TUBES PENGOLAHAN CITRA DIGITAL

April 2025

ANGGOTA

GET TO KNOW US BETTER

Hafidz Zaenul Ahkam

231511012

Muhammad Reivan Naufal Mufid

231511021

Yazid Fauzan Prasatria

231511032

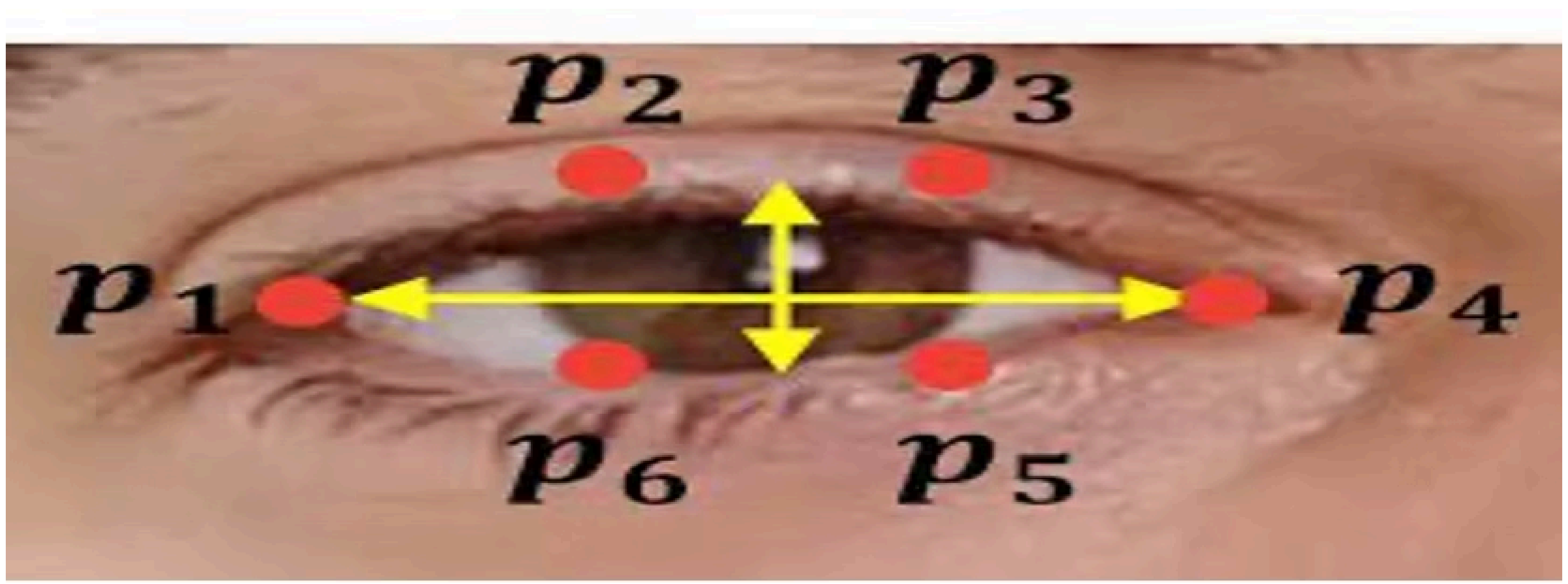
DESKRIPSI TUBES

DESKRIPSI

Your Last Gaze adalah game horror berbasis mediapipe face mesh dengan interaksi utama menutup dan membuka mata. Menampilkan scene kamar karakter utama yang mempunyai banyak objek. Objek – objek tersebut akan bergetar secara acak, menandai bahwa jumpscare akan terjadi. Player menghindari jumpscare dengan menutup mata, saat player menutup mata game akan menjalankan audio setelah audio tersebut selesai player dapat membuka kembali matanya. Jika player tidak menutup mata dengan cepat maka game akan menampilkan jumpscare dan game over.

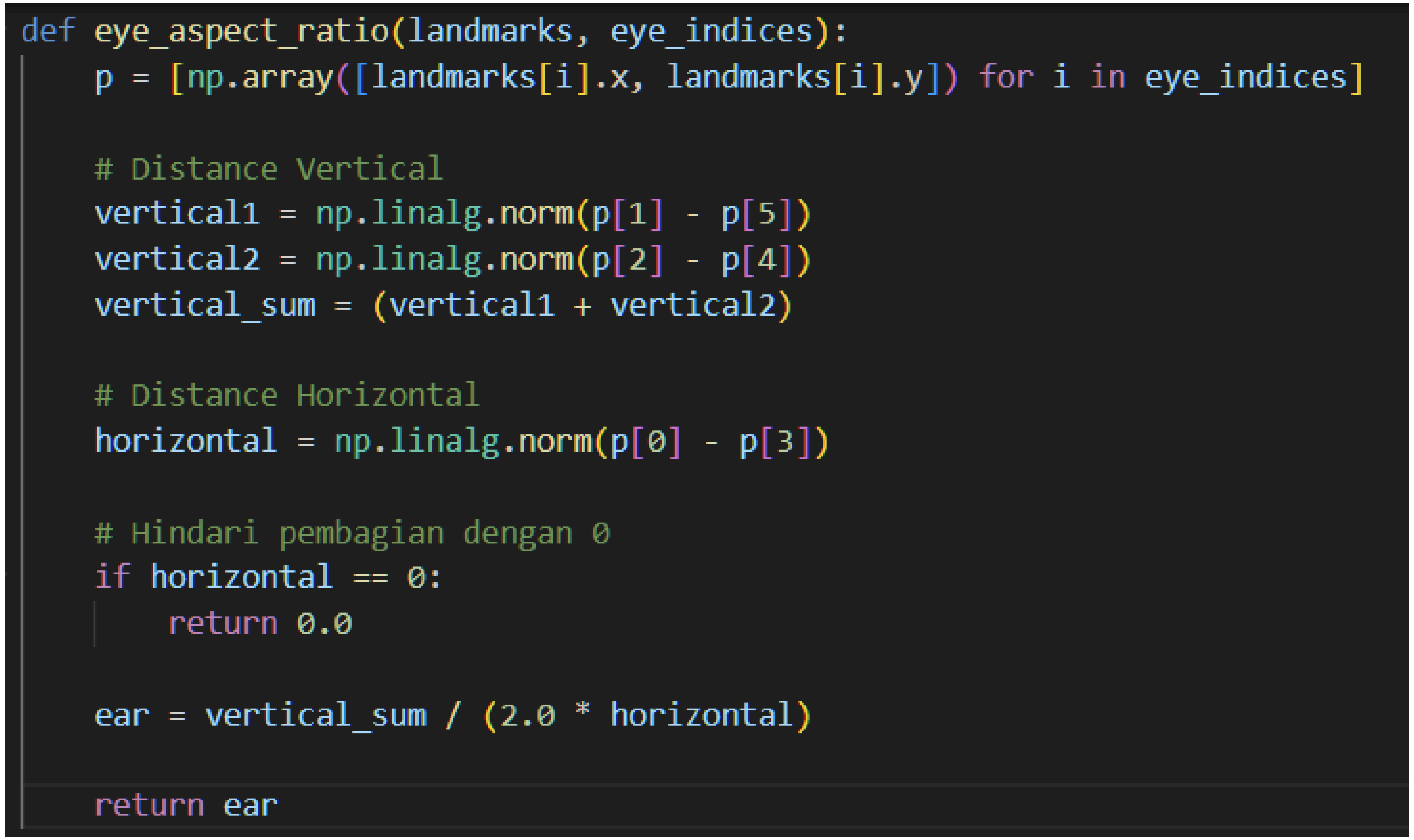
BACKEND

PERHITUNGAN EYE ASPECT RATIO



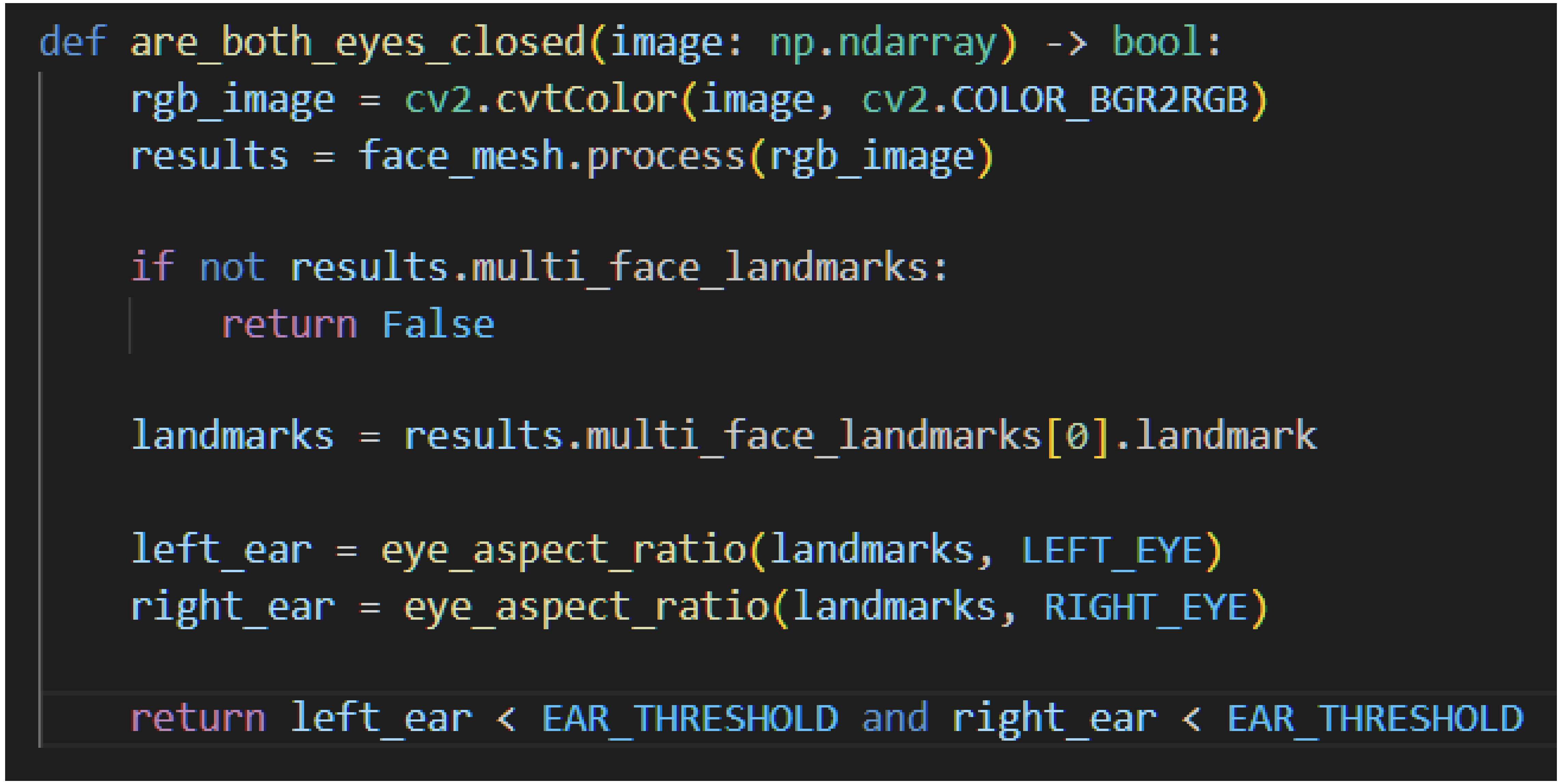
$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

PERHITUNGAN EYE ASPECT RATIO



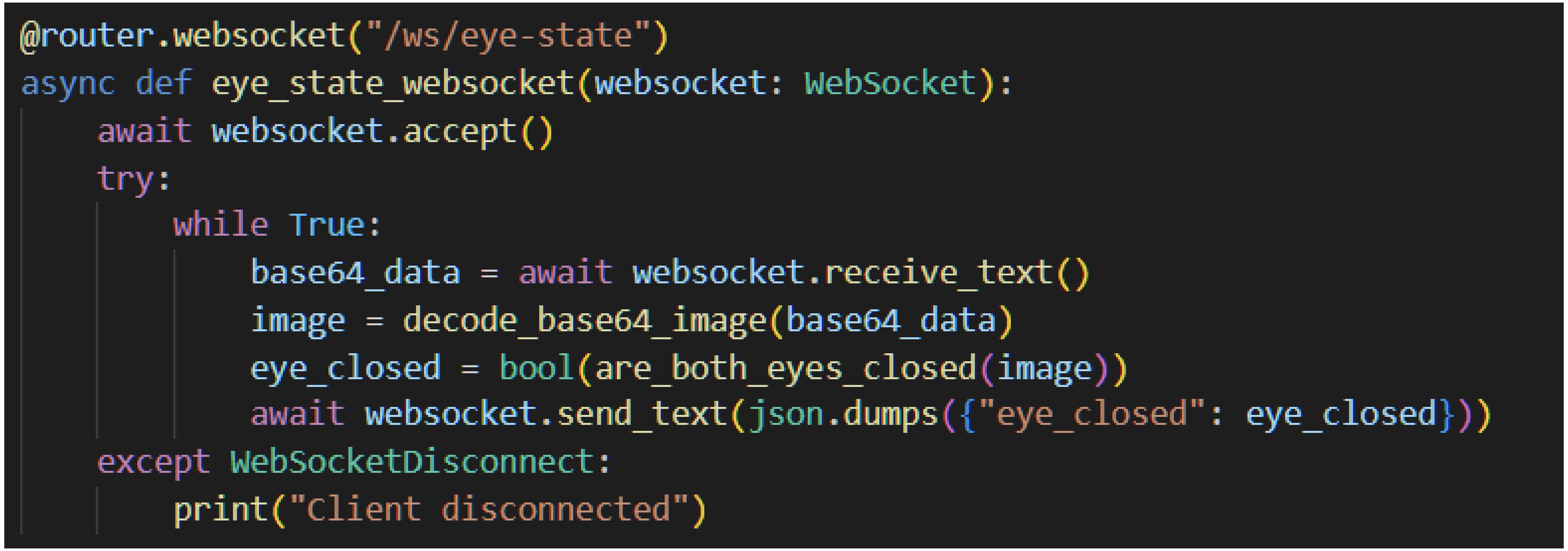
```python
def eye_aspect_ratio(landmarks, eye_indices):
    p = [np.array([landmarks[i].x, landmarks[i].y]) for i in eye_indices]

    # Distance Vertical
    vertical1 = np.linalg.norm(p[1] - p[5])
    vertical2 = np.linalg.norm(p[2] - p[4])
    vertical_sum = (vertical1 + vertical2)

    # Distance Horizontal
    horizontal = np.linalg.norm(p[0] - p[3])

    # Hindari pembagian dengan 0
    if horizontal == 0:
        return 0.0

    ear = vertical_sum / (2.0 * horizontal)

    return ear
```

DETEKSI TUTUP MATA



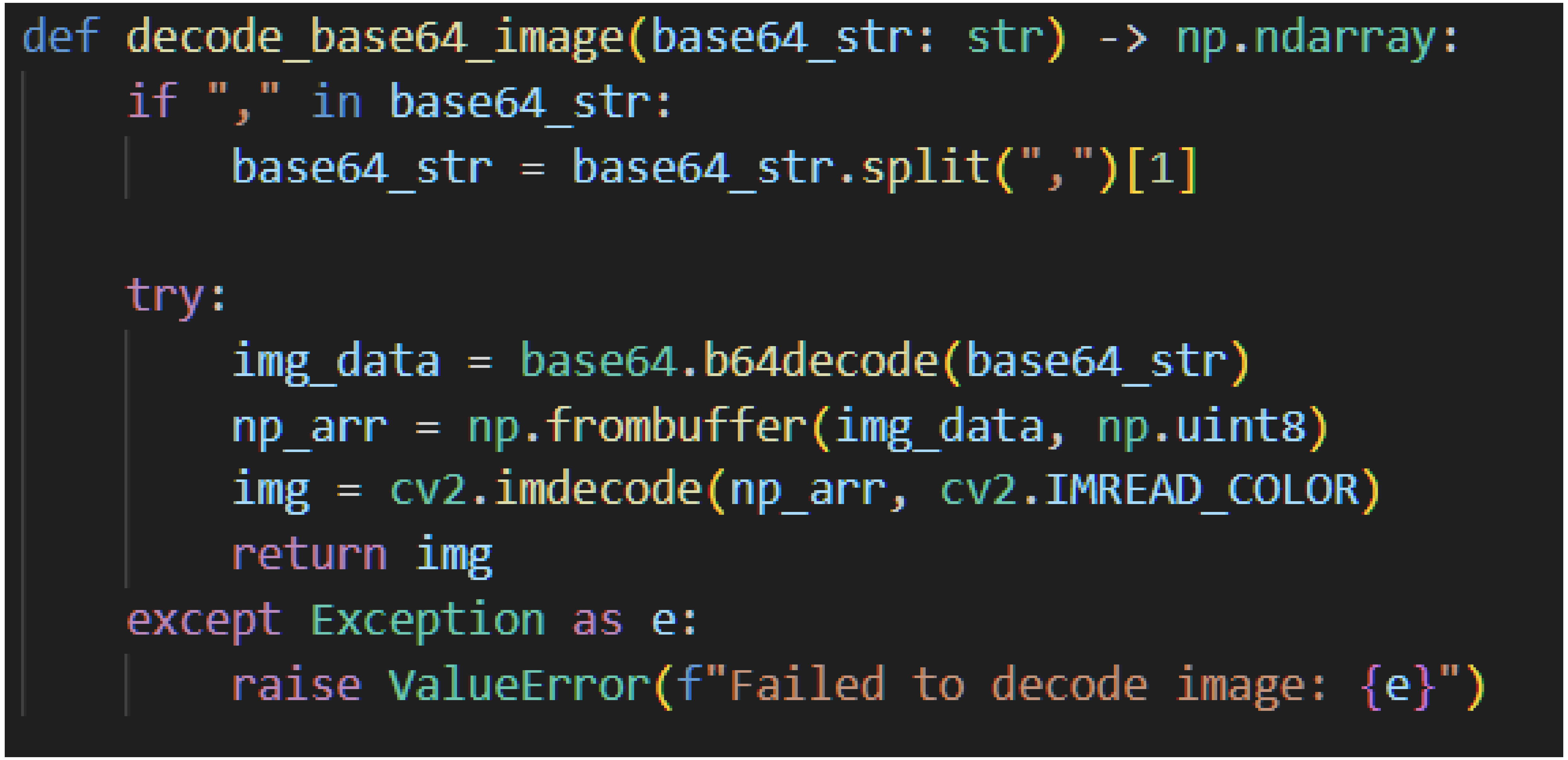
```python
def are_both_eyes_closed(image: np.ndarray) -> bool:
    rgb_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    results = face_mesh.process(rgb_image)

    if not results.multi_face_landmarks:
        return False

    landmarks = results.multi_face_landmarks[0].landmark

    left_ear = eye_aspect_ratio(landmarks, LEFT_EYE)
    right_ear = eye_aspect_ratio(landmarks, RIGHT_EYE)

    return left_ear < EAR_THRESHOLD and right_ear < EAR_THRESHOLD
```

WEBSOCKET



```python
@router.websocket("/ws/eye-state")
async def eye_state_websocket(websocket: WebSocket):
    await websocket.accept()
    try:
        while True:
            base64_data = await websocket.receive_text()
            image = decode_base64_image(base64_data)
            eye_closed = bool(are_both_eyes_closed(image))
            await websocket.send_text(json.dumps({"eye_closed": eye_closed}))
    except WebSocketDisconnect:
        print("Client disconnected")
```

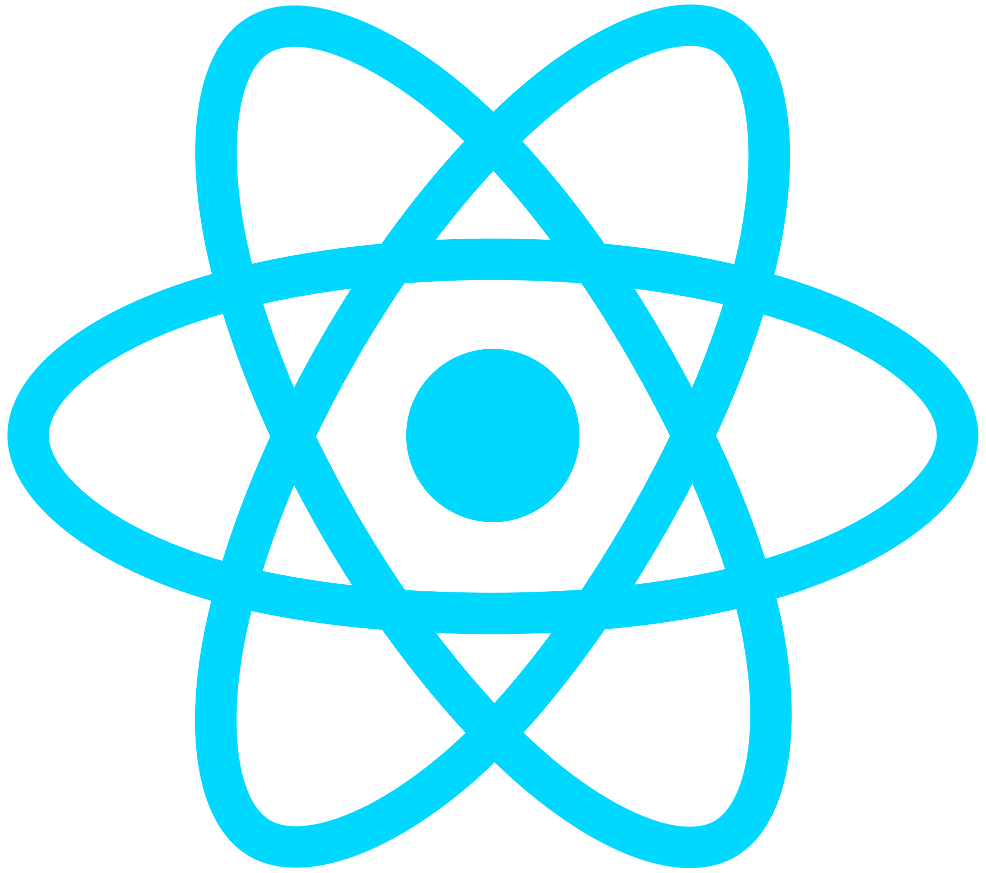
DECODE GAMBAR



```python
def decode_base64_image(base64_str: str) -> np.ndarray:
    if "," in base64_str:
        base64_str = base64_str.split(",")[1]

    try:
        img_data = base64.b64decode(base64_str)
        np_arr = np.frombuffer(img_data, np.uint8)
        img = cv2.imdecode(np_arr, cv2.IMREAD_COLOR)
        return img
    except Exception as e:
        raise ValueError(f"Failed to decode image: {e}")
```

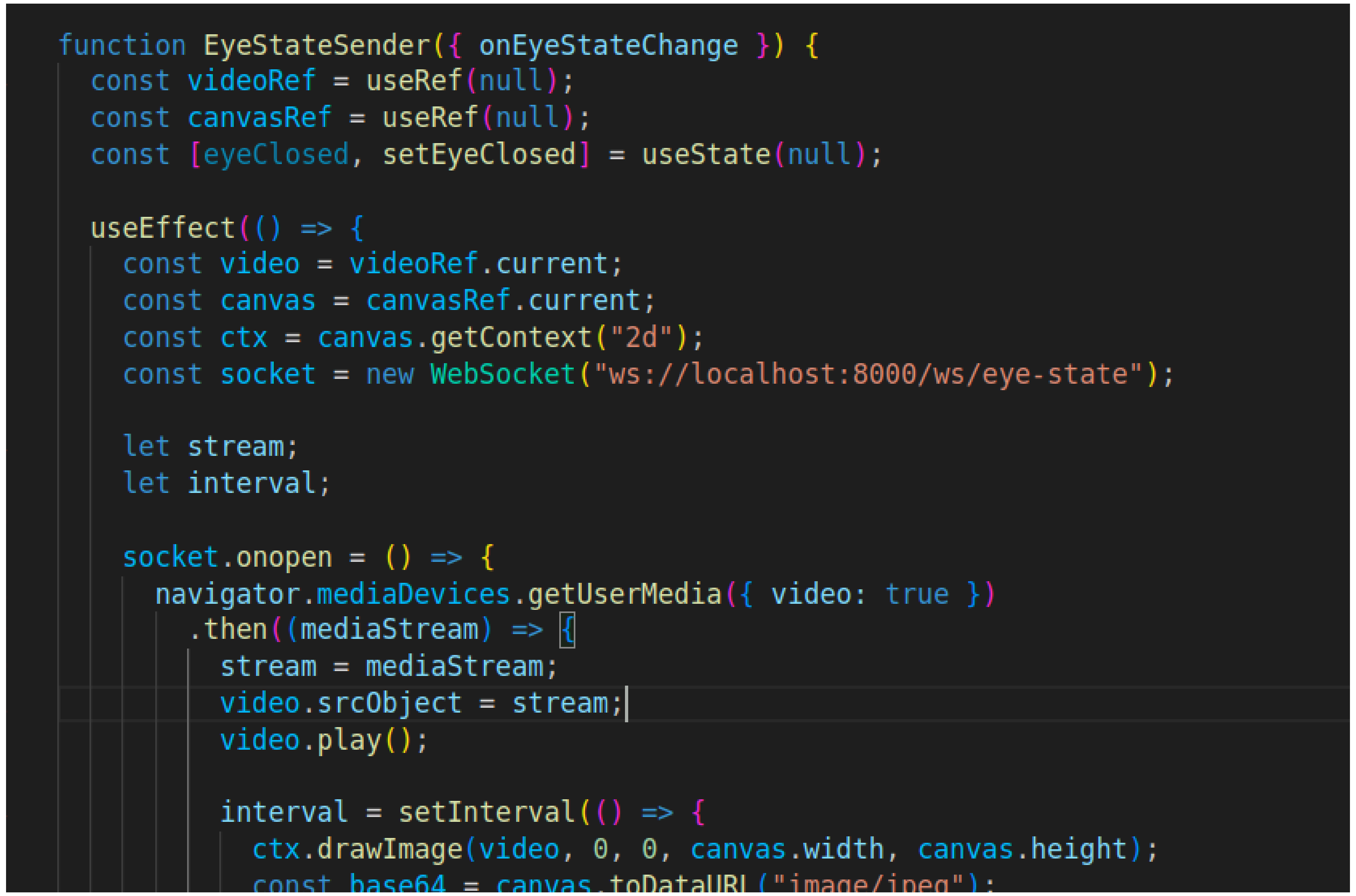
FRONTEND

TOOLS



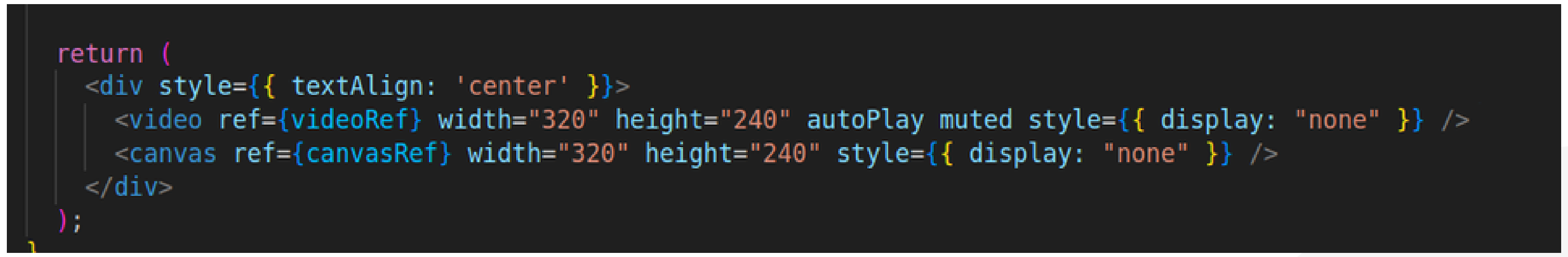
KOMPONEN

Penggunaan WebSocket dari frontend



```
function EyeStateSender({ onEyeStateChange }) {
  const videoRef = useRef(null);
  const canvasRef = useRef(null);
  const [eyeClosed, setEyeClosed] = useState(null);

  useEffect(() => {
    const video = videoRef.current;
    const canvas = canvasRef.current;
    const ctx = canvas.getContext("2d");
    const socket = new WebSocket("ws://localhost:8000/ws/eye-state");

    let stream;
    let interval;

    socket.onopen = () => {
      navigator.mediaDevices.getUserMedia({ video: true })
        .then((mediaStream) => {
          stream = mediaStream;
          video.srcObject = stream;
          video.play();

          interval = setInterval(() => {
            ctx.drawImage(video, 0, 0, canvas.width, canvas.height);
            const base64 = canvas.toDataURL("image/jpeg");
```

Pengambilan API Backend

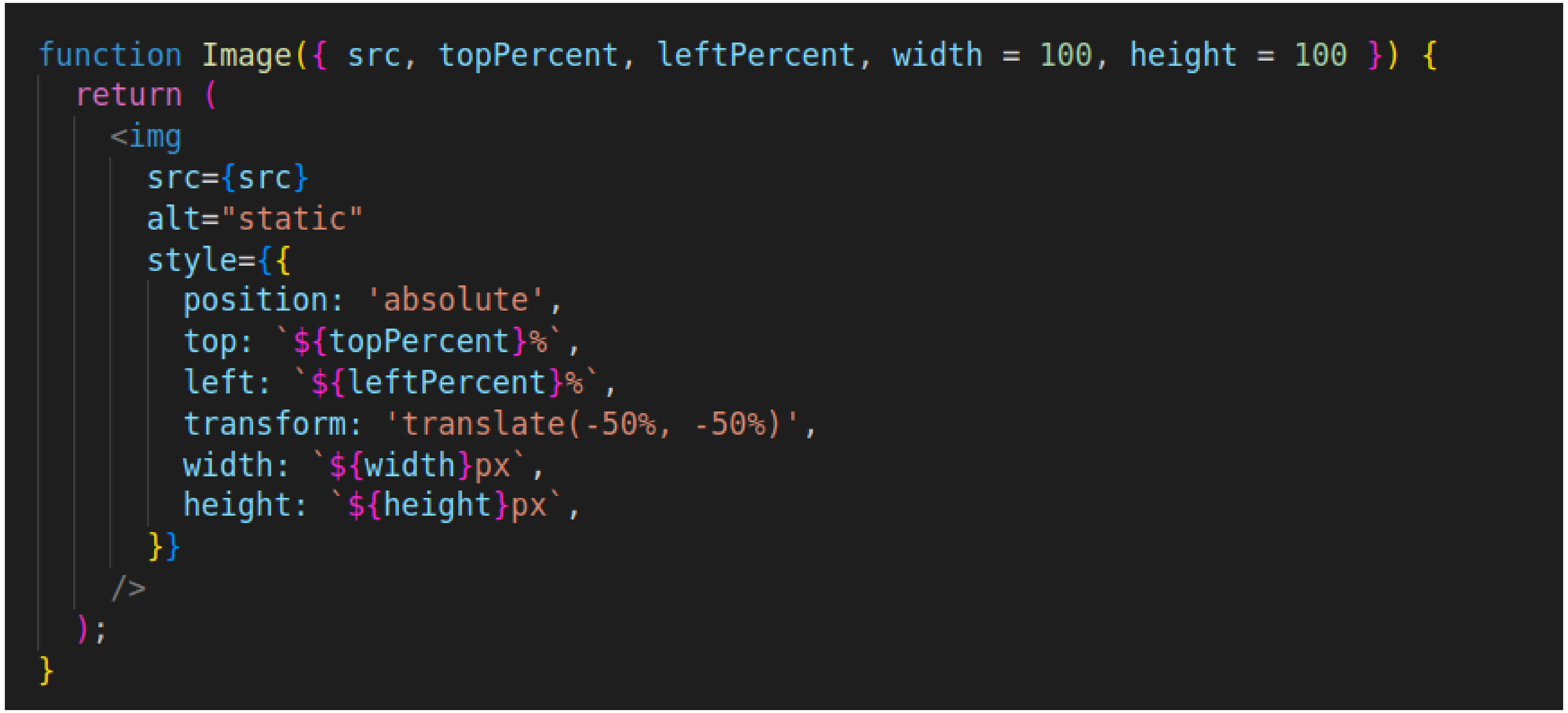


```
  return (
    <div style={{ textAlign: 'center' }}>
      <video ref={videoRef} width="320" height="240" autoPlay muted style={{ display: "none" }} />
      <canvas ref={canvasRef} width="320" height="240" style={{ display: "none" }} />
    </div>
  );
}
```

Pengambilan Frame Video

KOMPONEN

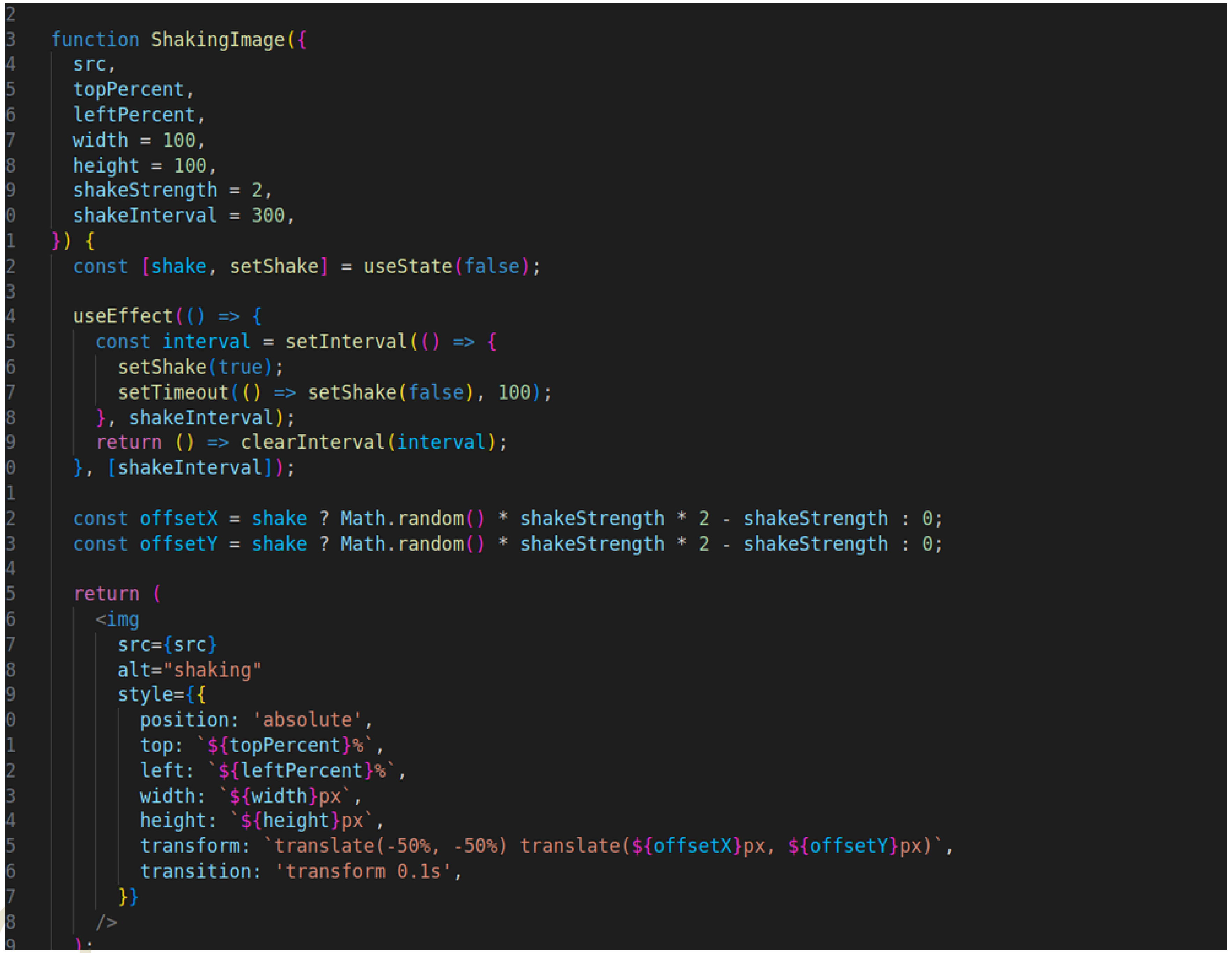
Gambar diam



```
function Image({ src, topPercent, leftPercent, width = 100, height = 100 }) {
  return (
    <img
      src={src}
      alt="static"
      style={{
        position: 'absolute',
        top: `${topPercent}%`,
        left: `${leftPercent}%`,
        transform: 'translate(-50%, -50%)',
        width: `${width}px`,
        height: `${height}px`,
      }}
    />
  );
}
```

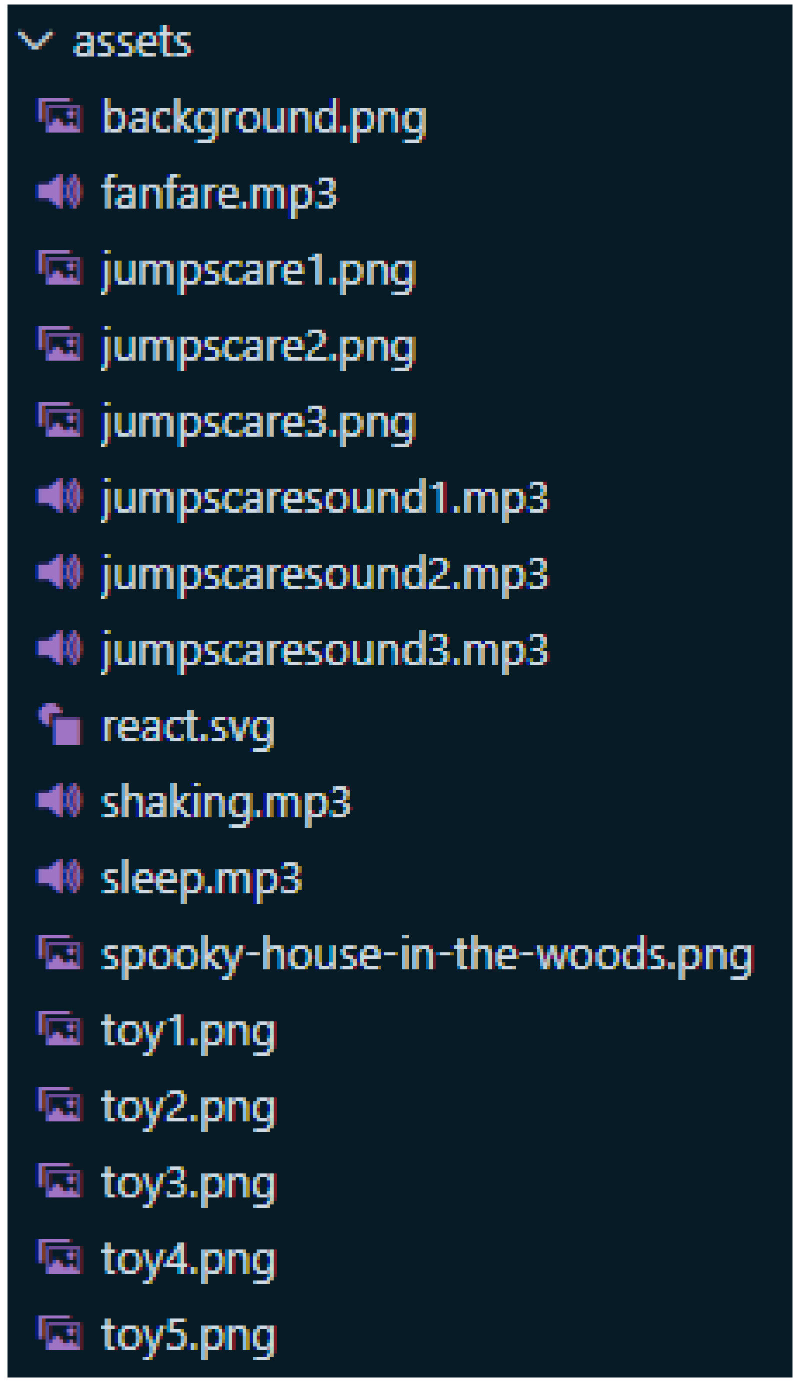
KOMPONEN

Gambar bergetar



```
function ShakingImage({
  src,
  topPercent,
  leftPercent,
  width = 100,
  height = 100,
  shakeStrength = 2,
  shakeInterval = 300,
}) {
  const [shake, setShake] = useState(false);

  useEffect(() => {
    const interval = setInterval(() => {
      setShake(true);
      setTimeout(() => setShake(false), 100);
    }, shakeInterval);
    return () => clearInterval(interval);
  }, [shakeInterval]);

  const offsetX = shake ? Math.random() * shakeStrength * 2 - shakeStrength : 0;
  const offsetY = shake ? Math.random() * shakeStrength * 2 - shakeStrength : 0;

  return (
    <img
      src={src}
      alt="shaking"
      style={{
        position: 'absolute',
        top: `${topPercent}%`,
        left: `${leftPercent}%`,
        width: `${width}px`,
        height: `${height}px`,
        transform: `translate(-50%, -50%) translate(${offsetX}px, ${offsetY}px)`,
        transition: 'transform 0.1s',
      }}
    />
  );
```

ASSETS



```
assets
  background.png
  fanfare.mp3
  jumpscare1.png
  jumpscare2.png
  jumpscare3.png
  jumpscaresound1.mp3
  jumpscaresound2.mp3
  jumpscaresound3.mp3
  react.svg
  shaking.mp3
  sleep.mp3
  spooky-house-in-the-woods.png
  toy1.png
  toy2.png
  toy3.png
  toy4.png
  toy5.png
```

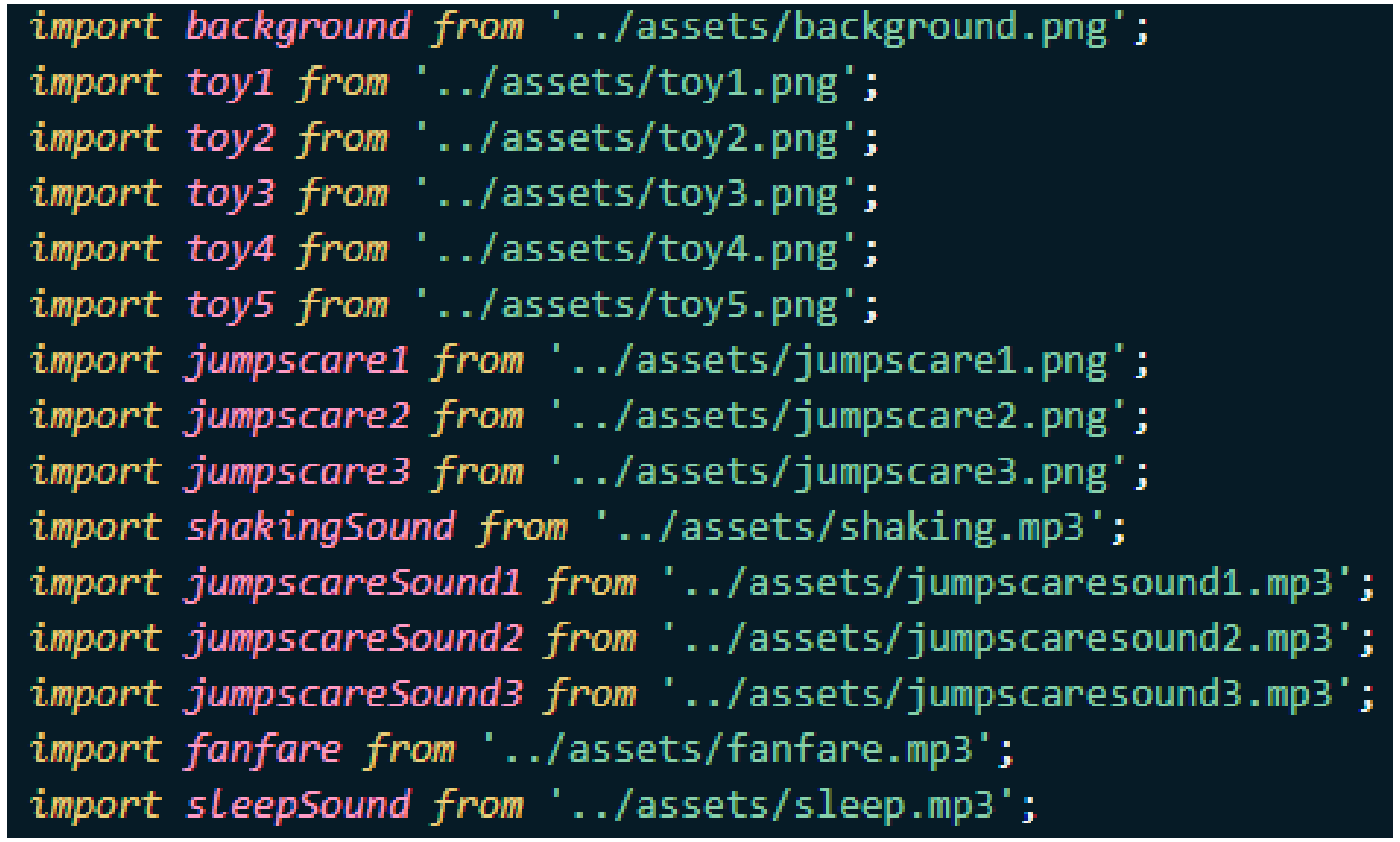
GAME

Alur permainan

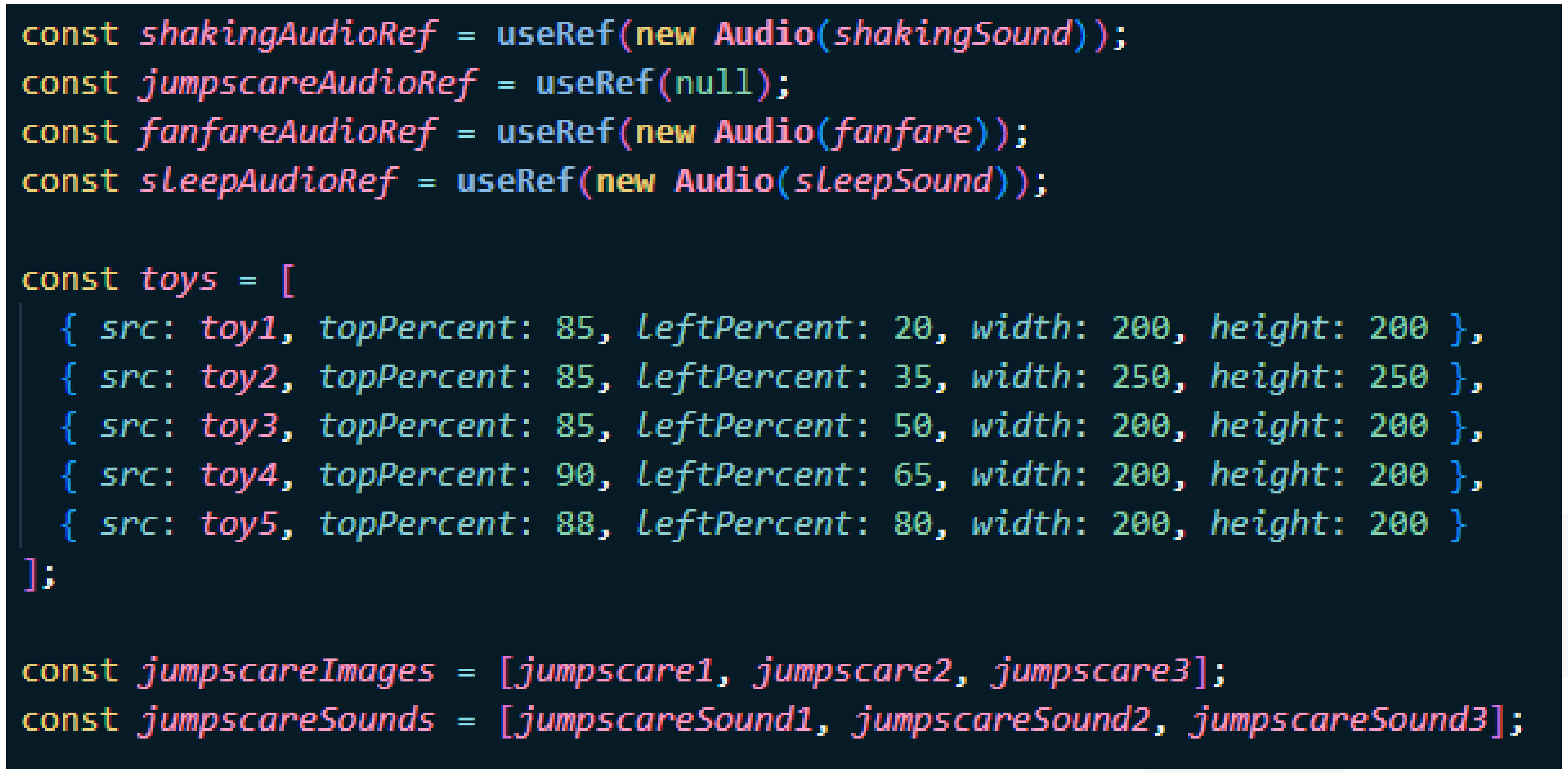
1. Ketika memulai game, pemain akan berada di kamar dan terdapat objek di kamar tersebut
2. Setiap 10 - 30 detik, objek akan bergetar selama 5 detik. Objek yang bergetar dipilih secara acak
3. Ketika objek bergetar, pemain menutup mata sampai objek berhenti bergetar, maka permainan akan berlanjut. Alur 2 dan 3 akan terus diulang.
4. Jika selama objek bergetar pemain masih membuka mata selama 1,5 detik, akan muncul jumpscare selama 2 detik dan langsung game over
5. Jika pemain berhasil bertahan selama 2 menit, maka dinyatakan victory

GAME

Pemanggilan assets



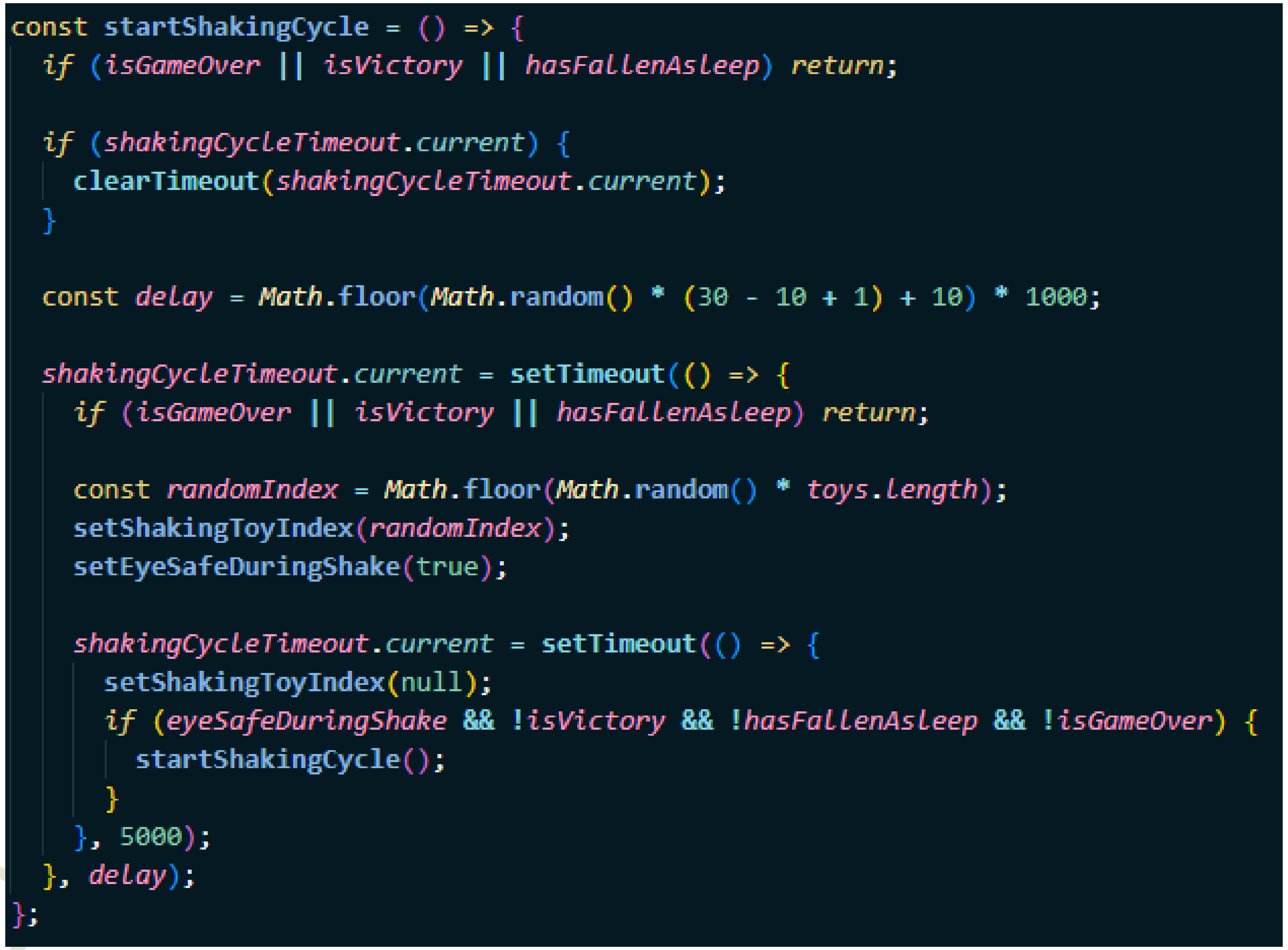
```
import background from '../assets/background.png';
import toy1 from '../assets/toy1.png';
import toy2 from '../assets/toy2.png';
import toy3 from '../assets/toy3.png';
import toy4 from '../assets/toy4.png';
import toy5 from '../assets/toy5.png';
import jumpscare1 from '../assets/jumpscare1.png';
import jumpscare2 from '../assets/jumpscare2.png';
import jumpscare3 from '../assets/jumpscare3.png';
import shakingSound from '../assets/shaking.mp3';
import jumpscareSound1 from '../assets/jumpscaresound1.mp3';
import jumpscareSound2 from '../assets/jumpscaresound2.mp3';
import jumpscareSound3 from '../assets/jumpscaresound3.mp3';
import fanfare from '../assets/fanfare.mp3';
import sleepSound from '../assets/sleep.mp3';
```



```
const shakingAudioRef = useRef(new Audio(shakingSound));
const jumpscareAudioRef = useRef(null);
const fanfareAudioRef = useRef(new Audio(fanfare));
const sleepAudioRef = useRef(new Audio(sleepSound));

const toys = [
  { src: toy1, topPercent: 85, leftPercent: 20, width: 200, height: 200 },
  { src: toy2, topPercent: 85, leftPercent: 35, width: 250, height: 250 },
  { src: toy3, topPercent: 85, leftPercent: 50, width: 200, height: 200 },
  { src: toy4, topPercent: 90, leftPercent: 65, width: 200, height: 200 },
  { src: toy5, topPercent: 88, leftPercent: 80, width: 200, height: 200 }
];

const jumpscareImages = [jumpscare1, jumpscare2, jumpscare3];
const jumpscareSounds = [jumpscareSound1, jumpscareSound2, jumpscareSound3];
```

GAME

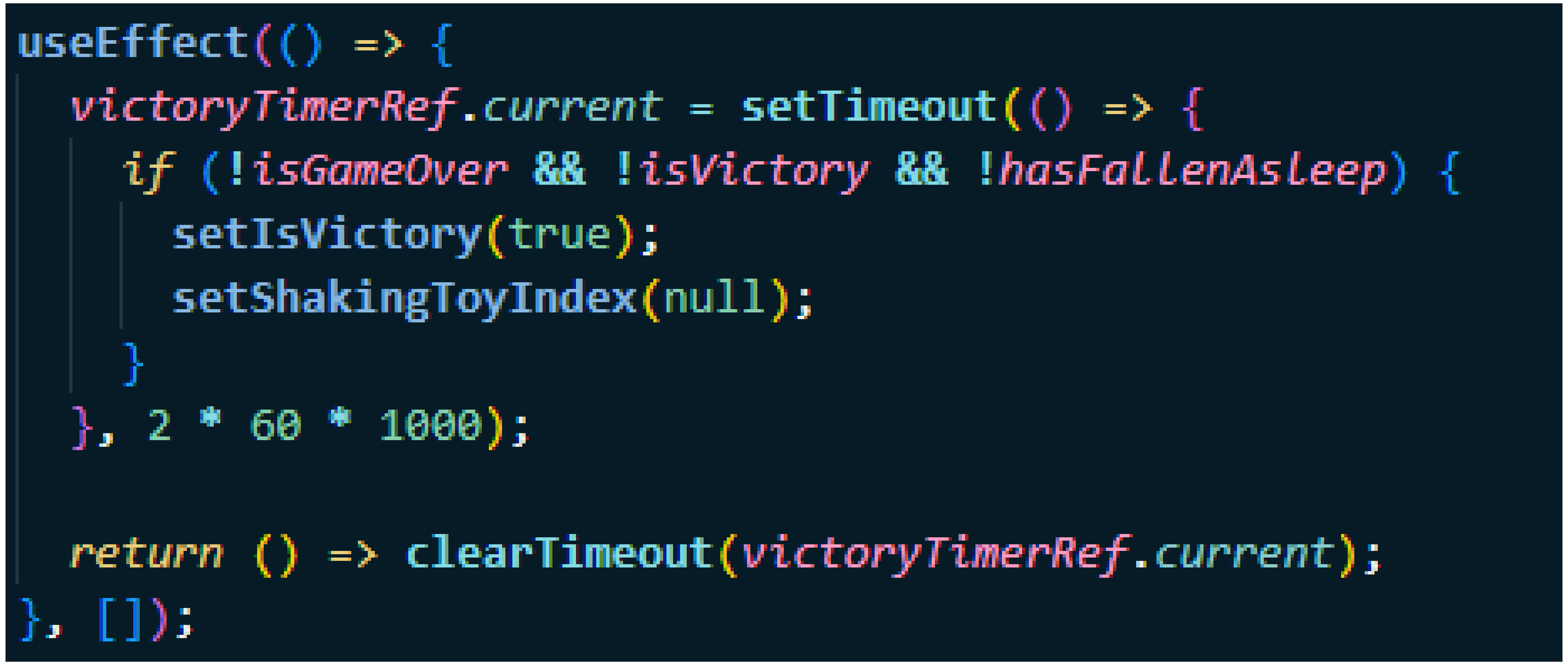
Logic durasi objek bergetar dan kapan objek bergetar



```
const startShakingCycle = () => {
  if (isGameOver || isVictory || hasFallenAsleep) return;

  if (shakingCycleTimeout.current) {
    clearTimeout(shakingCycleTimeout.current);
  }

  const delay = Math.floor(Math.random() * (30 - 10 + 1) + 10) * 1000;

  shakingCycleTimeout.current = setTimeout(() => {
    if (isGameOver || isVictory || hasFallenAsleep) return;

    const randomIndex = Math.floor(Math.random() * toys.length);
    setShakingToyIndex(randomIndex);
    setEyeSafeDuringShake(true);

    shakingCycleTimeout.current = setTimeout(() => {
      setShakingToyIndex(null);
      if (eyeSafeDuringShake && !isVictory && !hasFallenAsleep && !isGameOver) {
        startShakingCycle();
      }
    }, 5000);
  }, delay);
};
```

GAME

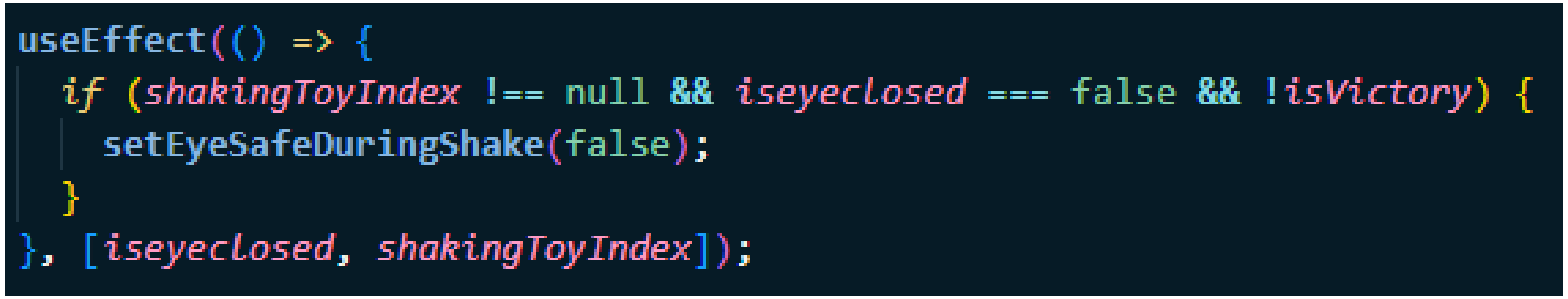
Logic kondisi victory



```
useEffect(() => {
  victoryTimerRef.current = setTimeout(() => {
    if (!isGameOver && !isVictory && !hasFallenAsleep) {
      setIsVictory(true);
      setShakingToyIndex(null);
    }
  }, 2 * 60 * 1000);

  return () => clearTimeout(victoryTimerRef.current);
}, []);
```

GAME

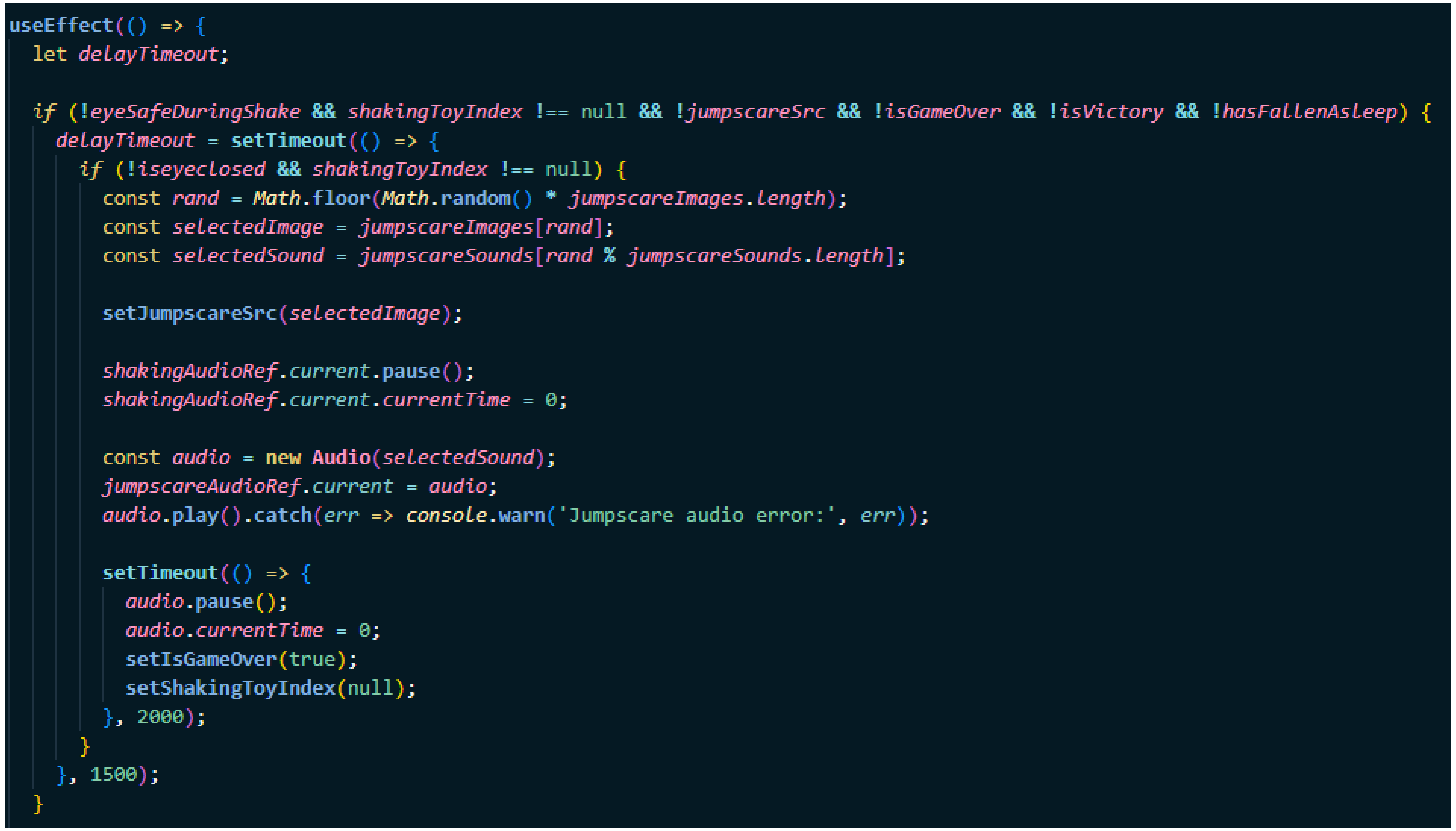
Logic game berjalan kembali ketika pemain tutup mata



```
useEffect(() => {
  if (shakingToyIndex !== null && iseyeclosed === false && !isVictory) {
    setEyeSafeDuringShake(false);
  }
}, [iseyeclosed, shakingToyIndex]);
```

GAME

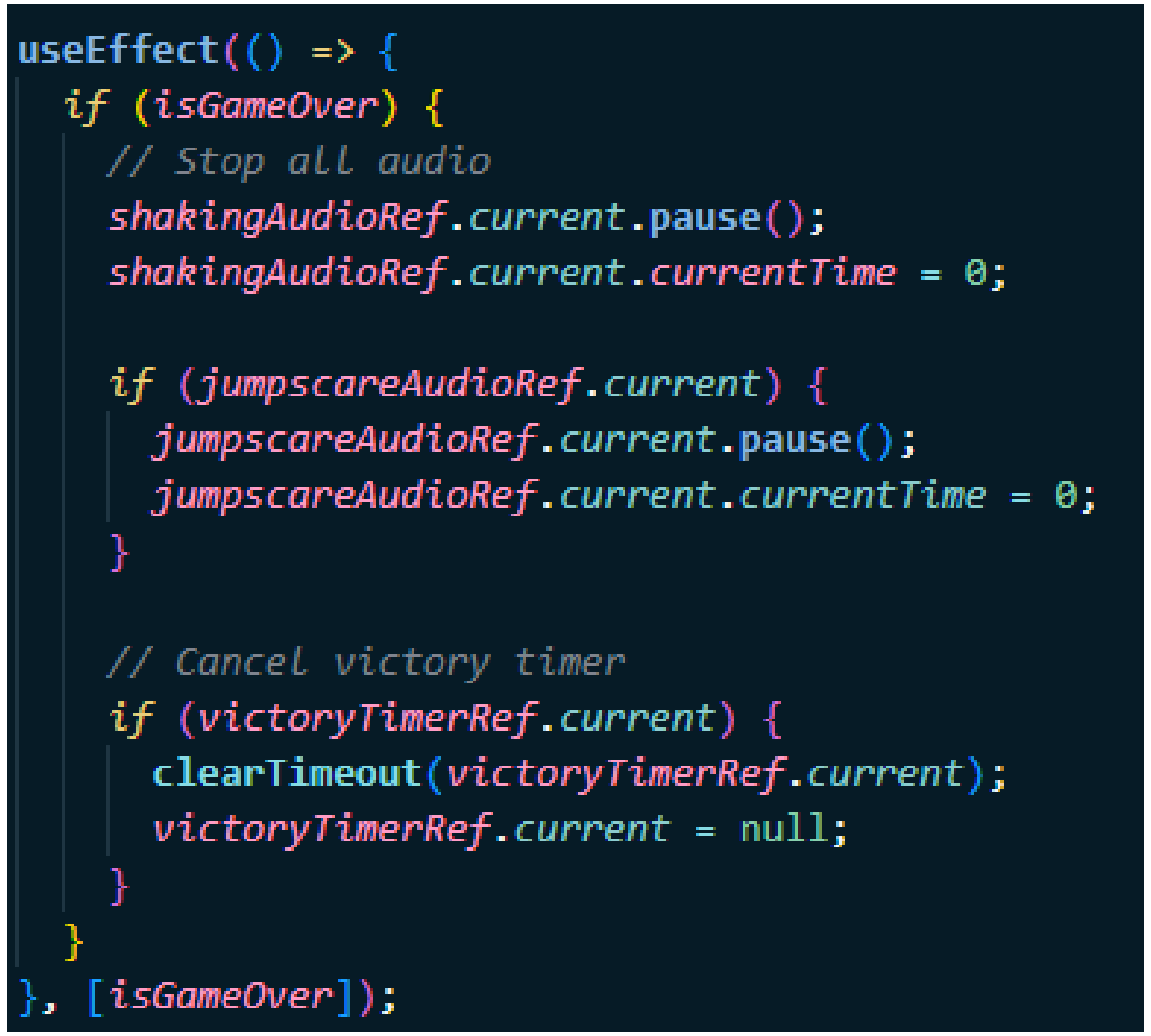
Logic jumpscare dan audio jumpscare



```
useEffect(() => {
  let delayTimeout;

  if (!eyeSafeDuringShake && shakingToyIndex !== null && !jumpscareSrc && !isGameOver && !isVictory && !hasFallenAsleep) {
    delayTimeout = setTimeout(() => {
      if (!iseyeclosed && shakingToyIndex !== null) {
        const rand = Math.floor(Math.random() * jumpscareImages.length);
        const selectedImage = jumpscareImages[rand];
        const selectedSound = jumpscareSounds[rand % jumpscareSounds.length];

        setJumpscareSrc(selectedImage);

        shakingAudioRef.current.pause();
        shakingAudioRef.current.currentTime = 0;

        const audio = new Audio(selectedSound);
        jumpscareAudioRef.current = audio;
        audio.play().catch(err => console.warn('Jumpscare audio error:', err));

        setTimeout(() => {
          audio.pause();
          audio.currentTime = 0;
          setIsGameOver(true);
          setShakingToyIndex(null);
        }, 2000);
      }
    }, 1500);
  }
```

GAME

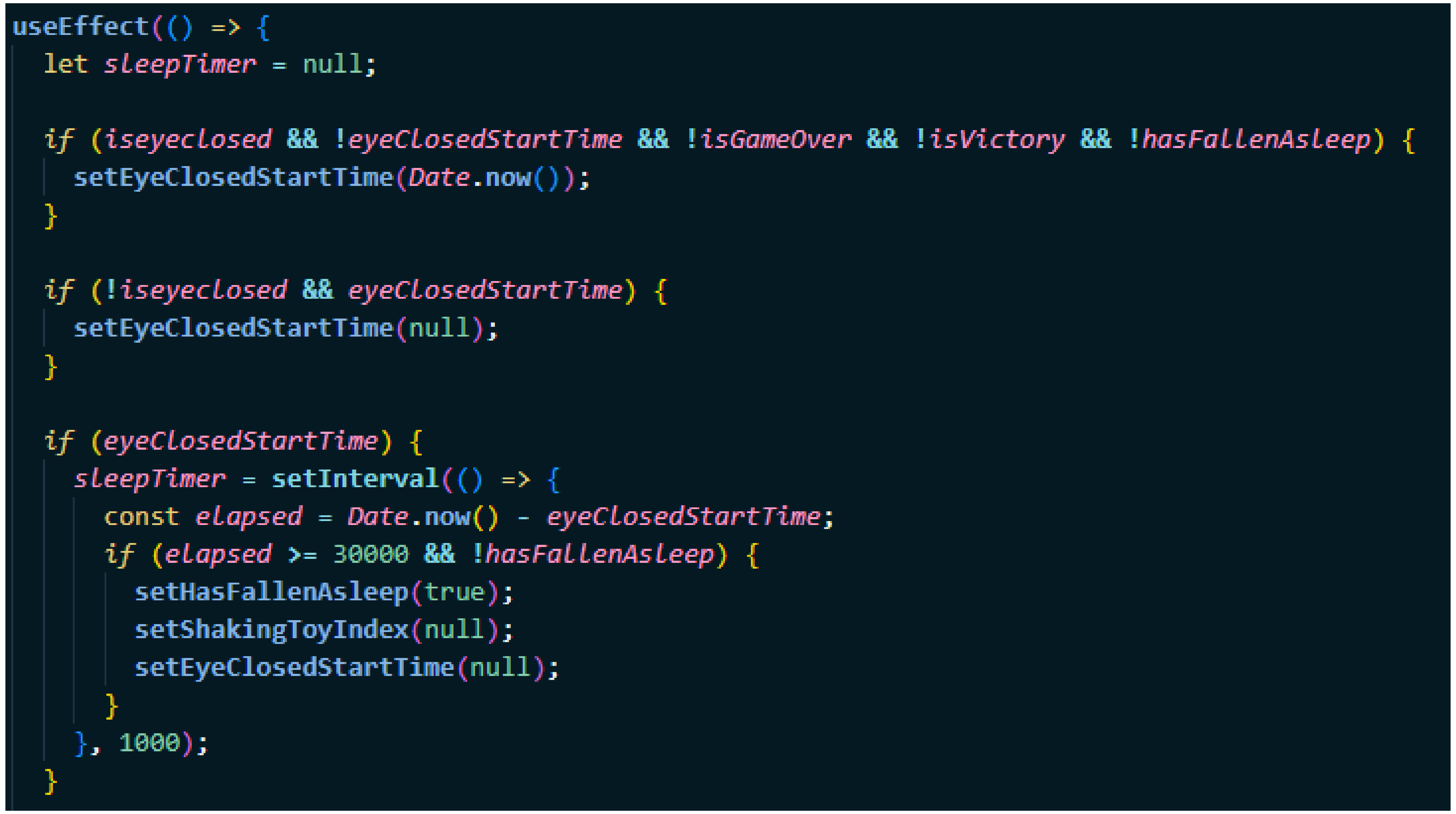
Logic game over



```
useEffect(() => {
  if (isGameOver) {
    // Stop all audio
    shakingAudioRef.current.pause();
    shakingAudioRef.current.currentTime = 0;

    if (jumpscareAudioRef.current) {
      jumpscareAudioRef.current.pause();
      jumpscareAudioRef.current.currentTime = 0;
    }

    // Cancel victory timer
    if (victoryTimerRef.current) {
      clearTimeout(victoryTimerRef.current);
      victoryTimerRef.current = null;
    }
  }
}, [isGameOver]);
```

GAME

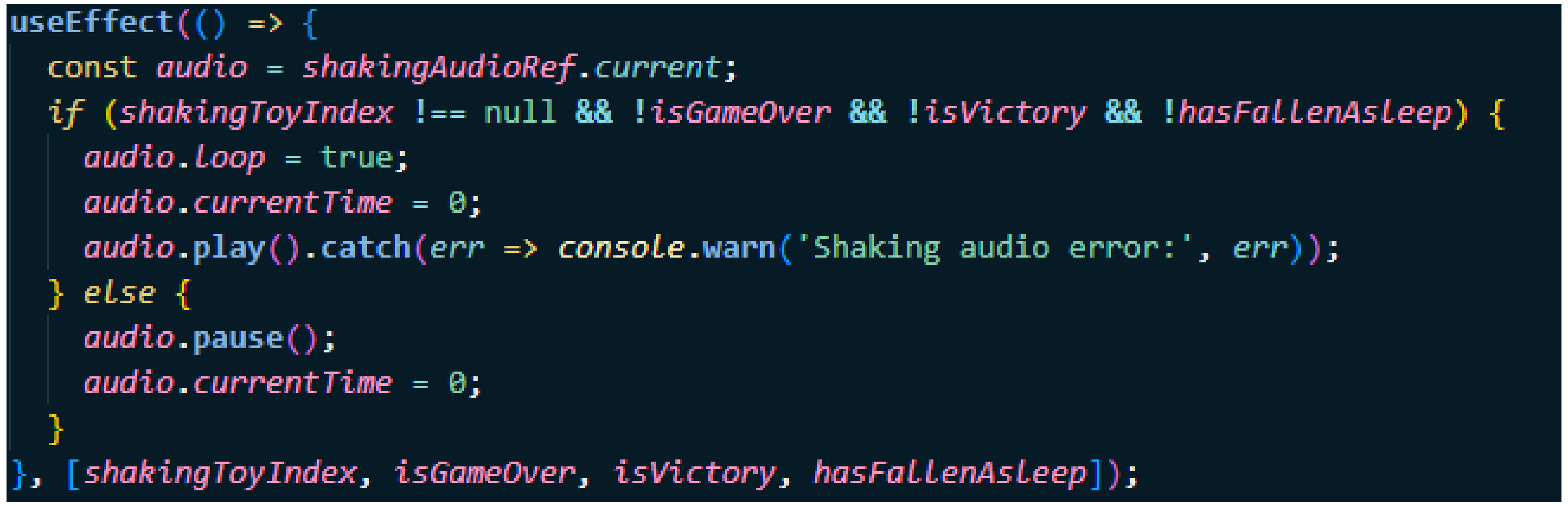
Logic easter egg ketiduran



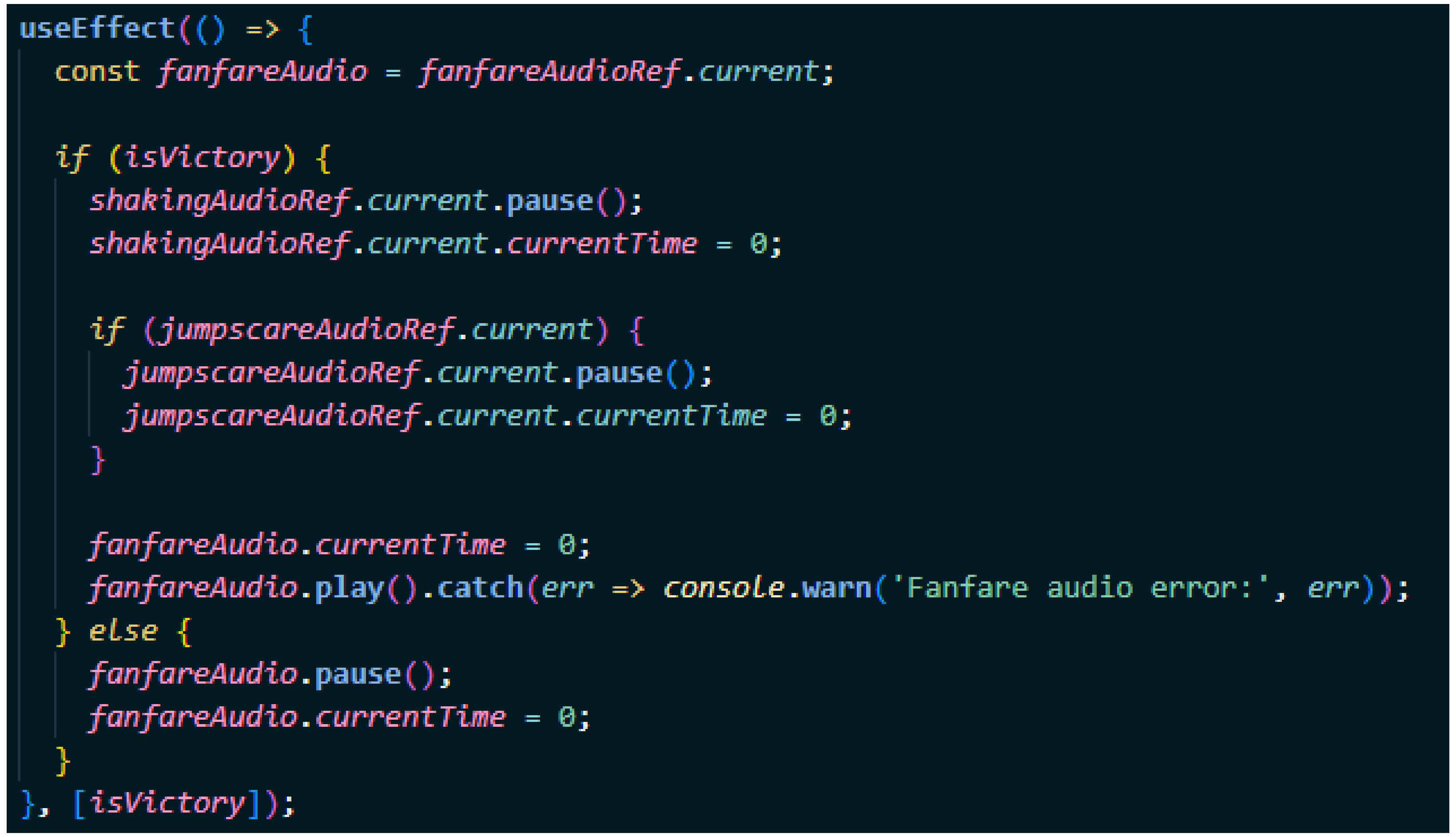
```
useEffect(() => {
  let sleepTimer = null;

  if (iseyeclosed && !eyeClosedStartTime && !isGameOver && !isVictory && !hasFallenAsleep) {
    setEyeClosedStartTime(Date.now());
  }

  if (!iseyeclosed && eyeClosedStartTime) {
    setEyeClosedStartTime(null);
  }

  if (eyeClosedStartTime) {
    sleepTimer = setInterval(() => {
      const elapsed = Date.now() - eyeClosedStartTime;
      if (elapsed >= 30000 && !hasFallenAsleep) {
        setHasFallenAsleep(true);
        setShakingToyIndex(null);
        setEyeClosedStartTime(null);
      }
    }, 1000);
  }
```

GAME

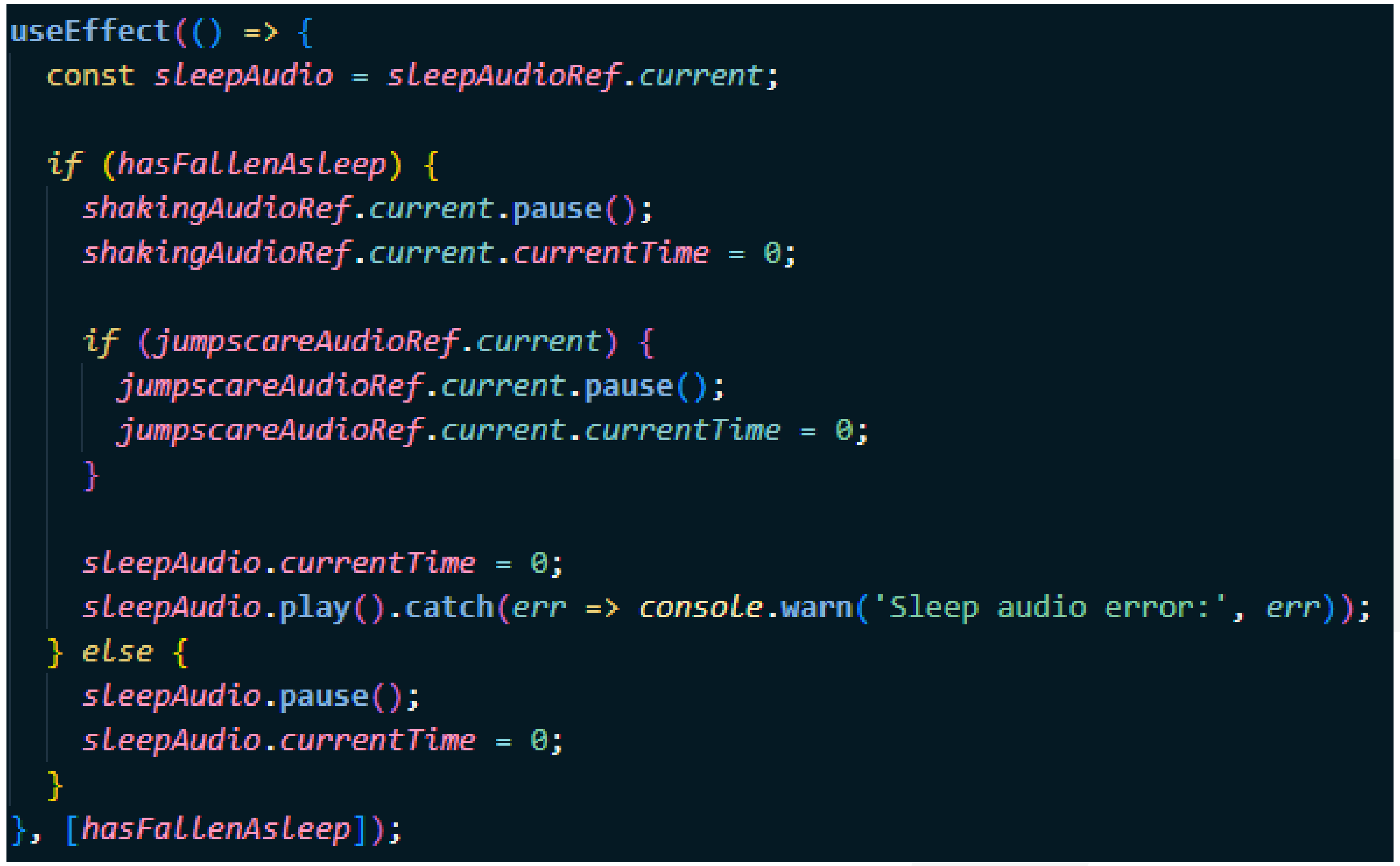
Logic audio (objek bergetar, victory, dan easter egg)



```
useEffect(() => {
  const audio = shakingAudioRef.current;
  if (shakingToyIndex !== null && !isGameOver && !isVictory && !hasFallenAsleep) {
    audio.loop = true;
    audio.currentTime = 0;
    audio.play().catch(err => console.warn('Shaking audio error:', err));
  } else {
    audio.pause();
    audio.currentTime = 0;
  }
}, [shakingToyIndex, isGameOver, isVictory, hasFallenAsleep]);
```



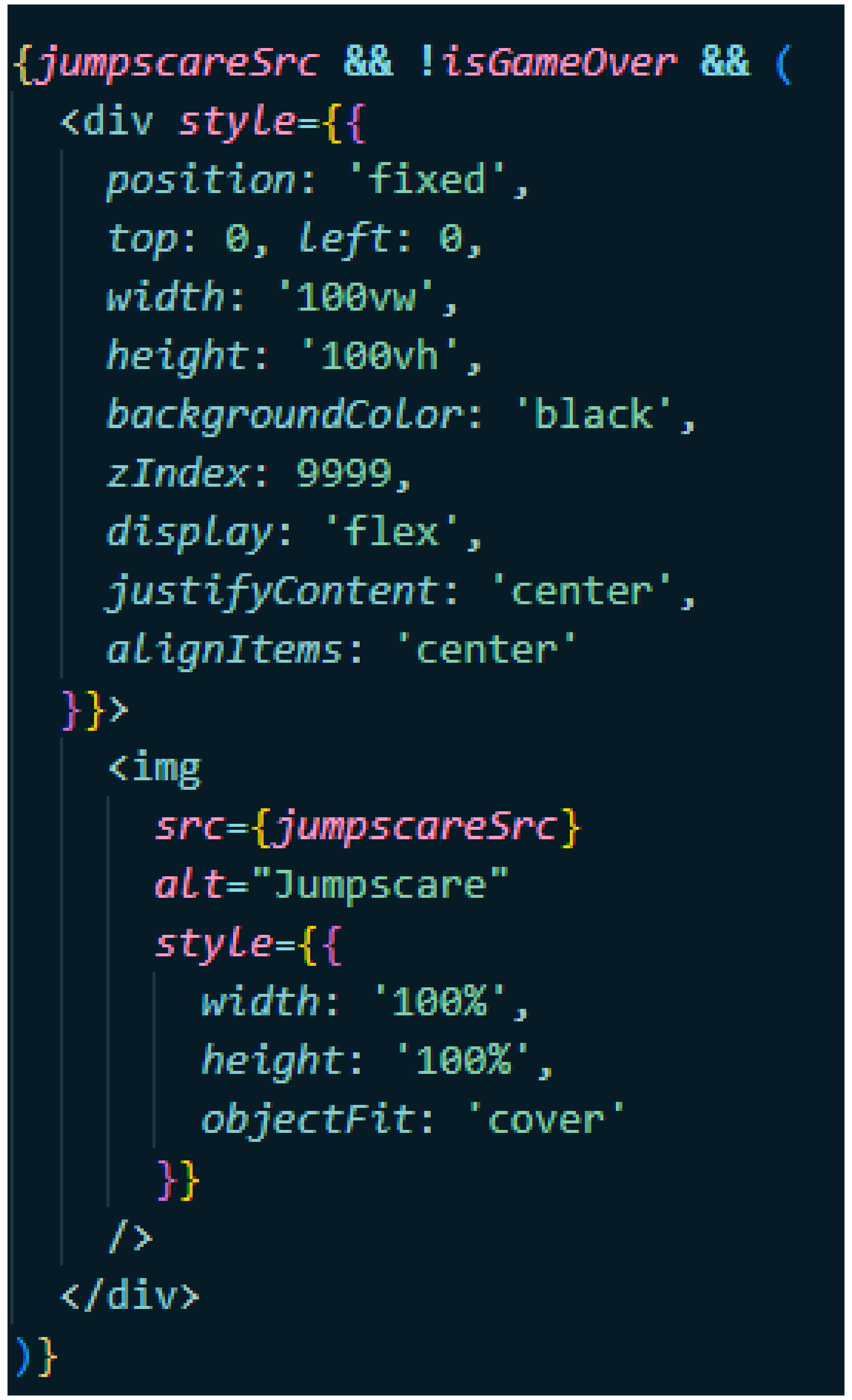
```
useEffect(() => {
  const fanfareAudio = fanfareAudioRef.current;

  if (isVictory) {
    shakingAudioRef.current.pause();
    shakingAudioRef.current.currentTime = 0;

    if (jumpscareAudioRef.current) {
      jumpscareAudioRef.current.pause();
      jumpscareAudioRef.current.currentTime = 0;
    }

    fanfareAudio.currentTime = 0;
    fanfareAudio.play().catch(err => console.warn('Fanfare audio error:', err));
  } else {
    fanfareAudio.pause();
    fanfareAudio.currentTime = 0;
  }
}, [isVictory]);
```



```
useEffect(() => {
  const sleepAudio = sleepAudioRef.current;

  if (hasFallenAsleep) {
    shakingAudioRef.current.pause();
    shakingAudioRef.current.currentTime = 0;

    if (jumpscareAudioRef.current) {
      jumpscareAudioRef.current.pause();
      jumpscareAudioRef.current.currentTime = 0;
    }

    sleepAudio.currentTime = 0;
    sleepAudio.play().catch(err => console.warn('Sleep audio error:', err));
  } else {
    sleepAudio.pause();
    sleepAudio.currentTime = 0;
  }
}, [hasFallenAsleep]);
```

GAME

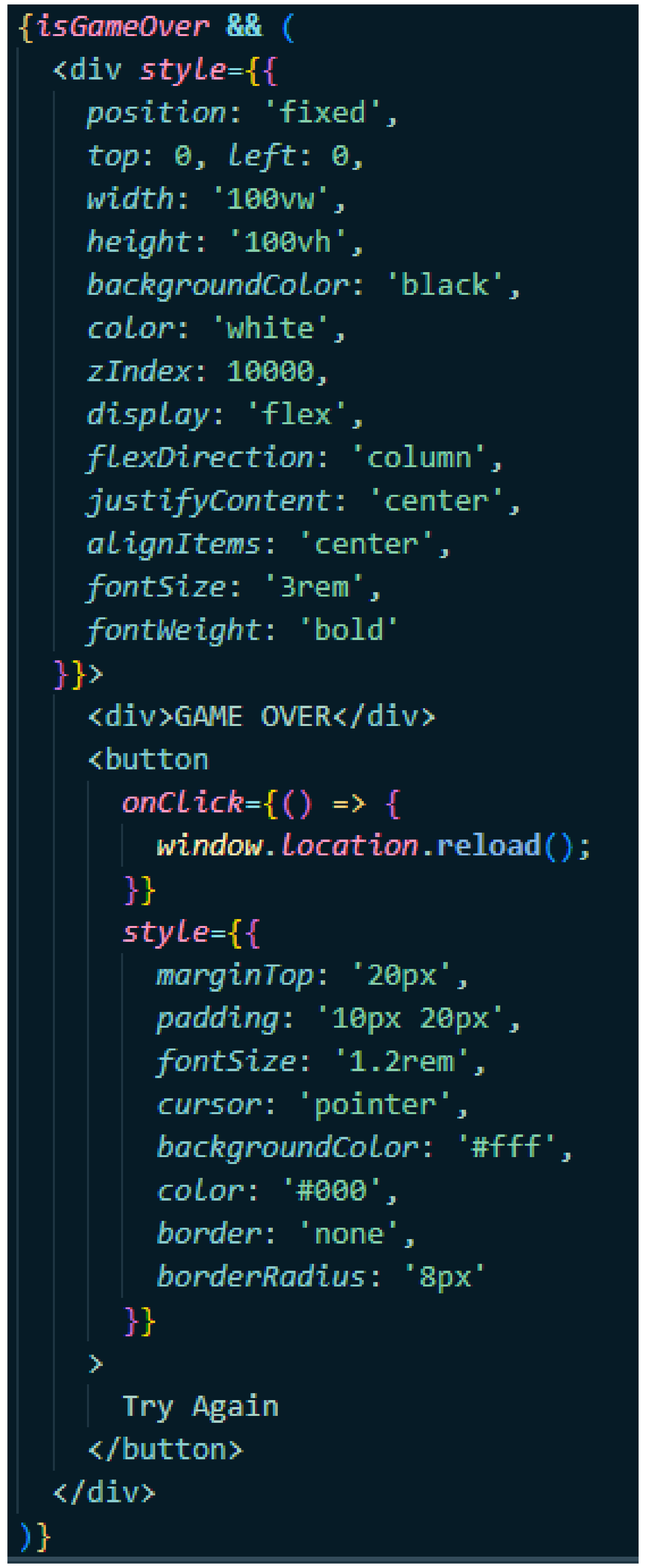
Penggunaan logic dan assets (jumpscare)



```
{jumpscareSrc && !isGameOver && (
  <div style={{
    position: 'fixed',
    top: 0, left: 0,
    width: '100vw',
    height: '100vh',
    backgroundColor: 'black',
    zIndex: 9999,
    display: 'flex',
    justifyContent: 'center',
    alignItems: 'center'
  }}>
    <img
      src={jumpscareSrc}
      alt="Jumpscare"
      style={{
        width: '100%',
        height: '100%',
        objectFit: 'cover'
      }}
    />
  </div>
)}
```

GAME

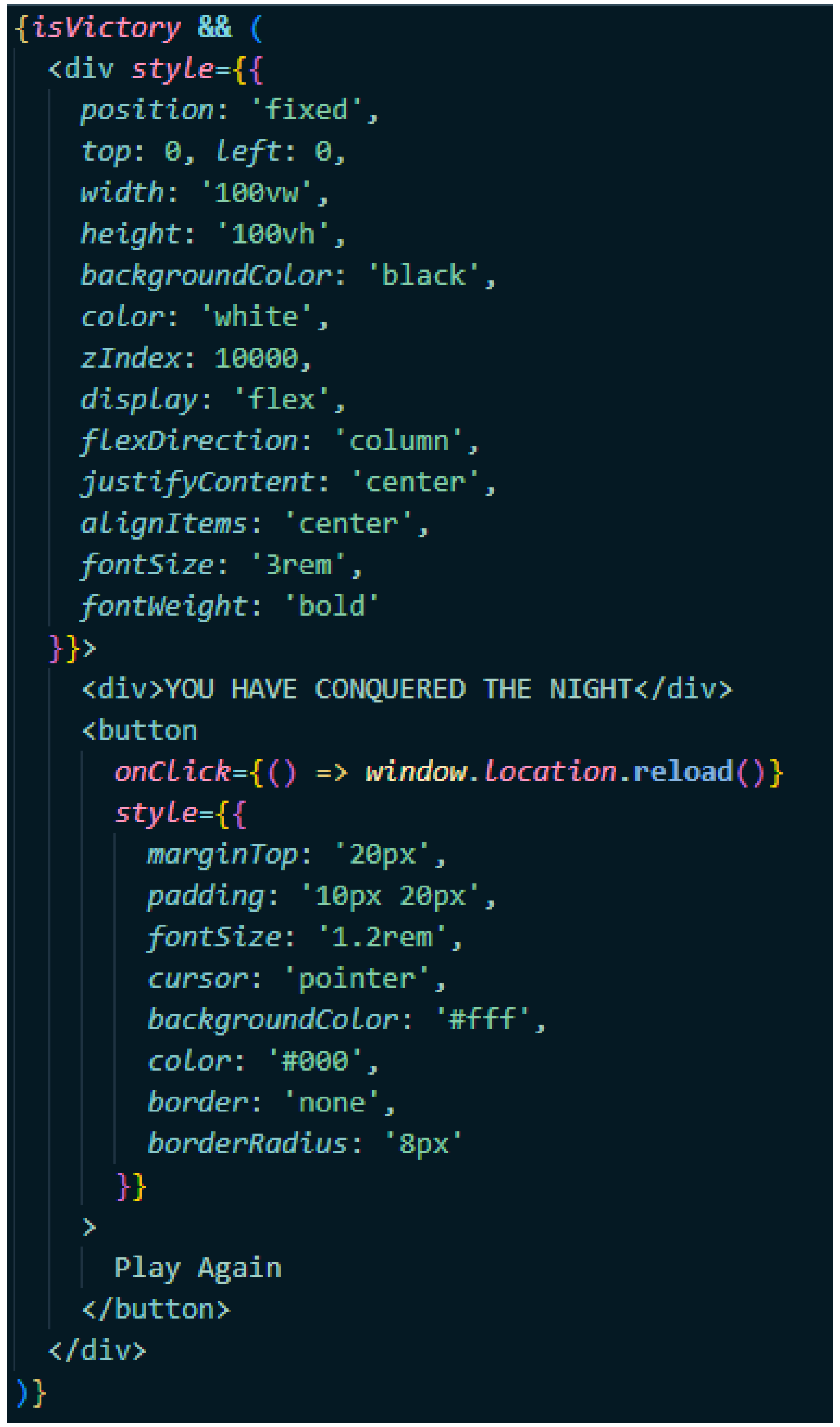
Penggunaan logic dan assets (game over)



```
{isGameOver && (
  <div style={{
    position: 'fixed',
    top: 0, left: 0,
    width: '100vw',
    height: '100vh',
    backgroundColor: 'black',
    color: 'white',
    zIndex: 10000,
    display: 'flex',
    flexDirection: 'column',
    justifyContent: 'center',
    alignItems: 'center',
    fontSize: '3rem',
    fontWeight: 'bold'
  }}>
    <div>GAME OVER</div>
    <button
      onClick={() => {
        window.location.reload();
      }}
      style={{
        marginTop: '20px',
        padding: '10px 20px',
        fontSize: '1.2rem',
        cursor: 'pointer',
        backgroundColor: '#fff',
        color: '#000',
        border: 'none',
        borderRadius: '8px'
      }}
    >
      Try Again
    </button>
  </div>
)}
```

GAME

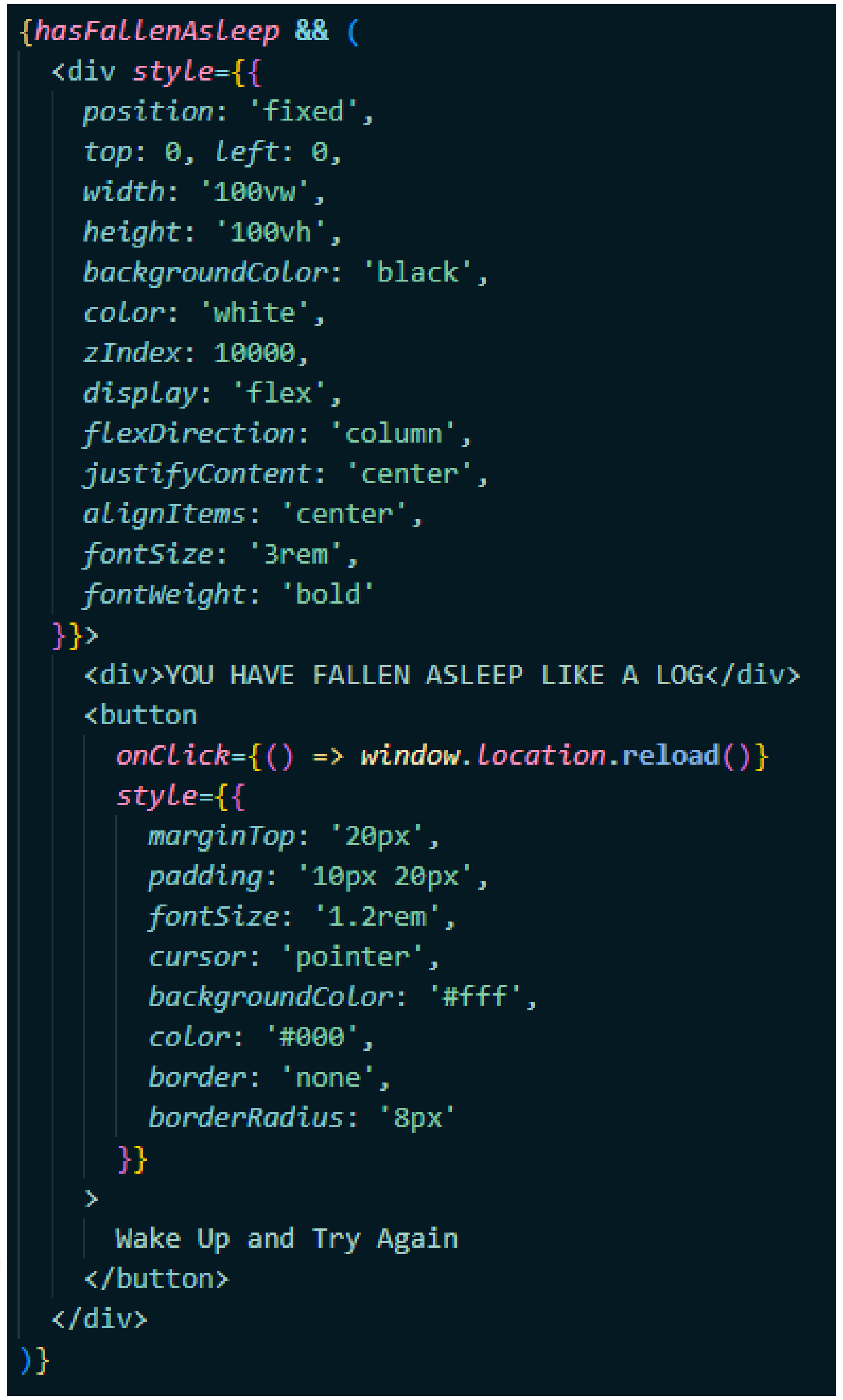
Penggunaan logic dan assets (victory)



```
{isVictory && (
  <div style={{
    position: 'fixed',
    top: 0, left: 0,
    width: '100vw',
    height: '100vh',
    backgroundColor: 'black',
    color: 'white',
    zIndex: 10000,
    display: 'flex',
    flexDirection: 'column',
    justifyContent: 'center',
    alignItems: 'center',
    fontSize: '3rem',
    fontWeight: 'bold'
  }}>
    <div>YOU HAVE CONQUERED THE NIGHT</div>
    <button
      onClick={() => window.location.reload()}
      style={{
        marginTop: '20px',
        padding: '10px 20px',
        fontSize: '1.2rem',
        cursor: 'pointer',
        backgroundColor: '#fff',
        color: '#000',
        border: 'none',
        borderRadius: '8px'
      }}
    >
      Play Again
    </button>
  </div>
)}
```

GAME

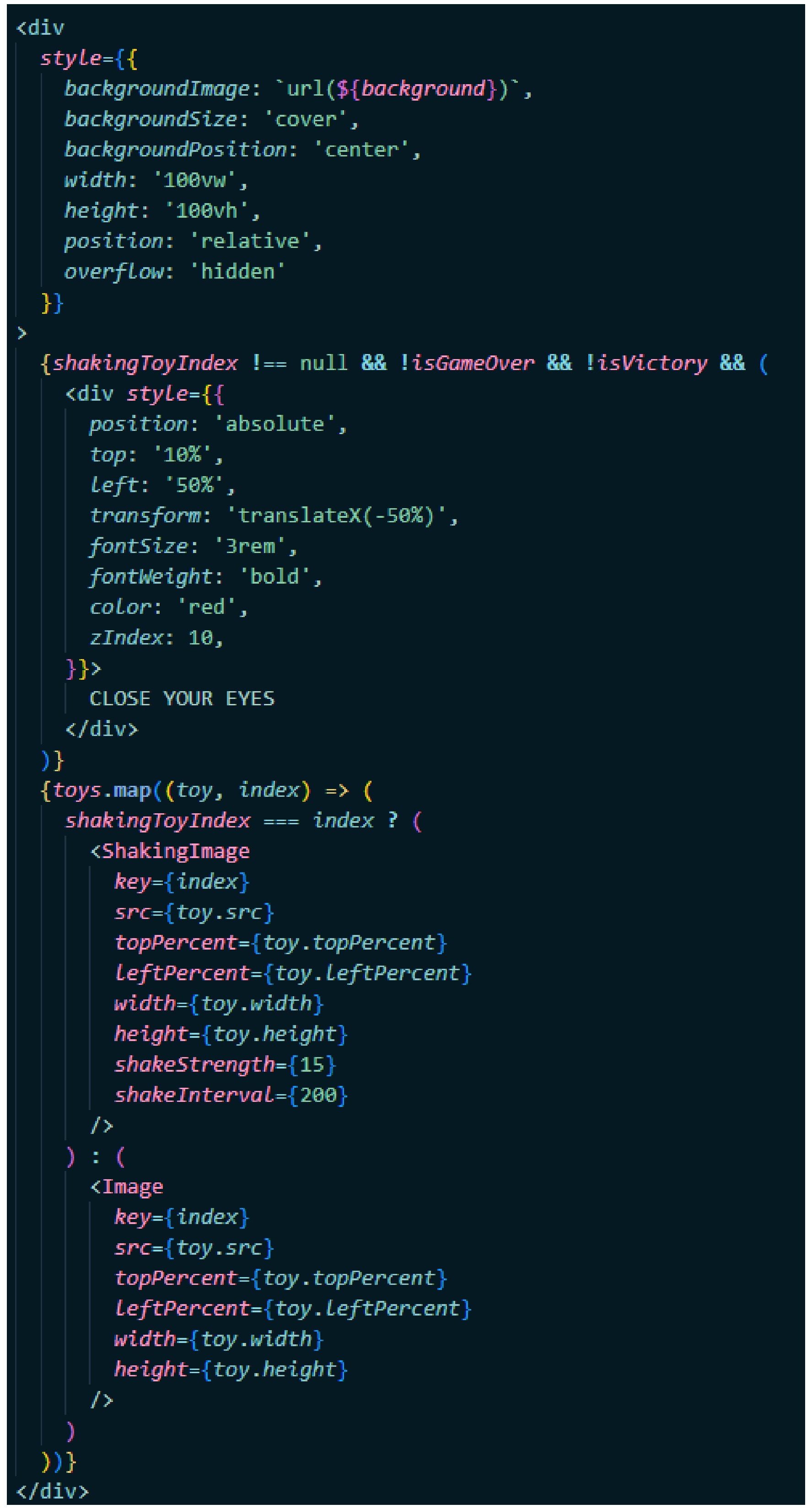
Penggunaan logic dan assets (easter egg)



```
{hasFallenAsleep && (
  <div style={{
    position: 'fixed',
    top: 0, left: 0,
    width: '100vw',
    height: '100vh',
    backgroundColor: 'black',
    color: 'white',
    zIndex: 10000,
    display: 'flex',
    flexDirection: 'column',
    justifyContent: 'center',
    alignItems: 'center',
    fontSize: '3rem',
    fontWeight: 'bold'
  }}>
    <div>YOU HAVE FALLEN ASLEEP LIKE A LOG</div>
    <button
      onClick={() => window.location.reload()}
      style={{
        marginTop: '20px',
        padding: '10px 20px',
        fontSize: '1.2rem',
        cursor: 'pointer',
        backgroundColor: '#fff',
        color: '#000',
        border: 'none',
        borderRadius: '8px'
      }}
    >
      Wake Up and Try Again
    </button>
  </div>
)}
```

GAME

Penggunaan logic dan assets (background, objek bergetar, dan mapping objek)



```
<div
  style={{
    backgroundImage: `url(${background})`,
    backgroundSize: 'cover',
    backgroundPosition: 'center',
    width: '100vw',
    height: '100vh',
    position: 'relative',
    overflow: 'hidden'
  }}
>
  {shakingToyIndex !== null && !isGameOver && !isVictory && (
    <div style={{
      position: 'absolute',
      top: '10%',
      left: '50%',
      transform: 'translateX(-50%)',
      fontSize: '3rem',
      fontWeight: 'bold',
      color: 'red',
      zIndex: 10,
    }}>
      CLOSE YOUR EYES
    </div>
  )}
  {toys.map((toy, index) => (
    shakingToyIndex === index ? (
      <ShakingImage
        key={index}
        src={toy.src}
        topPercent={toy.topPercent}
        leftPercent={toy.leftPercent}
        width={toy.width}
        height={toy.height}
        shakeStrength={15}
        shakeInterval={200}
      />
    ) : (
      <Image
        key={index}
        src={toy.src}
        topPercent={toy.topPercent}
        leftPercent={toy.leftPercent}
        width={toy.width}
        height={toy.height}
      />
    )
  ))}
</div>
```

THANK YOU