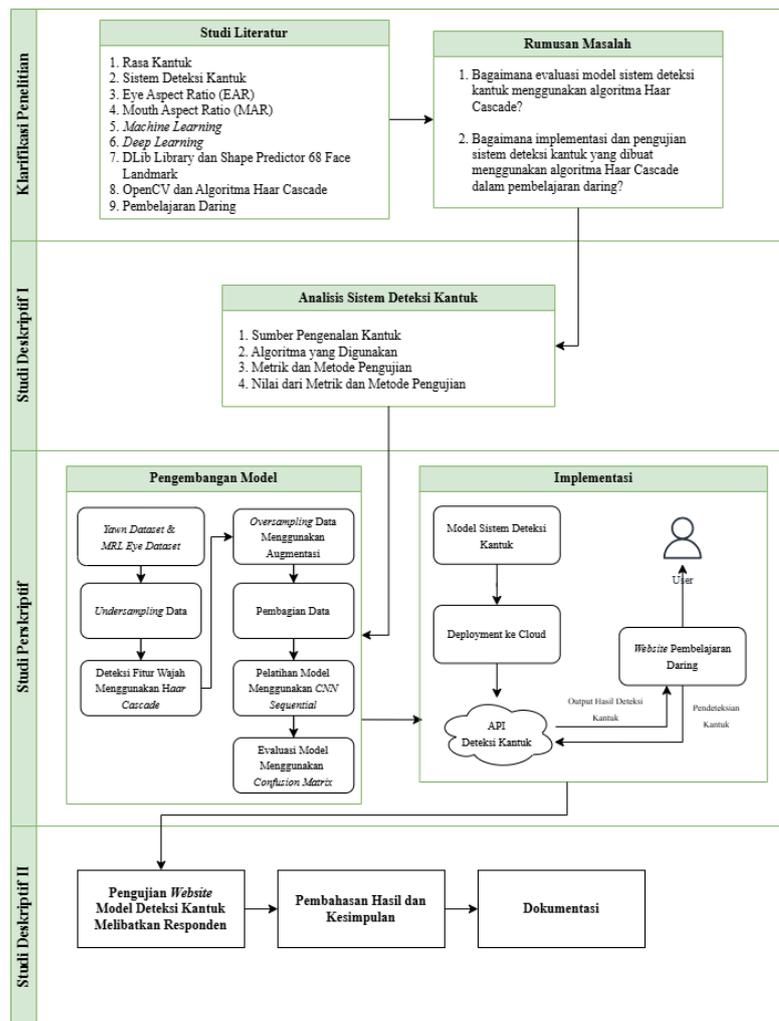


## BAB III METODE PENELITIAN

### 3.1 Desain Penelitian

*Design Research Methodology* (DRM) adalah metodologi yang memberikan kerangka kerja, pedoman, dan dorongan bagi peneliti untuk mendefinisikan pertanyaan penelitian, mengembangkan rencana yang sudah dibuat (perancangan), melakukan penelitian dengan cermat (pengembangan), dan merefleksikan temuan penelitian (evaluasi/pengujian) (Blessing dan Chakrabakti, 2009). Metodologi ini cocok digunakan dalam pengembangan sistem deteksi menurut penelitian yang dilakukan oleh Rahayu dkk. (2022).

Adapun desain penelitian yang akan dilakukan dapat dilihat dalam Gambar 3.1.



Gambar 3.1 Desain Penelitian

### 3.1.1 Klarifikasi Penelitian

Tahap pertama pada metode DRM adalah klarifikasi penelitian. Tujuan dari tahap ini adalah mengidentifikasi lebih lanjut permasalahan dan tujuan penelitian menurut latar belakang permasalahan yang dilakukan. Studi literatur dilakukan dengan cara pengumpulan data dan mengkaji terkait rasa kantuk, sistem deteksi kantuk, metrik yang biasa digunakan dalam mendeteksi kantuk, komponen-komponen pendukung yang diperlukan dalam pendeteksian wajah mengantuk termasuk algoritma Haar Cascade, pengujian menggunakan dan implementasi pembelajaran daring. Berdasarkan kajian studi literatur tersebut, peneliti bisa menentukan rumusan masalah dan tujuan penelitian yang terdapat pada Bab I. Pembahasan kajian studi literatur terdapat pada Bab II.

### 3.1.2 Studi Deskriptif I

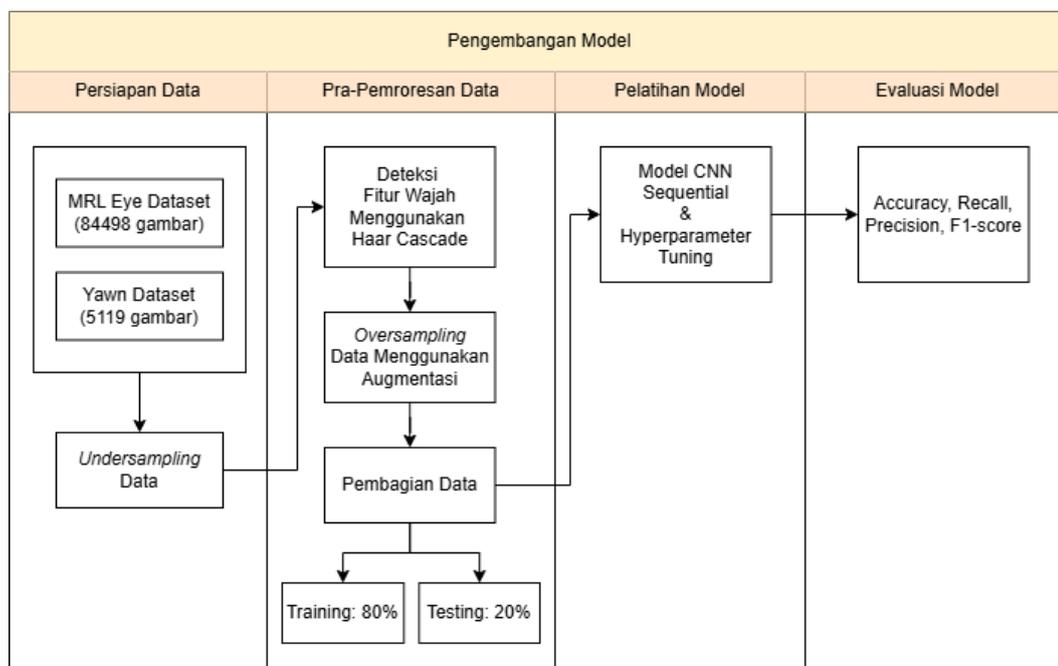
Pada tahap ini, akan dilakukan pengkajian literatur yang lebih komprehensif tentang penelitian-penelitian sebelumnya yang relevan dengan sistem deteksi kantuk. Kajian lebih lanjut terkait sumber pengenalan kantuk, algoritma deteksi kantuk yang digunakan, metrik dan metode pengujian, dan nilai dari hasil metrik dan pengujian. Tahap ini bertujuan agar peneliti bisa memahami lebih dalam masalah yang diteliti. Penelitian terdahulu yang berhubungan dengan sistem deteksi kantuk dan algoritma *Haar Cascade* akan dianalisis secara komparatif untuk memahami perkembangan terkini dari topik yang akan dibahas.

### 3.1.3 Studi Preskriptif

Tahap ketiga pada metode DRM adalah studi preskriptif. Pada tahapan ini, peneliti berfokus pada pengembangan model deteksi kantuk dan juga implementasinya. Pengembangan model deteksi kantuk dilakukan untuk membuktikan apakah dengan menggunakan dataset latih yang sudah ditetapkan pada model referensi, model memiliki akurasi yang memuaskan. Proses pengimplementasian dilakukan dengan membuat sebuah website pembelajaran asinkronus yang terdiri dari fitur video YouTube dan kamera yang sudah dipasang sistem deteksi kantuk. Adapun sistem deteksi kantuk sebelumnya diproses secara *dockerization* lalu diterapkan ke layanan *cloud* dengan hasil akhir berupa API.

### 3.1.3.1 Pengembangan Model Sistem Deteksi Kantuk

Pengembangan model dilakukan dalam beberapa tahap dan digambarkan dalam Gambar 3.2.



Gambar 3.2 Pengembangan Model Sistem Deteksi Kantuk

Secara umum, proses pengembangan model sistem deteksi kantuk diawali dengan persiapan data. Kedua dataset akan melakukan proses *undersampling* terlebih dahulu karena jumlah gambar di *MRL Eye Dataset* yang terlalu timpang. Di tahap pra-pemrosesan data, gabungan *dataset* akan dideteksi menggunakan algoritma Haar Cascade. Apabila terjadi ketidakseimbangan data akan dilakukan augmentasi data. Setelah itu, dilakukan pembagian data menjadi data *training* dan data *testing*. Pelatihan model dilakukan menggunakan model *CNN Sequential*. Selanjutnya, model akan dievaluasi menggunakan metrik *accuracy*, *recall*, *precision*, dan *F1-score*.

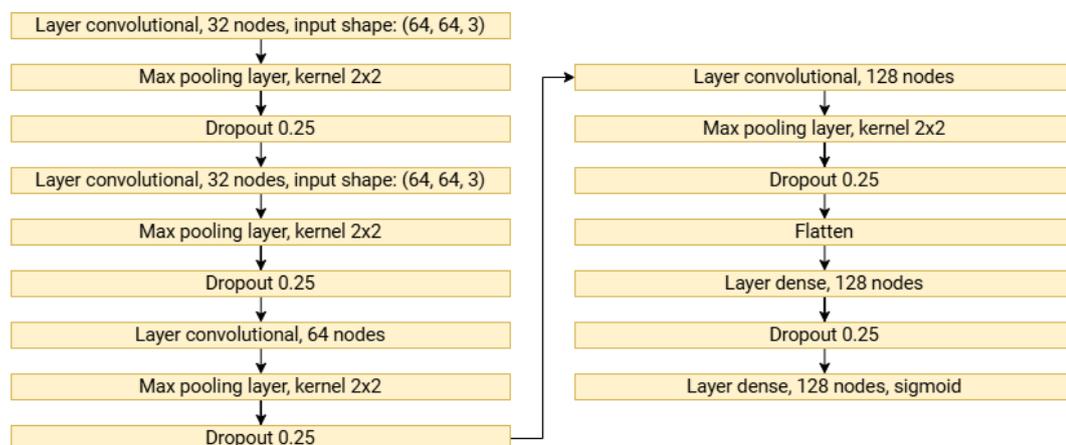
*Dataset* pertama yang akan digunakan adalah *MRL Eye Dataset* yang dibuat oleh Fusek (2018) dan diunggah oleh Djerarda (2022). *Dataset* mata ini pernah digunakan dalam beberapa penelitian seperti penelitian Rahman dkk. (2020); Jahan dkk. (2023); Shukla, Tiwari, dan Jha (2023). Adapun *dataset* kedua akan menggunakan *Yawn Dataset* yang dibuat dan diunggah oleh Vazquez (2021).

*Dataset* ini pun digunakan dalam berbagai penelitian yakni pada penelitian Lin dan Zuo (2024); Tian, Albakry, dan Du (2024). Kedua *dataset* dipilih karena sesuai dengan fitur wajah yang akan diteliti pendeteksian kantuknya yakni mata dan mulut serta sudah banyak digunakan dalam berbagai penelitian. Namun, selisih antara jumlah gambar pada kedua *dataset* membuat peneliti perlu melakukan penyesuaian dengan cara *undersampling*. *Undersampling* dilakukan dengan cara mengurangi gambar pada dataset terbanyak sehingga memiliki jumlah yang sama dengan dataset dengan gambar yang lebih sedikit. Hal ini dilakukan untuk menghindari gangguan pada hasil prediksi (Romadhan, Rizky, Faqihuddin, dan Siradjuddin, 2024).

Pada tahap pra-pemrosesan data, gabungan dataset akan dideteksi fitur wajahnya menggunakan algoritma *Haar Cascade*. Pada tahapan ini diperlukan model terlatih (*pre-trained*) berbentuk XML sebagai media pendeteksian. Parameter seperti *scaleFactor*, *minNeighbors*, dan *minSize*-pun diperlukan sebagai penerapan dari metode *detectMultiScale* (Antipona, Magsino, Dioses, dan Mata, 2024). Di dalamnya pun, gambar yang sudah dideteksi akan diubah skalanya menjadi 64×64 piksel. Skala ini dipilih agar mengurangi waktu komputasi untuk memproses setiap gambar dan menjaga setiap detail pada gambar (Magán dkk, 2022). Setelah proses pendeteksian, apabila masih terjadi perbedaan jumlah gambar pada setiap kelas dataset, maka akan dilakukan proses *oversampling*. *Oversampling* dilakukan dengan cara menyeimbangkan *dataset* yang memiliki jumlah gambar lebih sedikit agar diperbanyak sesuai jumlah gambar *dataset* terbanyak.. Adapun proses memperbanyak ini bisa dilakukan dengan cara augmentasi (Lashgari, Lian, dan Maoz, 2020). Cara augmentasi lazim menggunakan *library ImageDataGenerator* yang biasa digunakan dalam pra-pemrosesan data (Balamurugan, 2023). Beberapa metode augmentasi yang digunakan antara lain pencerminan secara horizontal (*horizontal flipping*), rotasi (*rotation*), pembesaran (*zoom*), dan pemotongan (*cropping*) (Zheng, Zhao, Wang, dan Yang, 2020; Mohamed, Patel, dan Naicker, 2023). Selanjutnya adalah pembagian data menjadi data latih (*training data*) dan data tes (*testing data*) dengan rasio 80:20. Penentuan rasio ini sesuai dengan penelitian Mardi, Ashtani, dan Mikaili (2011); Madni, Raza,

Sehar, Thalji, dan Abualigah (2024) yang model buatan penelitian mereka menghasilkan akurasi yang tinggi. Pengambilan rasio 80:20 didasari oleh konsep *Pareto Principle* yang menyatakan bahwa sekitar 80% hasil berasal dari 20% penyebabnya (Abyad, 2020). Dalam konteks *machine learning*, hal tersebut bisa diartikan sebagai 80% data latih memberikan informasi bagi model untuk mengenali pola utama pada dataset serta 20% data tes untuk menilai kinerja model. Tentunya hal ini diperkuat oleh penelitian Gholamy, Kreinovich, dan Kosheleva (2018) yang mengatakan bahwa dengan menentukan data latih di angka 80% dan data tes di angka 20% merupakan langkah yang paling efektif karena dapat memaksimalkan keakuratan dengan meminimalkan eror.

Di tahap pelatihan model, model akan dilatih menggunakan *Sequential CNN*. Adapun arsitektur yang digunakan sesuai yang digambarkan pada Gambar 3.3.



Gambar 3.3 Arsitektur *Sequential CNN*

Arsitektur yang digunakan merupakan adaptasi dari arsitektur CNN-1 dan CNN-2 dari penelitian Mohamed dkk. (2023). Alasan penggunaan model CNN adalah kemampuan ekstraksi pola-pola spasial pada fitur mata terbuka dan tertutup secara otomatis, mampu mencapai akurasi yang tinggi untuk mengklasifikasi gambar mata sesuai penelitian terkait, dan meningkatkan kemampuan generalisasi terhadap perubahan cahaya dan sudut pandang wajah.

Pelatihan model akan dilakukan menggunakan jumlah *epoch* sebesar 75 *epoch*. Hal tersebut dilakukan untuk mencegah model mempelajari terlalu banyak *noise* yang bisa menyebabkan *overfitting* (Fudholi, Nayoan, Suyuti, dan Rahmadi, 2021) dan akan menghasilkan akurasi yang lebih besar bila dibandingkan dengan menggunakan *epoch* 50 dan 100 (Zahro, Agustin, dan Ridlo, 2023). Adapun untuk *hyperparameter tuning*, beberapa parameter seperti *learning rate*, *optimizer*, dan *batch size* mempunyai pengaruh signifikan terhadap kinerja model. Terkait *optimizer*, ada beberapa yang biasa digunakan dalam *deep learning* yakni Adam, AdamW, dan SGD (Xu, Xiang, Cai, dan Wen, 2024); (Giannuzzi, 2024); (Nguyen, Putro, dan Jo, 2024). Nilai *learning rate* ada di rentang 0.1, 0.01, 0.001, dan 0.0001 di mana fungsinya sebagai pengoptimasi model (Tang, Mhamdi, McLernon, Zaidi, dan Ghogho, 2016). Parameter terakhir yakni *batch size* jika dataset dalam bentuk yang besar, *batch size* dengan nilai lebih besar pun bisa digunakan. Ukuran *batch size* tersebut adalah 32 atau 64 (Masters dan Luschi, 2018).

Pada tahapan terakhir yakni evaluasi model akan menggunakan *confusion matrix* dengan metrik *accuracy*, *recall*, *precision*, dan *f1-score*. *Confusion matrix* merupakan alat ukur yang biasa digunakan saat memecahkan masalah klasifikasi. Matriks ini dapat digunakan untuk klasifikasi dua kelas (biner) maupun masalah klasifikasi multikelas (Kulkarni, Chong, dan Batarseh, 2020).

### 3.1.3.2 Implementasi Sistem Deteksi Kantuk

Model deteksi kantuk yang sudah dibuat akan peneliti kembangkan dalam bentuk API. API mampu memisahkan pemrosesan dari sisi klien dengan sisi server. Pemisahan ini dilakukan untuk mengurangi beban komputasi di sisi klien ke sisi server yang mempunyai sumber data lebih besar untuk melakukan pemrosesan deteksi kantuk. Model deteksi kantuk akan dikemas dalam bentuk kontainer. Kontainer adalah paket berisi kode perangkat lunak beserta pustaka dan dependensi sistem operasi yang dibutuhkan untuk proses perjalanan aplikasi sehingga bisa berjalan di berbagai lingkungan. Kemudian kontainer akan melalui proses *dockerization* oleh Docker. Docker adalah sebuah layanan yang mampu melakukan proses pengemasan dan pengeksekusian kontainer itu. Selanjutnya, hasil dari

*dockerization*, API akan dijalankan dalam layanan *cloud*. Media server yang akan digunakan adalah Google Cloud Run. Google Cloud Run adalah layanan serverless dari Google yang digunakan untuk mengembangkan dan menjalankan aplikasi containerized dengan skalabilitas tinggi. Cloud Run secara otomatis menyesuaikan skala sesuai dengan lalu lintas permintaan, bahkan dapat turun hingga nol selama periode idle, dan hanya membebankan biaya berdasarkan sumber daya yang digunakan (Khatiwada dan Dhakal, 2024).

### 3.1.4 Studi Deskriptif II

Tahap keempat pada metode DRM adalah studi deskriptif II. Pada tahap ini akan dilakukan evaluasi dari sistem deteksi kantuk yang akan dibuat. Evaluasi dilakukan dengan cara pengujian melibatkan responden yang menguji langsung *website* deteksi kantuk. Pengujian dilakukan untuk menguji akurasi dari sistem deteksi kantuk. Setelah itu, akan dilakukan evaluasi hasil dengan menghitung nilai akurasi menurut beberapa uji coba terkait. Hasil, pembahasan, dan simpulan yang menjawab pertanyaan pada rumusan masalah akan didokumentasikan pada bab 4 dan 5. Sebagai penutup, peneliti akan mencantumkan rekomendasi dan saran bagi peneliti di masa depan agar bisa meningkatkan kualitas penelitian terkait.

## 3.2 Populasi dan Sampel

Pada penelitian ini, keterlibatan pihak untuk menjadi populasi dan sampel dibutuhkan untuk mengevaluasi sistem yang sudah dibuat sebelumnya. Partisipan diminta untuk menggunakan *website* kelas pembelajaran daring yang sudah diimplementasikan sistem deteksi kantuk. Kriteria partisipan yang akan terlibat dalam penelitian antara lain:

1. Mahasiswa yang termasuk kelompok remaja akhir berusia antara 18-24 tahun.
2. Mahasiswa yang pernah dan sudah terbiasa dalam melakukan pembelajaran daring.
3. Mahasiswa yang terbiasa menggunakan perangkat laptop dan/atau komputer.

Penentuan kriteria ini bertujuan memastikan partisipan merepresentasikan populasi target pengguna sistem, yakni mahasiswa yang pernah melakukan pembelajaran daring selama perkuliahan. Pemilihan partisipan menggunakan metode *purposive sampling*, di mana kriteria utama adalah mahasiswa yang memiliki pengalaman dalam pembelajaran daring. Menurut Sugiyono (2010), *purposive sampling* dilakukan dengan mengambil sampel sumber data dengan mempertimbangkan karakteristik tertentu yang sesuai dengan tujuan penelitian.

Jumlah partisipan ditetapkan sebanyak 20 orang mahasiswa, mengacu pada penelitian yang dilakukan oleh Nggiku, Rabi, dan Subairi (2022) yang melakukan 20 kali pengujian sistem deteksi kantuk pengemudi menurut dua sudut wajah ( $0^\circ$  dan  $30^\circ$ ). Dengan melakukan 20 kali pengujian, tingkat keberhasilan pengujian adalah akurasi rata-rata 92,5% dan eror 6.5% (Nggiku dkk., 2022). Sehingga pemilihan jumlah partisipan tersebut dinilai memadai untuk validasi awal sistem deteksi kantuk dan diadaptasi sebagai dasar untuk memastikan konsistensi dalam evaluasi kinerja sistem deteksi kantuk.

### 3.3 Alat dan Bahan Penelitian

Penelitian ini akan menggunakan alat-alat penelitian yang terdiri dari perangkat keras dan perangkat lunak. Perangkat keras yang digunakan untuk pengembangan aplikasi menggunakan spesifikasi sebagai berikut:

- a. Processor 11<sup>th</sup> Gen Intel® Core™ i5-1135G7 @ 2.40GHz (8CPUs), ~2.4GHz
- b. RAM 8GB DDR4
- c. NVIDIA GeForce MX330
- d. SSD 512GB

Perangkat lunak yang digunakan dalam pengembangan meliputi sistem operasi, aplikasi, dan *library* sebagai berikut:

- a. Sistem Operasi Windows 11 Home Single Language 64-bit
- b. Microsoft Edge
- c. Visual Studio Code
- d. Google Collaboratory
- e. Python Library

### 3.4 Instrumen Penelitian

Instrumen yang dipakai untuk menguji pengembangan model bergantung kepada pengujian yang akan dilakukan yakni melakukan eksperimen menurut metrik-metrik tertentu seperti *confusion matrix*. *Confusion matrix* adalah alat ukur kinerja model klasifikasi yang digunakan untuk menilai seberapa baik model dapat mengklasifikasikan data ke dalam kelas yang benar. *Confusion matrix* dengan bentuk 2×2 terdiri dari empat kuadran, masing-masing mewakili kombinasi antara prediksi model dan kelas aktual data. Instrumen ini digunakan untuk pengembangan model deteksi kantuk.

Tabel 3.1 *Confusion Matrix*

Jenis Kelas	Aktual Positif	Aktual Negatif
Prediksi Positif	True Positive (TP)	False Positive (FP)
Prediksi Negatif	False Negative (FN)	True Negative (TN)

Adapun formula penghitungan nilai-nilai dalam *confusion matrix* adalah sebagai berikut:

1. *True Positive* (TP): Jumlah data yang kelas aktual dan prediksinya sama-sama positif serta sistem berhasil mengklasifikasikan dengan benar
2. *False Positive* (FP): Jumlah data yang kelas aktual negatif, tetapi prediksinya positif. Sistem mengklasifikasikannya positif alih-alih negatif.
3. *False Negative* (FN): Jumlah data yang kelas aktual positif, tetapi prediksinya negatif. Sistem mengklasifikasikannya negatif alih-alih positif.
4. *True Negative* (TN): Jumlah data yang kelas aktual dan prediksinya sama-sama negatif serta sistem berhasil mengklasifikasikan dengan benar.

Nilai-nilai dalam *confusion matrix* dapat digunakan untuk menginterpretasi kinerja model klasifikasi. Beberapa metrik kinerja yang umum digunakan untuk menginterpretasi *confusion matrix* adalah sebagai berikut:

1. Akurasi

Akurasi adalah rasio antara jumlah data yang diprediksi dengan benar terhadap jumlah total data.

$$\text{Akurasi} = \frac{TP+TN}{TP + FP + TN + FN} \quad (1)$$

## 2. Presisi

Presisi adalah rasio antara jumlah data yang diprediksi positif dengan benar terhadap jumlah data yang diprediksi positif.

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2)$$

## 3. Recall

Recall adalah rasio antara jumlah data yang diprediksi positif dengan benar terhadap jumlah data yang sebenarnya positif.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

## 4. F1 score

F1 score adalah kombinasi antara akurasi dan recall.

$$\text{F1 score} = \frac{2 \times \text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (4)$$

Adapun instrumen penelitian mengenai pengujian *website* deteksi kantuk, peneliti merancang 4 skenario uji coba deteksi kantuk yang tercantum pada Tabel 3.2 di bawah:

Tabel 3.2 Skenario Uji Coba Sistem Deteksi Kantuk

No.	Skenario	Ekspektasi
1.	Responden mempraktekkan matanya dibuka dan mulutnya ditutup	Sistem deteksi kantuk tidak akan mendeteksi kantuk pada responden
2.	Responden mempraktekkan matanya ditutup dan mulutnya ditutup	Sistem deteksi kantuk akan mendeteksi kantuk pada responden
3.	Responden mempraktekkan matanya dibuka dan mulutnya menguap	Sistem deteksi kantuk akan mendeteksi kantuk pada responden

4.	Responden mempraktekkan matanya ditutup dan mulutnya menguap	Sistem deteksi kantuk akan mendeteksi kantuk pada responden
----	--	---

Hasil dari setiap skenario adalah jawaban ‘Terdeteksi Mengantuk’ dan ‘Tidak Terdeteksi Mengantuk’. Setiap responden mempraktekkan 4 skenario itu dalam waktu 3 detik. Hal ini sesuai dengan penelitian Puteri dan Utamingrum (2020) yang menggunakan indikator 3 detik.. Hal ini didukung oleh penelitian Saputra, Erwanto, dan Rahayu (2021) yang menyebutkan bahwa 3 detik hingga 5 detik merupakan durasi *microsleep* yang sangat singkat. *Microsleep* adalah suatu keadaan saat seseorang tertidur atau mengantuk secara tiba-tiba dalam beberapa detik.