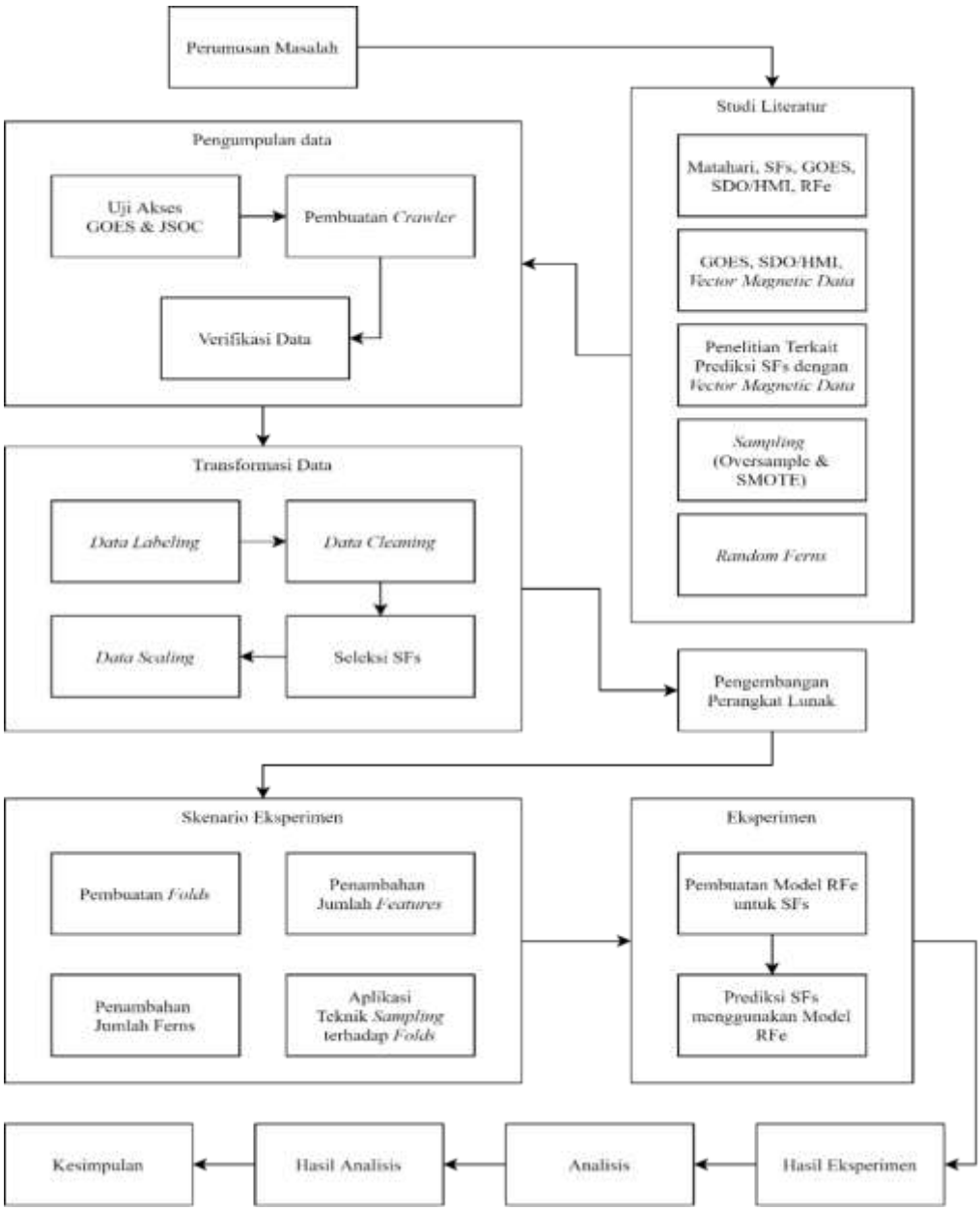
BAB III
METODOLOGI PENELITIAN

3.1 Desain Penelitian

Subbab ini akan memaparkan rangkaian tahapan dalam penelitian yang telah penulis rancang. Pada Gambar 3.1 diilustrasikan desain penelitian dimulai dari tahapan awal penelitan hingga penarikan kesimpulan.



Gambar 3.1 Ilustrasi desain penelitian yang digunakan dalam penelitian.

1. Pada tahap pertama, penulis melakukan diskusi untuk memahami latar belakang dari topik yang penulis ambil dan merumuskan masalah, tujuan, serta batasan sehingga terbentuknya fokus penelitian. Hal ini terangkum dalam bab pertama dari penelitian ini.
2. Tahapan selanjutnya adalah studi literatur, di mana pada tahap ini penulis melakukan pendalaman materi sehingga pemahaman akan topik yang penulis ambil menjadi lebih komprehensif. Secara garis besar, materi yang penulis dalami merupakan fokus dari dua rumpun keilmuwan yang berbeda, yaitu *Solar Physics* dan *Machine Learning*.
3. Setelah memahami literasi terkait fokus penelitian, penulis kemudian melakukan tahap pengumpulan data SFs. Data SFs yang penulis gunakan berasal dari beberapa sumber yang berbeda, yaitu rekam data Satelit GOES sebagai sumber data katalog SFs *flux events* dan instrument HMI pada Satelit SDO untuk rekam *vector magnetic* yang terjadi di matahari. Kedua sumber data ini memiliki spesifikasi tersendiri terkait bagaimana alur dan parameter yang dibutuhkan untuk dapat mengakses data mereka. Namun, terdapat irisan yang dapat menjembatani penggabungan data dari kedua sumber tersebut, yaitu NOAA *Active Region Numbers* dan waktu puncak terjadinya suatu aktivitas di matahari. Untuk menghindari adanya kesalahan dalam penggabungan kedua sumber tersebut, maka penulis mengadaptasi dan membentuk *crawler* untuk mengautomasi proses pengumpulan data. Penulis kemudian melanjutkan tahapan pengumpulan data ini dengan melakukan verifikasi manual dan acak dari *instances* yang telah dikumpulkan.
4. Data yang telah dikumpulkan dan diverifikasi kemudian disebut sebagai *raw data*. Hal ini dikarenakan adanya beberapa *missing data* yang terekam dan dapat merusak hasil penelitian. Penulis kemudian melakukan pra proses terhadap data secara semi *automated*, yaitu dengan mengurutkan *missing data* dan menghapusnya secara manual dengan tetap mempertimbangkan kaidah keilmuwan terkait SFs. Penulis juga melakukan penghapusan terhadap kolom-kolom yang tidak digunakan dalam penelitian. Data kemudian dilabeli berdasarkan huruf dari kelas SFs yang terjadi, semisal X5.0 menjadi X saja. Setelah itu, sedikit berbeda dengan (Liu et al., 2017) yang mengambil kejadian

terakhir SF pada suatu hari – jika terdapat lebih dari satu SF di hari tersebut – penulis melakukan seleksi data SFs dengan cara mengambil kelas terbesar yang muncul di suatu hari jika terdapat lebih dari satu SF. Penulis kemudian melakukan *labeling* kembali untuk membentuk dua *SFs datasets*, yaitu *multiclass dataset* dengan kelas B, C, M, dan X serta *binary dataset* dengan kelas B/C dan M/X.

5. Penulis kemudian melanjutkan tahapan penelitian dengan membuat perangkat lunak. Perangkat lunak dimaksud merupakan implementasi dari RFe dengan fungsi pembuatan model serta prediksi dengan menggunakan bahasa R. Perangkat lunak ini juga dilengkapi beberapa fungsi pendukung tambahan seperi fungsi untuk mengautomasi pembuatan *folds*, fungsi *models and results writter*, penghitung jumlah eksperimen, dan beberapa fungsi lain untuk mendukung penelitian ini.
6. Sebelum memulai eksperimen, penulis terlebih dahulu merancang skenario eksperimen. Skenario ini diharapkan dapat menjadi bentuk *tuning* dari pemodelan yang akan dilakukan. Penulis menggunakan penambahan pada jumlah *features* dan *ferns* sebagai *iterator* bersama dengan *folds* dan skema *sampling* untuk menangani *imbalanced data*. Sebagai ilustrasi, dengan menggunakan rentang 3-10 *features* (kelipatan satu), 200-1200 *ferns* (kelipatan 200), 5 *folds* (diulang sebanyak 10 kali), serta implementasi satu teknik *sampling oversample*, maka akan terdapat $8 \times 5 \times 10 \times 6$ atau 2400 eksperimen.
7. Skenario eksperimen yang telah dibuat kemudian dijalankan pada perangkat lunak. Seperti alur prediksi pada bidang ML lainnya, penulis membuat model dengan menggunakan training data dan mengujinya dengan testing data. Selain melakukan kedua hal tersebut, pada tahap ini perangkat lunak yang penulis kembangkan juga merekam atau menyimpan seluruh model dan hasil keluaran dari RFe.
8. Hasil eksperimen keluaran perangkat lunak dipisah berdasarkan jumlah *ferns* yang digunakan. Penulis juga membaginya menjadi dua tipe keluaran, yaitu keluaran yang besifat spesifik untuk model dan hasil serta keluaran yang bersifat merangkum keseluruhan eksperimen pada jumlah *ferns* tertentu.

Keluaran yang bersifat spesifik ini ditujukan untuk penggunaan kembali model yang telah dibuat ke dalam tahap prediksi.

9. Keluaran perangkat lunak yang bersifat rangkuman ini kemudian dianalisis dengan cara *plotting* data ke dalam diagram grafik. Penggunaan grafik ini memudahkan penulis untuk menarik kesimpulan dan evaluasi terhadap eksperimen yang telah dilakukan.
10. Tahap terakhir adalah tahap di mana penulis menuliskan laporan hasil dari penelitian yang telah dilakukan.

3.2 Alat dan Bahan Penelitian

Subbab ini menjelaskan secara spesifik komponen perangkat keras, perangkat lunak, dan *dependencies* yang digunakan dalam penelitian.

3.2.1 Alat Penelitian

1. Perangkat Keras berupa *personal computer* dengan spesifikasi:
   - CPU Intel® Core™ i7-8750H @CPU 2.20GHz Turbo Boost 4.10GHz, 6 *Cores*, 12 *Threads*
   - *Memory* sebesar 24GB
     - 16GB Micron, DDR4-2666 PC4-21300 DDR4 SDRAM SO-DIMM
     - 8GB SK Hynix, DDR4-2666 PC4-21300 DDR4 SDRAM SO-DIMM
   - V-NAND SSD 970 EVO NVMe M.2 Samsung 250GB
   - GPU NVIDIA GeForce GTX 1050 Ti 4GB GDDR5-SDRAM 128-bit
   - Monitor 15.6” (1920 × 1080) 120Hz
   - *Keyboard* Logitech K120
   - *Mouse* Rexus VR3 Warfaction
2. Perangkat Lunak
   - Microsoft® Windows 10 Home Single Language
   - Microsoft® Word for Microsoft 365 MSO 32-bit
   - Microsoft® Excel® for Microsoft 365 MSO 32-bit
   - R 3.5.3
     - *xlsx* 0.6.1

    - *caret* 6.0-85
    - *rjson* 0.2.20
    - *ggplot2* 3.1.0
    - *beepr* 1.3
- RStudio 1.2.5033 "*Orange Blossom*"
- *python* 3.7.1
    - *sunpy* 1.1.3
        - *sunpy.instr.goes*
        - *sunpy.time*
    - *pandas* 0.23.4
    - *requests* 2.21.0
    - *urllib*
    - *json* 2.0.9
    - *time*
    - *datetime*
- Anaconda Navigator 1.9.6
- Spyder 3.3.2
- Visual Studio Code 1.46.1
- Diagrams.net (draw.io) 13.3.5

3.2.2 Bahan Penelitian

Bahan yang diperlukan untuk melakukan penelitian terdiri dari data katalog SFs *flux events* yang direkam oleh Satelit GOES diambil melalui *library sunpy* dalam *sunpy.instr.goes* untuk periode Mei 2010-Desember 2019 dan penambahan secara manual SFs *flux events* untuk periode Januari 2020-Mei 2020 yang bersumber dari https://www.spaceweatherlive.com/. Data ini kemudian diolah agar dapat menjadi parameter yang valid untuk mengambil produk data *vector magnetic* Instrumen HMI pada Satelit SDO melalui http://jsoc.stanford.edu/. Selain itu penelitian ini juga menggunakan berbagai buku, jurnal, *repository*, artikel, serta video yang berkaitan untuk menunjang wawasan penulis terhadap fokus penelitian.

3.3 Metode Penelitian

Pada subbab ini dijabarkan bagaimana metode yang penulis gunakan dalam memahami, mengumpulkan data, serta merancang alur perangkat lunak untuk menangani kasus prediksi SFs dengan menggunakan RFe.

3.3.1 Metode Literasi

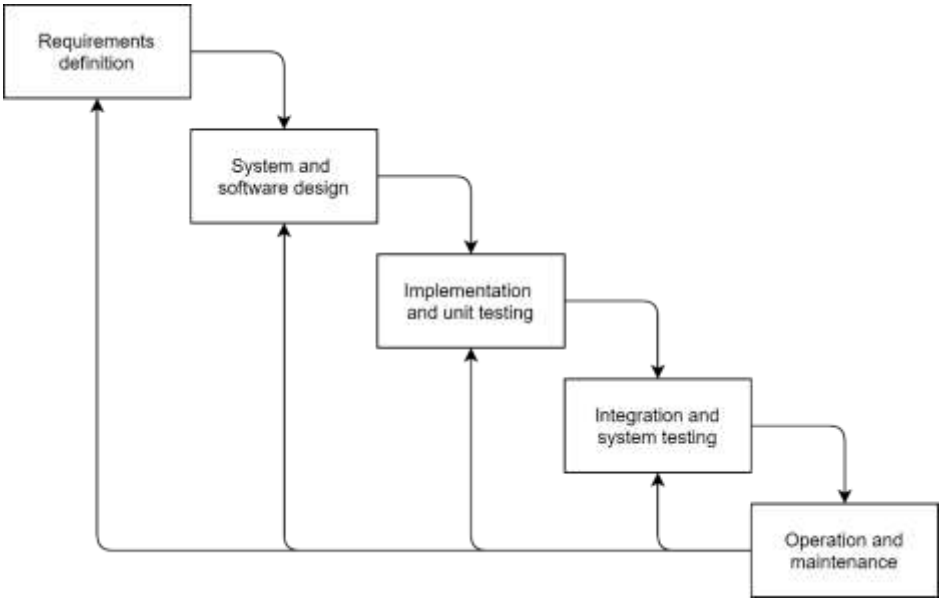
Pemahaman akan topik penelitian merupakan faktor fundamental karena digunakan sebagai pedoman untuk menyelesaikan tahap demi tahap suatu penelitian. Dalam hal ini penulis melakukan studi terhadap literasi penelitian SFs terdahulu beserta turunan maupun akar dari SFs itu sendiri. Selain itu, penulis juga mempelajari lebih jauh tentang teknik komputasi yang dirasa mampu dan sesuai untuk diimplementasikan pada kasus SFs ini. Oleh karena itu, penulis melakukan studi literatur untuk topik matahari, siklus matahari, aktivitas matahari, SFs, ML, klasifikasi, NB, RFe, serta *imbalance data*.

3.3.2 Metode Pengumpulan Data

Data SFs yang digunakan merupakan data gabungan dari beberapa sumber penyedia data yang memiliki peraturan khusus mengenai bagaimana suatu data dapat diambil. Oleh karena itu, penulis membuat *crawler* untuk menangani secara spesifik peraturan tersebut serta mengautomasi proses penggabungan data. Penulis menggunakan NOAA AR *Numbers*-HARP *Numbers* yang tersedia pada sumber-sumber data sebagai jembatan atau penghubung sumber-sumber tersebut.

3.3.3 Metode Pengembangan Perangkat Lunak

Penelitian ini menggunakan model pengembangan perangkat lunak *waterfall*. Di mana pengembangan perangkat lunak dengan model ini diawali dengan tahap penentuan *requirements definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, serta *operation and maintenace* seperti diilustrasikan pada Gambar 3.2.



Gambar 3.2 Model pengembangan perangkat lunak *waterfall* yang diadaptasi dari (Sommerville, 2016).

Pada praktiknya, tidak ada model proses universal dalam pengembangan suatu perangkat lunak. Artinya, penyesuaian bisa saja dilakukan tergantung dengan kebutuhan. Sebagai contoh, penulis bisa saja tidak sepenuhnya menggunakan sekuens seperti pada Gambar 3.2. Hal ini dikarenakan adanya kemungkinan pengetahuan baru yang penulis dapatkan seiring dengan pembuatan perangkat lunak yang berimplikasi pada terminasi pada sekuens tertentu. Namun demikian, penulis tetap menggunakan pendekatan *Waterfall Model* sebagai kerangka dari keseluruhan proses pengembangan perangkat lunak dalam penelitian ini. Adapun penjelasan dari setiap komponen pada *Waterfall Model* yang penulis adaptasi dari (Sommerville, 2016) adalah sebagai berikut:

1. *Requirements analysis and definition*, merupakan tahap di mana batasan dan tujuan ditentukan untuk kemudian didefinisikan sebagai spesifikasi dari sistem.
2. *System and software design*, merupakan tahapan untuk menggambarkan abstraksi dan *relationship* dari perangkat lunak. Termasuk pada tahap ini juga dialokasikan perangkat keras dan perangkat lunak pendukung sebagai basis arsitektur dari perangkat lunak yang akan dibangun.
3. *Implementation and unit testing*, pada tahap ini rancangan dan spesifikasi yang sudah ada diimplementasikan ke dalam bentuk program atau unit program.

Selanjutnya dilakukan pengujian atas kesesuaian dari program yang telah dibuat dengan spesifikasi yang telah ditentukan.

4. *Integration and system testing*, di tahap ini setiap program atau unit program diintegrasikan menjadi suatu sistem yang paripurna. Kemudian dilanjutkan dengan pengujian menyeluruh pada sistem.
5. *Operation and maintenance*, merupakan tahapan di mana sistem digunakan dalam *production level*. Kendala yang ditemukan pada tahap ini akan diperbaiki dengan cara iterasi pada sekuens tertentu dari keseluruhan proses pengembangan perangkat lunak.