null) {
      tile.classList.add("empty");
      tile.textContent = "";
    } else {
      tile.textContent = num;
      tile.addEventListener("click", () => moveTile(idx));
    }
    puzzleEl.appendChild(tile);
  });
}

function moveTile(idx) {
  if(canMove(idx)) {
    tiles[emptyIndex] = tiles[idx];
    tiles[idx] = null;
    emptyIndex = idx;
    render();
    if(checkWin()) {
      messageEl.textContent = "Selamat! Kamu berhasil menyusun puzzle.";
    }
  }
}

function canMove(idx) {
  const emptyRow = Math.floor(emptyIndex / 3);
  const emptyCol = emptyIndex % 3;
  const idxRow = Math.floor(idx / 3);
  const idxCol = idx % 3;

  const rowDiff = Math.abs(emptyRow - idxRow);
  const colDiff = Math.abs(emptyCol - idxCol);

  return (rowDiff + colDiff === 1);
}

function shuffle() {
  for(let i=0; i<1000; i++) {
    const neighbors = getNeighbors(emptyIndex);
    const rand = neighbors[Math.floor(Math.random() * neighbors.length)];
    tiles[emptyIndex] = tiles[rand];
    tiles[rand] = null;
    emptyIndex = rand;
```

```
  }
  render();
  messageEl.textContent = "";
}

function getNeighbors(idx) {
  const row = Math.floor(idx / 3);
  const col = idx % 3;
  const neighbors = [];
  if(row > 0) neighbors.push(idx - 3);
  if(row < 2) neighbors.push(idx + 3);
  if(col > 0) neighbors.push(idx - 1);
  if(col < 2) neighbors.push(idx + 1);
  return neighbors;
}

function checkWin() {
  for(let i=0; i<8; i++) {
    if(tiles[i] !== i+1) return false;
  }
  return true;
}

shuffleBtn.addEventListener("click", shuffle);

initPuzzle();
```