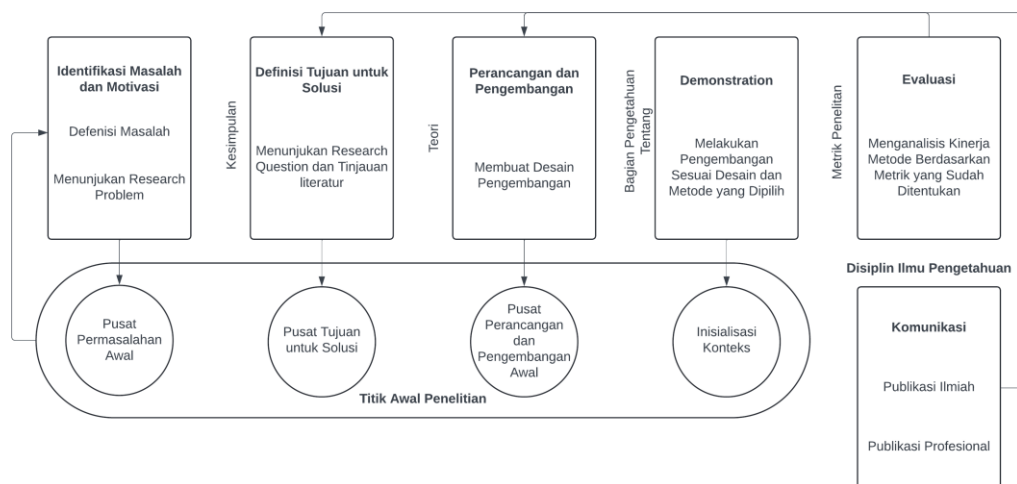


BAB III

METODOLOGI

3.1 Metodologi berdasarkan *Design Science Research Methodology*

Design Science Research Methodology (DSRM) akan digunakan sebagai kerangka kerja penelitian dalam laporan ini. Kerangka kerja ini dikembangkan oleh Peffers, Tuunanen, Rothenberger, dan Chatterjee pada tahun 2007. DSRM dipilih karena memiliki target dan objektif yang sama dengan penelitian ini yaitu pengembangan artefak teknologi baru, seperti model VGG, CBAM, BN, dan SK yang diusulkan dalam penelitian ini. Selain itu, metodologi ini memungkinkan iterasi sistematis antara pengembangan konsep dan evaluasi empiris (Pries-Heje et al., 2017). Metodologi penelitian DSRM terdiri dari enam tahap penelitian yang diilustrasikan dalam Gambar 3. 1 (Nugroho, 2013).



Gambar 3. 1 Diagram alir metodologi penelitian

3.1.1 Identifikasi Masalah dan Motivasi

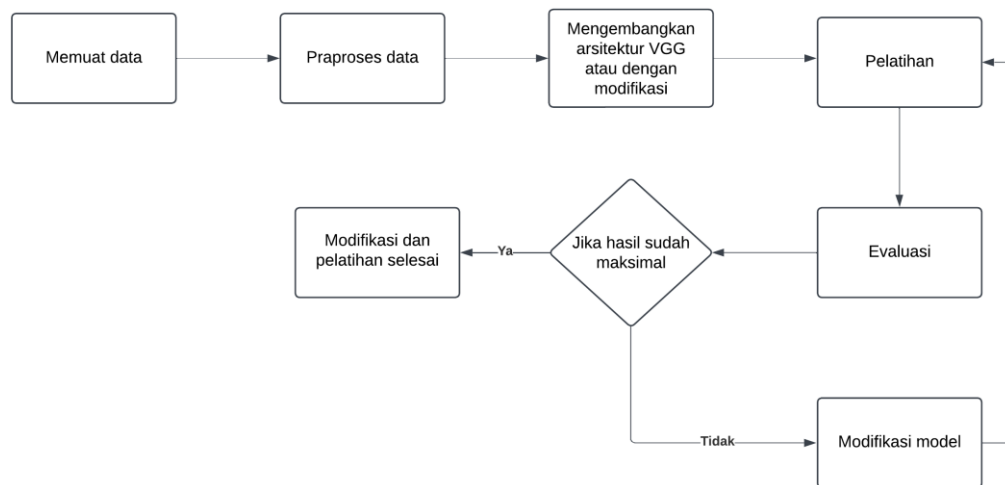
Pada tahap identifikasi masalah dan motivasi dalam laporan ini, ditunjukkan permasalahan yang menjadi fokus penelitian. Untuk menganalisis masalah tersebut, digunakan berbagai referensi, termasuk survei terbaru, studi *state-of-the-art*, serta studi kasus terkini. Setelah proses analisis, masalah yang mendasari laporan ini berhasil diidentifikasi. Hasil dari proses ini adalah perumusan permasalahan penelitian serta latar belakang penelitian.

3.1.2 Definisi Tujuan untuk Solusi

Pada tahap definisi tujuan untuk solusi dalam laporan ini, ditunjukkan pertanyaan penelitian dan tinjauan literatur. Setelah mengidentifikasi masalah, dilakukan proses tinjauan literatur untuk mengetahui kesenjangan dalam penelitian yang telah dilakukan sebelumnya. Analisis terhadap kesenjangan penelitian ini menghasilkan pertanyaan penelitian dan tujuan penelitian yang jelas. Proses ini selanjutnya menghasilkan landasan teori yang solid untuk penelitian.

3.1.3 Perancangan dan Pengembangan

Pada tahap perancangan dan pengembangan dalam laporan ini, dibuat desain pengembangan yang komprehensif. Tahapan-tahapan pengembangan dan desain tersebut dijelaskan berdasarkan Gambar 3. 2.



Gambar 3. 2 Diagram alir pengembangan dan desain

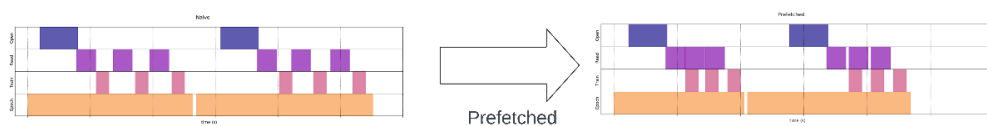
3.1.3.1 Memuat Data

Pada tahap memuat data, data yang sudah ditentukan akan diunduh pada lingkungan google colab dengan API TensorFlow. Berikutnya berdasarkan tinjauan pustaka, *dataset* yang akan digunakan meliputi CIFAR-10 dan CIFAR-100, yang dapat diimpor melalui *dataset* TensorFlow (Li et al., 2019). *Dataset* CIFAR-10 terdiri dari 60.000 gambar berwarna berukuran 32x32 piksel yang dikelompokkan ke dalam 10 kelas, dengan masing-masing kelas berisi 6.000 gambar. Sementara itu, *dataset* CIFAR-100 terdiri dari 60.000 gambar berwarna berukuran 32x32 piksel yang dikelompokkan ke dalam 100 kelas, dengan masing-masing kelas berisi 600 gambar.

3.1.3.2 Praproses Data

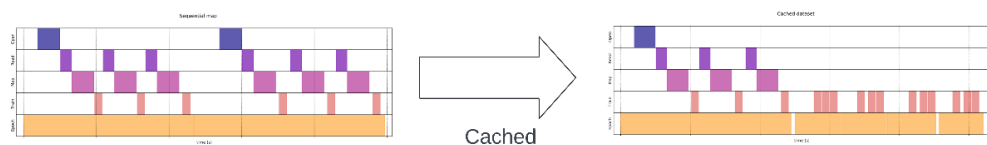
Pada tahap praproses data, data yang sudah dimuat akan melalui praproses transformasi data yang meliputi proses *shuffle*, *batch*, dan normalisasi. Untuk optimasi proses pelatihan diimplementasikan *cache* dan *prefetch*. Praproses dan

optimasi tersebut diintegrasikan menjadi *pipeline* data untuk model. Pada proses *shuffle* memiliki ukuran *buffer* 300. Pada proses *batch* memiliki ukuran 32 yang direkomendasikan (Bengio, 2012). Pada proses *prefetch* akan dilakukan secara dinamik ketika berjalan. Proses *prefetch* dan *cache* diilustrasikan pada Gambar 3. 3 dan Gambar 3. 4. Berikut tahapan *pipeline* data yang dilakukan. Pertama, dilakukan *shuffle* pada data. Kedua, dilakukan proses normalisasi data gambar sehingga nilai matriks gambar diantara nilai nol dan satu. Ketiga, dilakukan proses *cache*. Keempat, dilakukan proses *batch* data. Yang terakhir, dilakukan proses *prefetch* data. Semua proses dilakukan dengan menggunakan alat TensorFlow (TensorFlow Team, 2024).



Gambar 3. 3 Ilustrasi proses *prefetch*

Pada Gambar 3. 3 merupakan ilustrasi proses *prefetch*. Proses *prefetch* menumpang tindih praproses dengan eksekusi model dari langkah pelatihan (TensorFlow Team, 2024).



Gambar 3. 4 Ilustrasi proses *cache*

Pada Gambar 3. 4 merupakan ilustrasi proses *cache*. Proses *cache* menyimpan hasil transformasi *dataset* seperti membuka file dan membaca data baik di memori atau di penyimpanan lokal (TensorFlow Team, 2024).

3.1.3.3 Mengembangkan Arsitektur

Pada tahap mengembangkan arsitektur VGG atau dengan modifikasi, pada tahap ini dibuat arsitektur VGG yang disesuaikan dengan *size* dari gambar yang sudah diproses. Alat yang akan digunakan untuk membuat dan memodifikasi arsitektur menggunakan alat TensorFlow.

3.1.3.4 Pelatihan

Pada tahap pelatihan, model akan dilatih berdasarkan *dataset* dan *epoch* yang sudah ditentukan menggunakan GPU T4 pada lingkungan google colab. Alat yang akan digunakan untuk melakukan proses pelatihan menggunakan TensorFlow. Selain itu, proses augmentasi akan dilakukan model bersamaan dengan proses pelatihan. Pada proses ini, model yang akan disimpan merupakan model terbaik dari *epoch* yang sudah ditentukan.

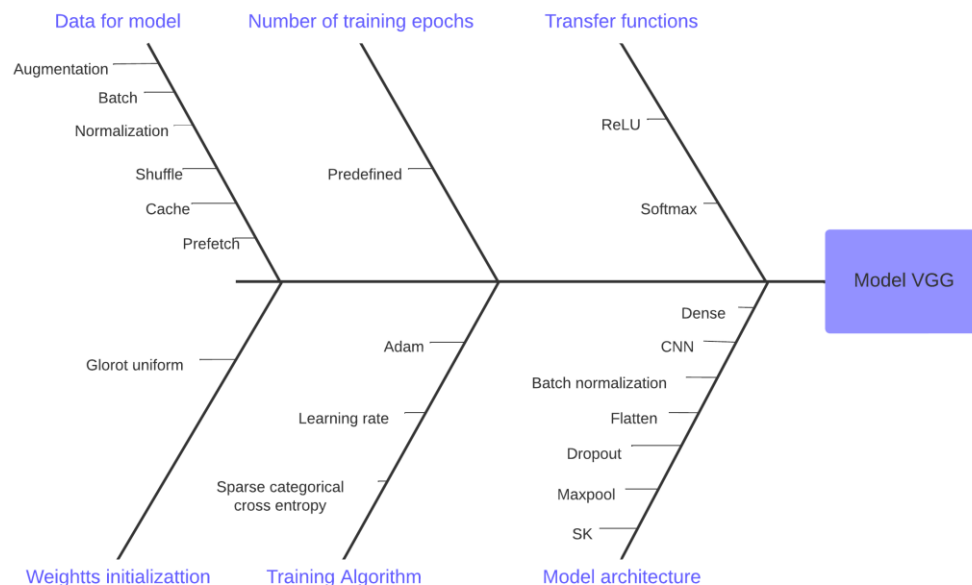
3.1.3.5 Evaluasi

Pada tahap evaluasi, model dievaluasi dengan *dataset* tes berdasarkan metrik yang sudah ditentukan. Alat yang digunakan untuk melakukan klasifikasi diantaranya Numpy, Tensorflow, dan Sklearn.

Setelah beberapa kali percobaan jika sudah ditemukan model dengan performa terbaik maka proses pengembangan selesai jika belum akan dilakukan modifikasi pada model.

Selanjutnya, untuk memberikan pemahaman yang lebih mendalam mengenai pemodelan yang digunakan dalam laporan ini, dibuatlah diagram *fishbone* dengan merujuk pada penelitian yang dilakukan oleh Madić dan Radovanović (Madić &

Radovanović, 2012). Berdasarkan referensi tersebut, disusunlah diagram *fishbone* yang ditunjukkan pada Gambar 3. 5.



Gambar 3. 5 Diagram *fishbone*

Penjelasan mengenai diagram *fishbone* adalah sebagai berikut:

1. *Transfer function* merupakan fungsi aktivasi yang akan digunakan pada model yang diantaranya *ReLU* dan *softmax*.
2. *Number of training epochs* merupakan banyaknya *epochs* dalam pelatihan. Dalam laporan ini akan ditentukan terlebih dahulu.
3. *Data for model* merupakan proses penyiapan data yang berupa proses *batch*, *normalization*, *cache*, *shuffle*, *prefetch*, dan *augmentation*.
4. *Model architecture* merupakan arsitektur yang akan dipakai pada model yang diantaranya *CNN*, *dense*, *maxpool*, *BN*, *flatten*, *dropout*, dan *SK*.

5. *Training algorithm* merupakan metode pelatihan yang akan digunakan yang meliputi *adam*, *learning rate*, dan *sparse categorical cross entropy*.
6. Inisialisasi bobot dan bias pada model ini dilakukan menggunakan metode *glorot uniform*, yang merupakan metode inisialisasi default pada kerangka kerja yang digunakan.

3.1.4 Demonstration

Pada tahap demonstrasi dalam laporan ini, pengembangan dilakukan berdasarkan desain yang telah dibuat sebelumnya. Tahap ini menghasilkan grafik pelatihan, kinerja model, dan waktu pelatihan model. Hasil-hasil tersebut akan dievaluasi dan dibandingkan dengan modifikasi yang dilakukan.

3.1.5 Evaluasi

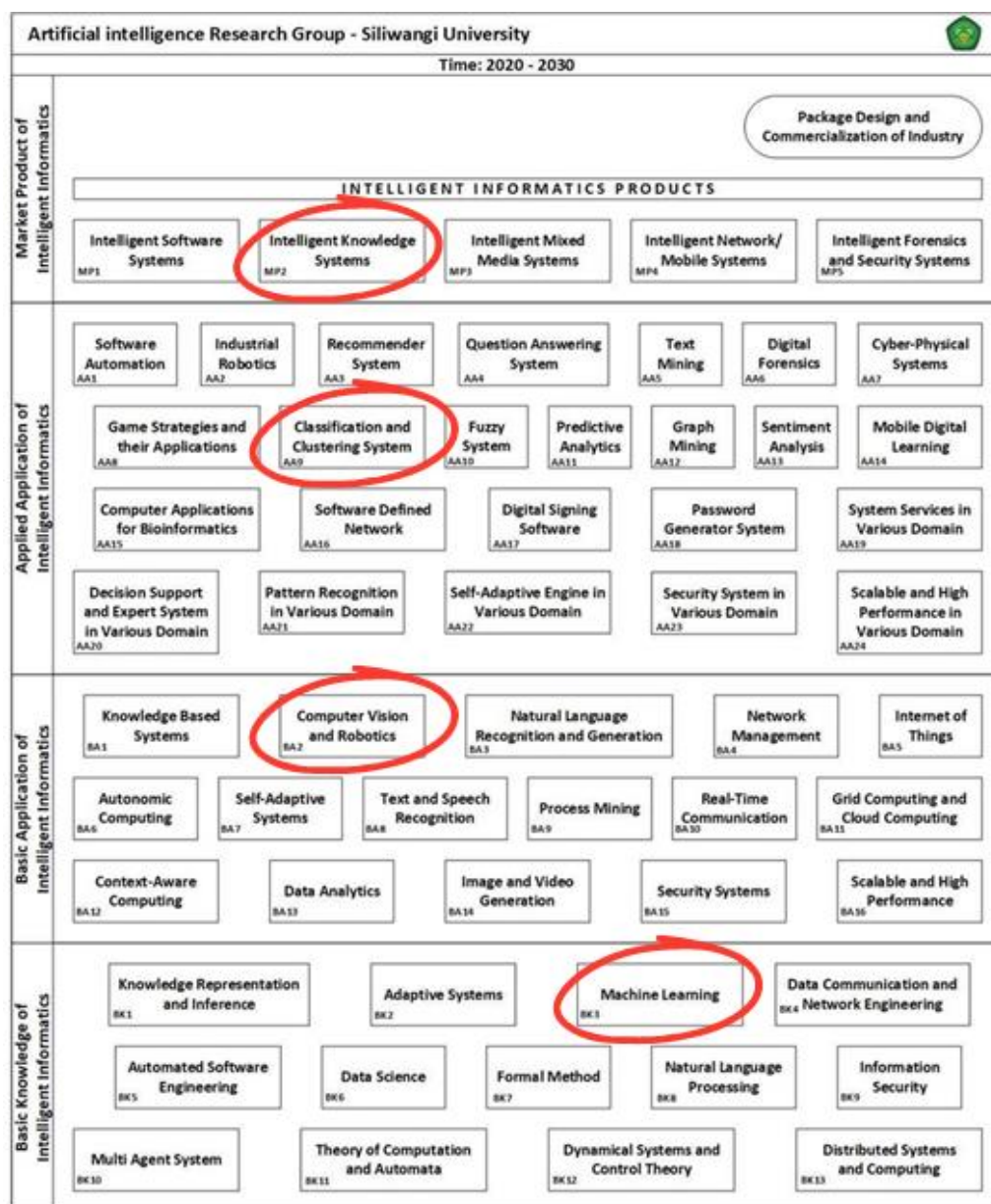
Pada tahap evaluasi, kinerja model akan dibandingkan dengan model yang telah dimodifikasi. Perbandingan ini akan dinilai berdasarkan metrik yang telah ditentukan sebelumnya. Tahap ini akan menghasilkan Bab 4, yang mencakup hasil dan pembahasan.

3.1.6 Komunikasi

Pada tahap ini, laporan akan dipublikasikan di jurnal bereputasi. Laporan tersebut akan disesuaikan dengan peraturan yang ditetapkan oleh jurnal. Akses terhadap laporan ini akan tersedia secara umum atau melalui sistem langganan, sesuai dengan kesepakatan yang telah disetujui.

3.2 Kontribusi *Roadmap* Kelompok Keahlian Informatika dan Sistem Inteligen

Dalam laporan ini berkontribusi pada kelompok keahlian informatika dan sistem inteligen. Adapun *roadmap* kontribusi kelompok keahlian (KK) informatika dan sistem inteligen (ISI) Gambar 3. 6.



Gambar 3. 6 *Roadmap* KK ISI

3.2.1 Basic Knowledge of Intelligent Informatics

Pada bagian pengetahuan dasar informatika cerdas, laporan ini memberikan kontribusi pada bidang pembelajaran mesin, khususnya *deep learning*. Laporan ini mengimplementasikan berbagai arsitektur *deep learning* seperti CNN, *dense*, *maxpool*, *batch normalization*, dan lainnya. Secara khusus, laporan ini berkontribusi pada pengembangan CNN dengan arsitektur VGG, serta integrasi dengan arsitektur *deep learning* lainnya, terutama *batch normalization*, SK, dan CBAM.

3.2.2 Basic Application of Intelligent Informatics

Pada bagian aplikasi dasar informatika cerdas, laporan ini berkontribusi pada bidang visi komputer dan robotika. Laporan ini berfokus pada pengembangan CNN, khususnya dengan arsitektur VGG, serta pendekatan perhatian (*attention*) menggunakan metode CBAM (*Convolutional Block Attention Module*) dan SK (*Selective Kernel*). Selain itu, normalisasi diterapkan dengan metode BN (*Batch Normalization*).

3.2.3 Applied Application of Intelligent Informatics

Laporan ini memberikan kontribusi pada sistem klasifikasi dan pengelompokan dalam bidang informatika cerdas terapan. Hal ini terwujud melalui pengembangan model klasifikasi gambar yang memanfaatkan arsitektur CNN (*Convolutional Neural Network*), BN (*Batch Normalization*), dan mekanisme perhatian (*attention*) dengan metode CBAM (*Convolutional Block Attention Module*) dan SK (*Selective Kernel*).

3.2.4 Market Product of Intelligent Informatics

Laporan ini memberikan kontribusi pada sistem pengetahuan cerdas dalam bidang informatika cerdas terapan, khususnya dalam aspek produk pasar. Hal ini terwujud melalui pengembangan model dengan basis CNN (*Convolutional Neural Network*), khususnya arsitektur VGG, untuk memperkaya ekstraksi fitur pada gambar dengan memanfaatkan mekanisme perhatian (*attention*), khususnya pada metode CBAM (*Convolutional Block Attention Module*) dan mengadaptifkan *kernel* dengan menggunakan SK (*Selective Kernel*). Selain itu, untuk meningkatkan generalisasi dan mempercepat konvergensi model digunakan BN (*Batch Normalization*).