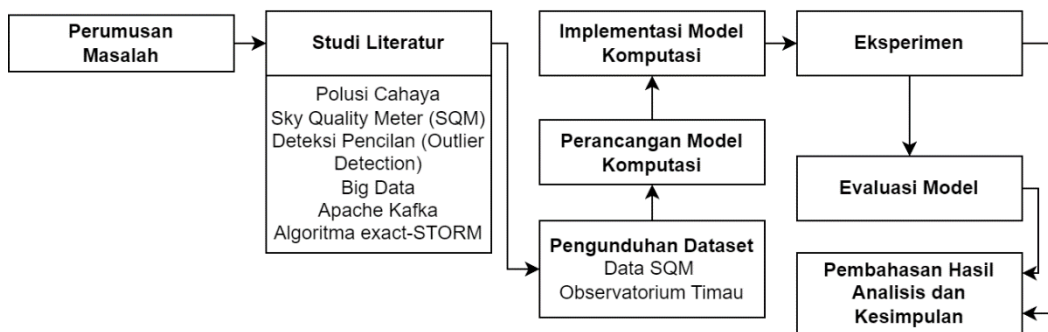


BAB III METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan bagaimana tahap-tahap penelitian untuk menyelesaikan masalah yang telah disampaikan sebelumnya. Bab ini dimulai dari desain penelitian, tahapan-tahapan dalam desain penelitian, metode penelitian, serta alat dan bahan yang digunakan dalam penelitian.

3.1 Desain Penelitian

Desain penelitian merupakan kerangka kerja yang digunakan untuk melakukan penelitian. Pada bagian ini penulis akan memaparkan kerangka kerja terkait penelitian dari memulai penelitian hingga selesai. Desain penelitian digambarkan pada Gambar 3.1.



Gambar 3.1 Desain Penelitian

Terdapat tujuh tahap utama dalam desain penelitian yang dilakukan. Penjelasan tahap-tahap tersebut adalah sebagai berikut:

1. Perumusan Masalah

Tahap ini merupakan tahap awal penelitian. Proses yang terjadi di tahap persiapan yaitu dimulai dari mengidentifikasi masalah yang akan dibahas, kemudian merumuskan masalah, lalu menentukan metode atau algoritma yang akan digunakan untuk menyelesaikan masalah tersebut, dan terakhir menentukan model penelitian untuk membantu penyelesaian masalah.

2. Studi Literatur

Selanjutnya penulis melakukan studi literatur terkait topik yang telah disetujui pada tahap pertama. Studi literatur dilakukan untuk mencari bahan referensi untuk penelitian. Dari hasil studi literatur ini, penelitian menjadi lebih terstruktur untuk mengerjakan setiap tahap penelitian. Adapun studi literatur ini

meliputi polusi cahaya, *Sky Quality Meter* (SQM), *big data*, Apache Kafka, deteksi pencilan (*outlier detection*), dan algoritma exact-STORM. Dalam mempelajari tentang bahasan di atas, penulis mempelajari dari beberapa sumber literatur seperti buku, jurnal, internet, maupun bahan bacaan lainnya yang didapat dari berbagai sumber.

3. Pengunduhan *Dataset*

Tahap ini merupakan tahap bagi penulis mempersiapkan untuk membangun sistem perangkat lunak dengan mengumpulkan data yang akan digunakan. Pengunduhan *dataset* dengan dilakukan dengan mengunduh data hasil pemantauan kecerahan langit malam menggunakan sensor *Sky Quality Meter* (SQM) yang telah dilakukan oleh Priyatikanto (2022) di lokasi Observatorium Timau. Sumber data SQM yang akan digunakan tersebut tersedia pada situs Repositori Ilmiah Nasional (RIN) Dataverse milik Badan Riset dan Inovasi Nasional (BRIN). Data tersebut memiliki rentang waktu dari tahun 2020-2021 dengan ukuran *file* sebesar 22 megabyte.

4. Perancangan Model Komputasi

Tahap ini merupakan tahap persiapan untuk membangun sistem perangkat lunak dengan model komputasi yang telah didesain. Pertama, data *Sky Quality Meter* (SQM) dimasukkan sebagai *input*. Sebelum melakukan proses deteksi pencilan, penulis melakukan preparasi data untuk menyiapkan data yang akan digunakan dalam proses eksperimen atau simulasi. Data yang sudah disiapkan akan diproses untuk kemudian disimulasikan secara *streaming* menggunakan Apache Kafka untuk diterapkan algoritma exact-STORM sehingga menghasilkan pencilan berdasarkan proses deteksi pencilan.

5. Implementasi Model Komputasi dan Eksperimen

Perangkat lunak yang dirancang untuk melakukan deteksi pencilan pada data *Sky Quality Meter* (SQM) kecerahan langit malam dibuat berdasarkan model komputasi yang telah dijelaskan pada tahapan sebelumnya. Pada tahap ini, perangkat lunak dibangun dengan menggunakan bahasa pemrograman Python. Metode pengembangan perangkat lunak yang digunakan adalah metode *Agile*. Berikutnya, akan dilakukan perancangan skenario eksperimen yang akan dilakukan untuk menguji apakah program berjalan sebagaimana mestinya dan

menganalisis hasil *output* dari program yang telah dikembangkan. Pada tahap ini, penulis membuat berbagai skenario eksperimen yang dapat digunakan untuk menguji performa dari model komputasi yang telah diimplementasikan.

6. Evaluasi Model

Tahap ini dapat dilakukan bersamaan dengan proses Eksperimen. Pada tahap evaluasi model, akan dilakukan uji coba pada model komputasi yang telah dibuat dilakukan dengan membandingkan kinerja model dengan parameter yang dibedakan. Lalu, akurasi dan performa model akan dibandingkan dengan performa dari model lainnya. Pada tahap ini pula berbagai informasi yang didapatkan akan disimpan dan digunakan untuk di tahap berikutnya.

7. Pembahasan Hasil Analisis dan Kesimpulan

Pada tahap ini hasil yang telah diproses oleh sistem akan dianalisis dan divalidasi seberapa akurat hasil *output* sistem serta seberapa efisien sistem dalam mengolah dan menampilkan data. Berdasarkan hasil analisis eksperimen, penulis akan menentukan apakah model komputasi pada setiap skenario eksperimen yang telah dirumuskan sudah bekerja dengan semestinya atau tidak. Jika hasilnya belum memuaskan, maka dilakukan evaluasi untuk memperbaiki sistem agar menjadi lebih baik dan sesuai dengan yang diharapkan. Setelah semua tahap sudah dilewati, maka selanjutnya adalah penarikan kesimpulan dari penelitian yang telah dilakukan. Kesimpulan ini didapatkan dari tahap hasil dan analisis. Dokumentasi laporan dan hasil penelitian ini akan dituangkan ke dalam skripsi.

3.2 Metode Penelitian

Adapun metode yang dilakukan dalam penelitian ini dibagi ke dalam dua bagian, yaitu metode pengumpulan data dan metode pengembangan perangkat lunak.

3.2.1 Metode Pengumpulan Data

Penulis berusaha mendapatkan data yang valid dan mampu menunjang penelitian. Adapun metode pengumpulan data pada penelitian ini adalah sebagai berikut:

1. Studi literatur

Studi literatur dilakukan dengan mempelajari teori dan konsep yang menjadi pendukung dalam penelitian ini, yaitu tentang polusi cahaya, *Sky Quality*

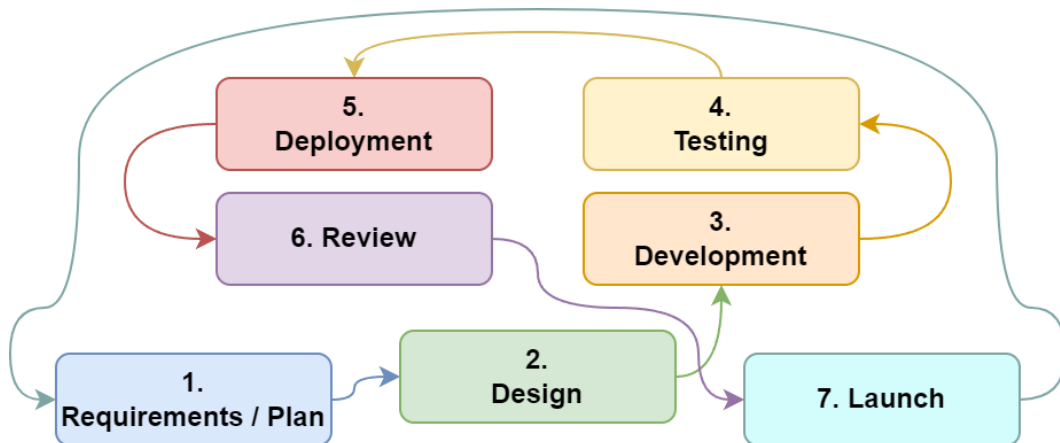
Meter (SQM), data stream, Apache Kafka, deteksi pencilan, dan algoritma exact-STORM.

2. Mendapatkan sekuens data

Metode mendapatkan data pada penelitian ini yaitu dengan mendapatkan data *open-source* dari RIN Dataverse milik BRIN dan lembaga penelitian terkait astronomi.

3.2.2 Metode Pengembangan Perangkat Lunak

Terdapat berbagai macam metode dalam pengembangan perangkat lunak. Untuk metode yang akan digunakan oleh penulis dalam penelitian ini adalah metodologi *Agile* (Beck et al., 2001). Metodologi *Agile* memungkinkan untuk merespons kebutuhan dinamis dari proyek penelitian secara lebih efisien khususnya penelitian dalam ranah *research and development (R&D)*. Penggunaan *Agile* diharapkan dapat meningkatkan fleksibilitas dan adaptasi terhadap perubahan, serta mempercepat siklus umpan balik dan iterasi. Tahapan dari metodologi *Agile* dapat dilihat pada Gambar 3.2.



Gambar 3.2 Metodologi *Agile*

Pada Gambar 3.2 dapat dilihat bahwa dalam metode pengembangan perangkat lunak menggunakan metodologi *Agile* memiliki tujuh tahapan. Tahapan-tahapan ini dapat membantu menyelesaikan proses pengembangan perangkat lunak. Pengertian dari setiap tahapan metodologi *Agile* dijelaskan sebagai berikut:

1. *Requirements / Plan* (Persyaratan / Perencanaan)

Pada tahap ini akan dilakukan pengumpulan seluruh kebutuhan dan perencanaan awal dari proyek yang akan dibuat. Pada tahap ini, akan

difokuskan untuk memahami apa yang dibutuhkan oleh *end-user* dan membuat rencana untuk mengembangkan fitur-fitur yang diperlukan.

2. *Design* (Desain)

Pada tahap ini, akan dikembangkan desain awal dari sistem termasuk arsitektur sistem, model komputasi, dan detail teknis lainnya. Desain harus bersifat fleksibel agar memudahkan apabila ada perubahan selama siklus pengembangan perangkat lunak berlangsung.

3. *Development* (Pengembangan)

Pada tahap ini, kode mulai ditulis untuk membangun sistem atau model yang telah direncanakan dan dirancang. Pengembangan dilakukan dalam iterasi atau *sprint* pendek, memungkinkan bagian kecil dari produk untuk selesai dan diuji dengan cepat.

4. *Testing* (Pengujian)

Setelah fitur selesai dikembangkan, fitur tersebut akan diuji untuk memastikan bahwa fiturnya berfungsi sesuai dengan kebutuhan.

5. *Deployment* (Penerapan)

Setelah pengujian selesai dan berhasil, fitur atau versi dari model sistem akan diterapkan di lingkungan produksi.

6. *Review* (Peninjauan)

Pada tahap ini, akan dilakukan peninjauan ulang terhadap pekerjaan yang telah selesai untuk mengevaluasi hasil dan proses. *Feedback* dari *end-user* dan ahli akan dikumpulkan untuk membuat perbaikan yang lebih lanjut.

7. *Launch* (Peluncuran)

Tahap ini melibatkan peluncuran resmi produk atau fitur kepada *end-user*. Peluncuran dapat dilakukan bertahap atau sekaligus, tergantung kepada strategi yang dipilih.

Metodologi *Agile* memiliki sifat yang adaptif dan inkremental. *Agile* memegang prinsip pada *feedback* yang berkelanjutan, setiap tahap dapat kembali ke tahap sebelumnya untuk perbaikan dan penyesuaian. Selain itu, pada setiap iterasi atau *sprint* mencakup sebagian kecil dari fitur yang sudah selesai dibuat, sehingga memungkinkan adanya siklus yang diulang atau inkremental sampai seluruh fitur yang diperlukan selesai dikembangkan dan sesuai dengan kebutuhan

pengguna. Dengan menggunakan metodologi *Agile*, dapat dipastikan bahwa pengembangan perangkat lunak dapat beradaptasi dengan perubahan kebutuhan dan *feedback* dari pemangku kepentingan, sehingga produk yang dihasilkan lebih dipersonalisasi sesuai dengan harapan *end-user*.

3.3 Alat dan Bahan Penelitian

Bagian ini menjelaskan secara detail alat dan bahan yang digunakan untuk melakukan penelitian. Dalam penelitian ini, alat yang digunakan untuk menunjang kebutuhan selama penelitian adalah sebagai berikut:

1. Perangkat keras (*hardware*) yaitu laptop, dengan spesifikasi sebagai berikut:
 - Processor AMD Ryzen 7 5800U with Radeon Graphics 1.90 GHz
 - RAM 16 GB
 - HardDrive 512 GB SSD
 - Graphics card GeForce RTX 3050 Ti Mobile 4GB
2. Perangkat lunak (*software*) sebagai berikut:
 - Sistem Operasi Windows 11 Home Single Language
 - Jupyter Notebook
 - Visual Studio Code
 - Apache Kafka
 - Python 3.10
 - Google Spreadsheet
 - Figma
 - draw.io
 - Microsoft Word

Adapun bahan yang diperlukan untuk melakukan penelitian adalah data hasil data hasil rekaman *Sky Quality Meter* (SQM) yang didapatkan RIN Dataverse. Data tersebut memiliki rentang waktu bervariasi dari tahun 2020-2021. Data tersebut memiliki format *.dat* serta dapat diperoleh secara bebas dan *open-source* pada situs <https://data.brin.go.id/>.