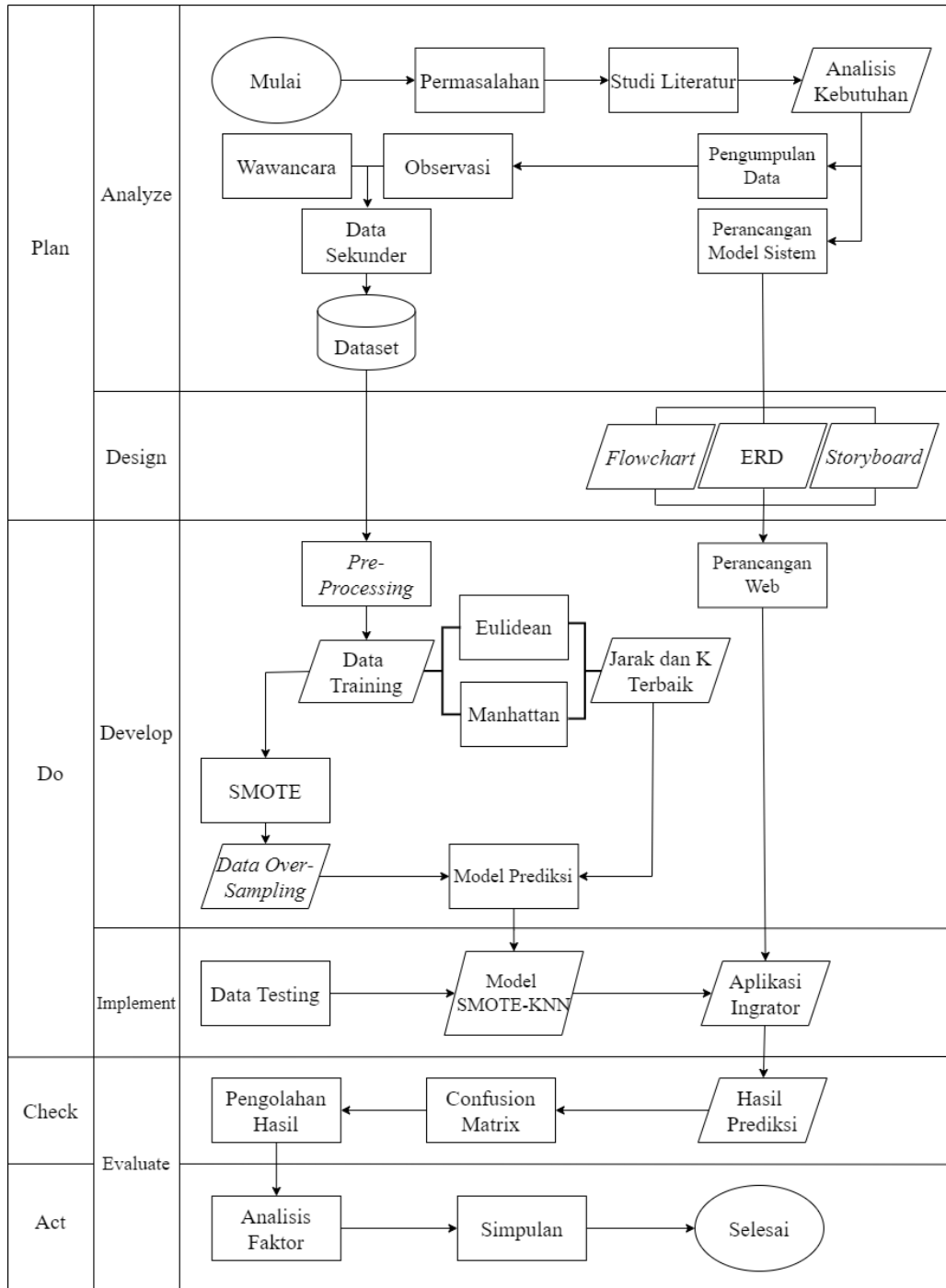


BAB III METODOLOGI PENELITIAN

Metodologi penelitian ini menggunakan *Smart Learning Environment Establishment Guideline (SLEEG)* berdasarkan *Analyze-Design-Develop-Implement-Evaluate (ADDIE)* (Rosmansyah et al., 2022).



Gambar 3. 1 Alur Metodologi Penelitian

Penjelasan terkait gambar diatas adalah sebagai berikut:

Andika Putra Kamula, 2022
**PEMBANGUNAN MODEL KLASIFIKASI SMOTE - KNN
 (SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE -
 K-NEAREST NEIGHBOR) DALAM MEMPREDIKSI WAKTU KELULUSAN SISWA BOOTCAMP BINAR
 ACADEMY**

3.1. Analyze

Pada tahap ini penelitian dimulai dengan melakukan kajian riset dan menentukan kebutuhan yang akan digunakan selama penelitian. Berdasarkan permasalahan yang telah diangkat, pertama terkait EDM pada bidang prediksi, berikutnya pemilihan algoritma klasifikasi yakni KNN, dan terakhir pembuatan sistem. Pada tahap ini pula dilakukan pemilihan objek penelitian yakni siswa *bootcamp* Binar Academy pada *course Full Stack Web Development* (FSW). Proses pengumpulan data dilakukan dengan wawancara dan observasi.

Wawancara dilakukan bersama tim AA selaku penanggungjawab siswa Binar Academy dari *silver level* hingga *graduation*. Proses ini dilakukan dengan tanya jawab mengenai hal-hal yang dibutuhkan dalam penelitian, untuk mencari informasi terkait faktor lain yang menjadi tolok ukur dalam menentukan kelulusan tepat waktu atau tidak tepat waktu.

Berdasarkan kesepakatan dengan pihak Binar Academy bahwa data bersifat rahasia, maka sebelum diberikan semua data yang dibutuhkan, terlebih dahulu dilakukan tanda tangan surat pernyataan terkait asas kerahasiaan data. Instrumen wawancara yang digunakan dapat dilihat pada lampiran 3.

Populasi merupakan keseluruhan dari kumpulan unit yang akan diteliti karakteristiknya, sedangkan sampel adalah sub unit dari populasi, memiliki sifat representative dan mencakup seluruh elemen dalam populasi (Ode et al., 2015). Populasi dalam penelitian ini adalah siswa bootcamp course Full Stack Web Development (FSW) dari wave 14 hingga 18.

Berdasarkan algoritma KNN bahwa data dibagi atas dua jenis yakni training dan testing, dari 68 data yang diperoleh dari Binar Academy 80% diantaranya adalah data training dan 20% data testing. Sampel dalam penelitian ini diambil menggunakan *simple random sampling*.

3.2. Design

Pada tahap *design* melaksanakan proses perancangan sistem. Perancangan sistem dimulai dengan membuat *flowchart*, kemudian *Entity Relationship Diagram* (ERD) untuk kebutuhan database, dan terakhir *storyboard*. *Flowchart* berdasarkan alur penggunaan sistem dari awal hingga

akhir, proses login, pemilihan menu, serta proses CRUD yang berjalan di dalamnya, diharapkan pengguna lain bisa paham alur pembuatan dan penggunaan sistem. *Flowchart* dibuat menggunakan *tools online diagram.net* dari *google*. EDR menampilkan semua entitas yang digunakan dalam *database* sistem, mulai dari *login*, data *training*, data *testing*, hasil perhitungan, dan lainnya, dalam ERD di tampilkan relasi antar entitas, diharapkan bisa dengan mudah memahami alur data dalam sistem. ERD dibuat menggunakan *tools Navicat Premium* versi 15.0.2. *Storyboard* menampilkan *user interface* sistem, mulai dari halaman *login*, *Create*, *Read*, *Update*, *Delete* (CRUD) pada data, halaman, data *training*, halaman data *testing*, hal ini bertujuan memberikan gambaran sistem yang akan dibuat nantinya. *Storyboard* ini dibuat menggunakan *tools Balsamiq Wireframes* versi 4.5.4.

3.3. Develop

Dataset yang telah di berikan oleh Binar Academy dilanjutkan ke tahap *develop* dimulai dengan *pre-processing* data melalui empat proses, pertama pengumpulan data, kedua pembersihan data, ketiga pemilihan data, dan terakhir perubahan data. Pengumpulan data mengkategorikan data berdasarkan atribut yang telah ditetapkan pada tahap analisis, setelah semua data dikumpulkan dilakukan proses pembersihan data, terdapat dua pilihan dalam permbersihan yakni menghilangkan data kosong atau menambah data berdasarkan mean, media, atau modus dari keseluruhan nilai dalam satu atribut data. Berikutnya dilakukan pemilihan data, pemilihan ini bertujuan melihat data mayoritas dan minoritas, sehingga bisa di tentukan metode berikutnya untuk mengatasi data tidak seimbang. Proses terakhir perubahan data, hal ini jika dalam data terdapat nilai dari atribut yang belum sama atau satu bersifat *string*, dan lainnya bersifat numerik.

Setelah proses *pre-processing* selesai dilanjutkan ke dalam proses *mining*, dalam penelitian ini akan dilakukan dua kali proses *mining* menggunakan *tools mining Rapid Miner* versi 9.10.10 (Rahayuningsih et al., 2018) yakni pertama melakukan *mining* terhadap algoritma KNN dengan menggunakan dua metode perhitungan jarak yakni, *Eulidean* dan *Manhattan* (Setiawan et al., 2022) (Hidayati et al., 2021) (Kesuma et al., 2020) diuji

dengan parameter K 1, 3, 5, 7, dan 9, terhadap data tanpa menggunakan metode SMOTE. Hasil akhir *mining* pertama akan berupa tingkat akurasi dari dua metode perhitungan jarak dengan semua parameter K yang telah disebutkan, di dapatkan hasil *accuracy*, *precision*, dan *recall*.

Mining kedua diterapkan metode SMOTE terlebih dahulu untuk mengatasi data tidak seimbang (Qiong et al., 2016). Berikutnya di dapat *dataset oversampling*, hasil data ini akan diimplementasikan dengan metode jarak dan parameter K yang sama dengan proses *mining* pertama. Selanjut akan dilakukan perhitungan nilai *accuracy*, *precision*, dan *recall*. Dari kedua hasil tersebut akan dianalisis perbedaan nilai *accuracy*, *precision*, dan *recall*, untuk mendapatkan data *training* yang cocok diimplementasikan ke dalam sistem.

Tahap ini juga dimulainya proses perancangan sistem. Sistem akan dibuat menggunakan *framework Codeigniter* (CI) menggunakan template sb admin. Berikutnya diberi nama dengan *In Time Predictor Graduation* (Ingrator).

3.4. Implement

Data *training* yang telah di dapat setelah melakukan proses *mining* pada tahap *develop* menjadi sebuah model prediksi dengan algoritma KNN menggunakan metode perhitungan jarak dan parameter K terbaik berdasarkan nilai *accuracy*, *precision*, dan *recall*, yang telah digabungkan dengan metode SMOTE untuk mengatasi data tidak seimbang. Data *training* inilah yang akan diimplementasikan ke dalam aplikasi *ingrator*. Berikutnya akan dilakukan testing data baru menggunakan *Ingrator* untuk melihat hasil implementasi model dan sistem.

3.5. Evaluate

Hasil prediksi akan dianalisis untuk dengan langkah-langkah sebagai berikut:

- 1) Menggunakan *Confusion Matrix* untuk menentukan kemampuan model prediksi berdasarkan jumlah data yang terprediksi benar dan salah terhadap aktual data (Luque et al., 2019). *True Positive* (TP) adalah nilai banyak data positif yang terprediksi benar oleh model prediksi. *True Negative* (TN) adalah nilai banyak data negatif yang terprediksi benar oleh model prediksi.

False Positive (FP) adalah nilai banyak data negatif yang terprediksi salah oleh model prediksi. *False Negative* (FN) adalah nilai banyak data positif yang terprediksi salah oleh model prediksi.

Tabel 3. 1 *Confusion Matrix*

Klasifikasi	Aktual = Tepat Waktu	Aktual = Tidak Tepat Waktu
Prediksi = Tepat Waktu	<i>True Positive</i> (<i>TP</i>)	<i>False Positive</i> (<i>FP</i>)
Prediksi = Tidak Tepat Waktu	<i>False Negatif</i> (<i>FN</i>)	<i>True Negative</i> (<i>TN</i>)

- a. Pertama, nilai yang dicari adalah nilai *accuracy*. Nilai *accuracy* adalah nilai ukur yang menentukan kemampuan model prediksi memprediksi data dengan benar terhadap keseluruhan data. Rumus yang digunakan adalah sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Rumus 3. 1 Mengitung Nilai *Accuracy* Model

- b. Nilai kedua yang dicari adalah nilai *precision*. Nilai *precision* adalah nilai ukur yang menentukan kemampuan model prediksi dalam memprediksi data dengan benar tanpa salah mengklasifikasikan data. Rumus yang digunakan adalah sebagai berikut:

$$Precision = \frac{TP}{TP + FP}$$

Rumus 3. 2 Mengitung Nilai *Precision* Model

- c. Ketiga adalah nilai *recall*. Nilai *recall* adalah nilai ukur yang menentukan kemampuan model prediksi dalam memprediksi data dengan benar sebanyak-banyaknya sesuai data aktual yang ada. Rumus yang digunakan adalah sebagai berikut:

$$Recall = \frac{TP}{TP + FN}$$

Rumus 3. 3 Mengitung Nilai *Recall* Model