BAB II

LANDASAN TEORI

2.1 Landasan Teori

2.1.1 Allo-HCT

Hematopoietic cell transplantation (HCT) merupakan pilihan pengobatan bagi pasien dengan gangguan hematologi ganas (*malignant*) atau non-ganas (*nonmalignant*) agar dapat kembali menghasilkan sistem kekebalan tubuh yang normal. HCT dibedakan menjadi *autologous* HCT (auto-HCT) dan *allogeneic* HCT (allo-HCT) dengan mengacu kepada sumber sel hematopoietik. Auto-HCT berarti menggunakan sel milik pasien sendiri, sedangkan allo-HCT menggunakan sel hematopoietik dari pendonor dengan *human leukocyte antigen* (HLA) yang kompatibel (Khaddour dkk., dalam Saad dkk., 2020).

Terdapat tiga sumber donor potensial, yaitu donor yang memiliki hubungan darah, sukarelawan tanpa hubungan keluarga, dan unit UCB (*Umbilical Cord Blood*). Rejimen persipan (*preparative regimen*) diberikan sebelum allo-HCT untuk menghilangkan sisa sel ganas dan menekan sistem kekebalan tubuh penerima, agar pencangkokan sel dari donor dapat dilakukan dan mencegah penolakan cangkok (Saad dkk., 2020). Berbagai gangguan hematologi yang terbukti berhasil ditangani melalui allo-HCT antara lain AML, ALL, MDS, *chronic myeloid leukemia*, *chronic lymphocytic leukemia*, *multiple myeloma*, *primary and secondary myelofibrosis*, *Hodgkin lymphoma*, dan *non-Hodgkin lymphoma*.

2.1.2 *Event-free Survival* (EFS)

Menurut *National Cancer Institute*, *event-free survival* (EFS) merupakan rentang waktu dimana pasien bebas terhadap komplikasi ataupun kejadian yang ingin dicegah atau ditunda oleh pengobatan. EFS dalam konteks HCT meliputi *relapse*, progresi penyakit, atau kematian dengan penyebab apapun. Status EFS seringkali digunakan dalam mengevaluasi *Overall Survival* (OS). OS didefinisikan sebagai waktu kelangsungan hidup total sejak HCT hingga terjadi kematian. Pasien yang tidak mengalami kejadian hingga akhir pengamatan akan dianggap "*censored*" (Fujimoto dkk., 2021). EFS berkaitan erat dengan morbiditas dan mortalitas. Morbiditas berarti keadaan bergejala atau tidak sehat pada suatu kondisi, sedangkan mortalitas mengacu kepada kematian yang disebabkan oleh suatu peristiwa (Bien dkk., 2022). *Graft versus host disease* (GVHD) menjadi penyebab morbiditas dan mortalitas yang signifikan, dimana terdapat komplikasi utama pasca HCT yang berdampak terhadap kualitas hidup pasien (de Vere Hunt dkk., 2021).

2.1.3 Analisis Survival

Analisis survival adalah metode statistika yang digunakan dalam menganalisis data waktu hingga terjadinya suatu peristiwa (*time-to-event data*) (Klein dkk., dalam Rai dkk., 2021). Data survival mencakup waktu kejadian dan berbagai variabel independen yang terkait dengan peristiwa tersebut. Tujuan utama dari analisis survival antara lain menganalisis pola waktu, mengevaluasi penyebab *censored data*, membandingkan kurva survival dan menilai hubungan antar variabel (Rai dkk., 2021).

Teknik yang umum digunakan dalam analisis survival, yaitu Kaplan-Meier (K-M). Metode K-M akan menghitung probabilitas bertahan hidup pada berbagai titik waktu dan menghasilkan kurva survival, termasuk nilai *censored* (Indrayan & Tripathi, 2022). Informasi minimal yang dibutuhkan untuk membentuk kurva survival dari K-M (D'Arrigo dkk., 2021) adalah *time-to-event* (t_1, t_2, ..., t_j dengan $j \leq n$) serta variabel status biner (1 jika peristiwa terjadi dan 0 jika *censored*). Probabilitas bertahan hidup pada waktu tertentu (t_j) dihitung berdasarkan jumlah individu yang masih hidup setelah waktu t_j dan jumlah individu berisiko sebelum waktu t_j. Rumus probabilitas survival terdapat pada Persamaan 2.1

$$\widehat{p_j} = \frac{n_j - d_j}{n_j} \tag{2.1}$$

di mana:

$\widehat{p_j}$ = probabilitas survial pada waktu t_j
n_j = jumlah individu yang masih hidup sebelum waktu t_j
d_j = jumlah individu yang mengalami kejadian pada waktu t_j

Perkalian semua probabilitas survival akan menghasilkan probabilitas survival kumulatif hingga waktu t_j. Probabilitas survival kumulatif ditunjukkan dengan simbol *S(t)* dan $\prod$ melambangkan operasi perkalian. Perhitungan probabilitas kumulatif terdapat pada Persamaan 2.2

$$\widehat{S(t)} = \prod_{j/t_{j \leq t}} \widehat{p_j} \tag{2.2}$$

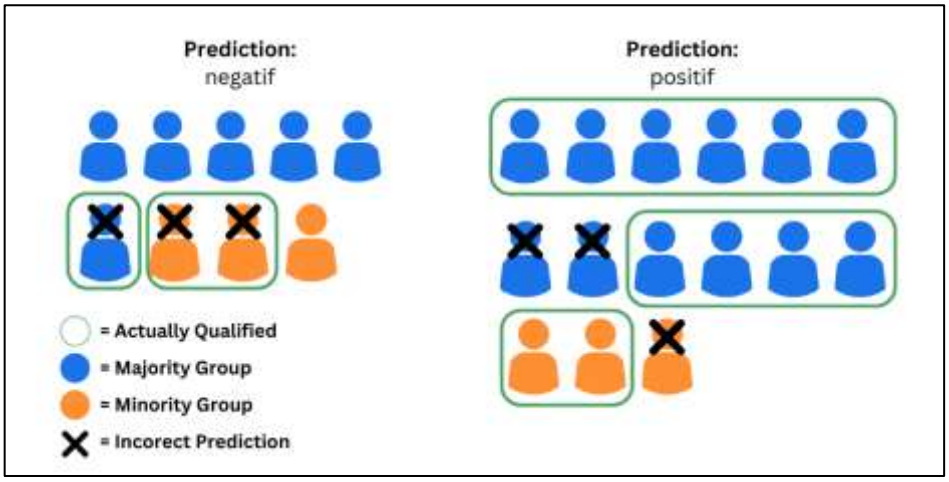
di mana:

$\widehat{S(t)}$ = probabilitas survial kumulatif
$\prod_{j/t_{j \leq t}}$ = perkalian untuk semua *j*, di mana waktu kejadian $t_{j \leq t}$

2.1.4 *Fairness* dalam *Machine Learning*

Fairness dalam *machine learning* didefinisikan sesuai dengan pengukuran spesifik pada konteks tertentu, sehingga tidak ada model yang adil menurut semua jenis pengukuran. *Fairness* dari perspektif komputasi dapat diartikan sebagai kondisi dimana hasil prediksi model tidak menunjukkan diskriminasi terhadap atribut sensitif. *Fairness* berbeda dengan bias karena berfokus pada hasil yang tidak diskriminatif, sedangkan bias mengacu kepada kesalahan sistematis yang memengaruhi performa model untuk kelompok berbeda (Wan dkk., 2023).

Sebuah model dapat memiliki kinerja yang baik, tetapi memerlukan pemeriksaan dampak model pada berbagai kelompok. Pemisahan demografi dan evaluasi pada setiap kelompok dapat mengidentifikasi adanya kemungkinan ketidakadilan. Seringkali ditemukan bahwa prediksi "positif" lebih banyak diterima oleh kelompok mayoritas daripada kelompok minoritas. Gambar 2.1 mengilustrasikan contoh ketidakadilan yang diberikan oleh model.



Gambar 2. 1 Ilustrasi Ketidakadilan dalam Model Prediksi *Machine Learning*

2.1.5 *Fairness Regression*

Fairness regression merupakan metode yang bertujuan untuk mengatasi ketidakadilan dalam model *machine learning* terhadap tugas regresi. *Fairness* dalam regresi lebih kompleks dibandingkan dengan klasifikasi karena label regresi bersifat kontinu. FaiReg (Mohamed & Schuller, 2022) bekerja dengan menormalisasi label berdasarkan atribut terlindungi untuk menghilangkan bias pelabelan. Normalisasi dilakukan dengan menghitung statistik label berdasarkan variabel terlindungi yang akan melakukan transformasi label ke dalam set *fair labels* ($\hat{y}_1$, ..., $\hat{y}_n$). Rumus transformasi label ke bentuk *fair labels* terdapat pada Persamaan 2.3

$$\hat{y}_i = \frac{y_i - \mu_{ci}}{\sigma_{ci}} . \sigma + \mu \tag{2.3}$$

di mana:

y_i	=	nilai asli dari data ke-*i*
$\hat{y}_i$	=	*fair labels*
μ_{ci}	=	mean dari semua *instances* (c_i) pada variabel terlindungi
μ	=	mean target
σ_{ci}	=	standar deviasi dari semua *instances* (c_i) pada variabel terlindungi
σ	=	standar deviasi target

Pelatihan model dilakukan untuk meminimalkan fungsi *loss* (*L*) berdasarkan *fair labels*. Prediksi model dinyatakan dengan *p = M (X; W)*, yaitu fungsi model (M) yang digunakan untuk menghasilkan prediksi berdasarkan data input *X* dan parameter model *W*. Fungsi loss *L(y,p)* merupakan *Mean Squared Error* (MSE), yang umum digunakan untuk mengukur selisih kuadrat antara label sebenarnya ($\hat{y}_i$) dan prediksi model (p_i) untuk seluruh data *n*. Perhitungan fungsi kerugian didefinisikan pada Persamaan 2.4

$$L(y,p) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - p_i)^2 \tag{2.4}$$

di mana:

$L(y,p)$ = fungsi kerugian, dengan y nilai sebenarnya dan p prediksi model

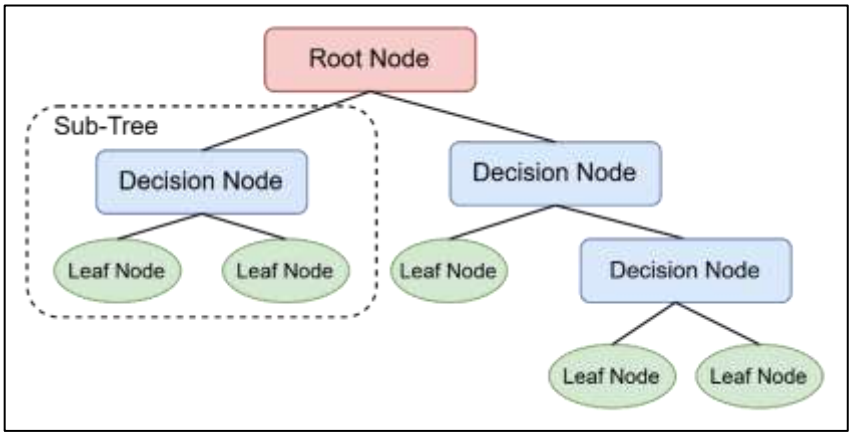
p_i = prediksi model untuk data ke-i

2.1.6 *Data Preprocessing*

Data preprocessing adalah serangkaian teknik yang bertujuan untuk meningkatkan kualitas data. Teknik yang digunakan dalam *data preprocessing* antara lain *data cleaning* dan *data transformation*. *Data cleaning* merupakan proses menangani *missing values* melalui penghapusan atau imputasi, serta deteksi *outliers*. Data *transformation* merupakan proses mengonversi data numerik menjadi kategori (atau sebaliknya) agar sesuai dengan algoritma yang digunakan, *serta* mencakup penggabungan fitur (C. Fan dkk., 2021). Selain itu, terdapat juga *Exploratory Data Analysis* (EDA) yang merupakan proses observasi awal pada data untuk menemukan pola, anomali, dan menguji hipotesis awal serta asumsi dengan beberapa statistik dan representasi visual (Da Poian dkk., 2023).

2.1.7 *Decision Tree*

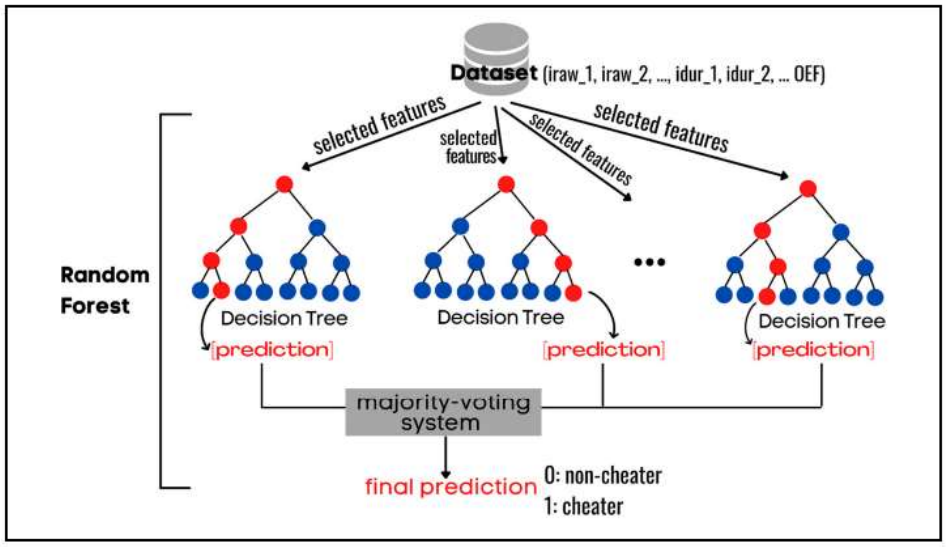
Decision Tree (DT) adalah metode untuk membuat model prediksi berdasarkan struktur pohon di mana setiap node mewakili keputusan atau pembagian berdasarkan atribut tertentu. Cara kerja DT dimulai dengan memilih fitur paling informatif sebagai akar pohon, kemudian membagi data menjadi subset berdasarkan nilai-nilai fitur tersebut, dan diulang secara rekursif hingga mencapai kondisi tertentu (Mienye & Jere, 2024). Contoh struktur DT ditunjukkan ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Struktur Decision Tree

2.1.8 *Bagging*

Bootstrap Aggregating (*bagging*) adalah metode yang menggunakan satu model pengklasifikasi untuk dilatih pada berbagai subset dari dataset yang sama. Subset data dibuat dengan cara *bootstrapping*, yaitu mengambil salinan data secara acak dengan penggantian yang akan digunakan untuk melatih beberapa model secara bersamaan. Hasil akhir didapatkan dengan menggabungkan prediksi dari berbagai model dasar yang umumnya menggunakan *majority voting* (Gheni & Al-Yaseen, 2023). Mekanisme *bagging* secara umum ditunjukkan pada Gambar 2.2



Gambar 2. 3 Mekanisme *Bagging (T. Zhou & Jiao, 2023)*

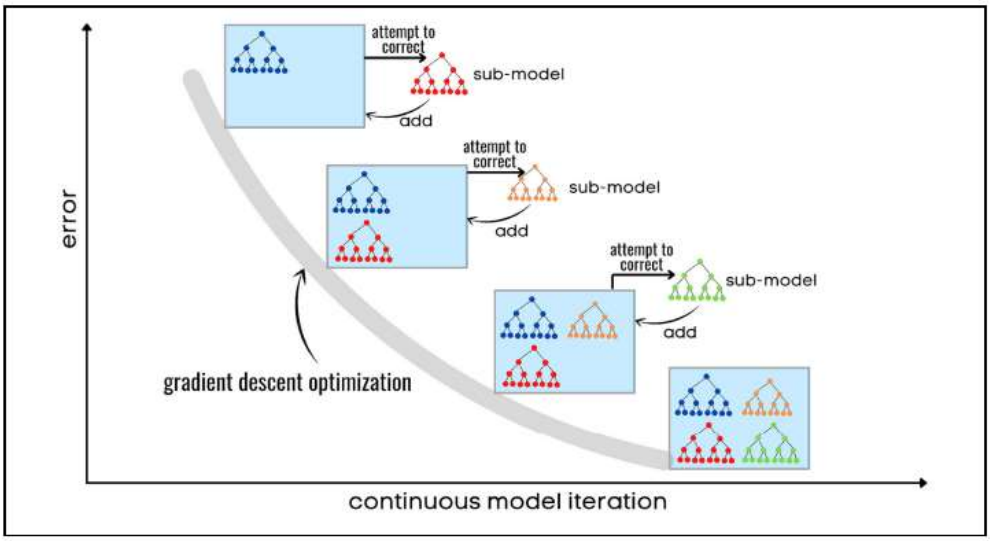
Random Forest (RF) merupakan teknik *bagging* yang menggunakan *decision trees* untuk membentuk sekumpulan pohon keputusan (*forest*). Setiap *tree* dalam RF dibangun melalui proses pemisahan (*node splitting*) hingga aturan penghentian terpenuhi. Prinsip utama pemisahan yaitu meminimalkan impuritas yang diukur dengan indeks gini dan varians. Prediksi final didapatkan dari *majority voting* atau *averaging* berdasarkan hasil dari seluruh *decision tree* pada *forest* (Hu & Szymczak, 2023). *Pseudocode* untuk algoritma RF terdapat pada Algoritma 2.1.

Algoritma 2.1 Random Forest

```
for b = 1 to B:
   1. Ambil bootstrap sample Z* dari data latih.
   2. Bangun pohon T_b dengan:
         a. Pilih m variabel acak dari p variabel.
         b. Pilih split terbaik menggunakan kriteria tertentu.
         c. Pisahkan node menjadi dua node anak.
         d. Ulangi proses hingga kondisi berhenti tercapai.
Output: kumpulan pohon {T_b}_1^B.
Prediksi:
Regresi: untuk prediksi di titik baru x, hitung rata-rata hasil
prediksi semua pohon.
Klasifikasi: ambil prediksi kelas dari setiap pohon dan gunakan
majority voting.
```

2.1.9 *Boosting*

Boosting adalah metode yang menggunakan proses *forward stagewise* untuk mengubah model lemah menjadi model kuat. Proses ini dilakukan dengan memberikan bobot lebih besar pada sampel pelatihan yang salah dihitung pada iterasi sebelumnya. Hasil akhir berupa penggabungan semua iterasi menggunakan pembobotan (Y. Zhang dkk., 2022). Gambar 2.3 menunjukkan mekanisme *boosting* secara umum.



Gambar 2. 4 Mekanisme *Boosting (T. Zhou & Jiao, 2023)*

a. *Adaptive Boosting* (AdaBoost) merupakan jenis algoritma *boosting* yang melibatkan penyesuaian bobot pada pelatihan model baru. Bobot dari setiap regresor mencerminkan kepentingannya ketika mendapatkan hasil akhir prediksi. Sampel yang salah diprediksi akan diberikan bobot yang lebih besar. Bobot baru tersebut digunakan untuk pelatihan selanjutnya. Setelah seluruh iterasi selesai, semua model akan digabungkan untuk membentuk *strong learner*. *Pseudocode* untuk algoritma AdaBoost terdapat pada Algoritma 2.

Algoritma 2.2 AdaBoost

```
Input: Data latih S = (x1, y1), ..., (xm, ym)
    Algoritma dasar L
    Jumlah iterasi T
    Prosedur:
    Inisialisasi bobot untuk setiap sampel x
    for t = 1, ..., T:
       1. Latih regresor dasar untuk meminimalkan weighted
          error
       2. Hitung bobot regresor menggunakan kesalahan weighted
          error
       3. Perbarui bobot untuk setiap sampel x
    end for
Output: Kombinasikan prediksi dari semua regresor dasar dengan
weighted sum
```

b. *Gradient Boosting Machine* (GBM) merupakan jenis algoritma *ensemble* yang membangun model prediktif secara bertahap dengan menambahkan *base learner* yang berkorelasi dengan gradien negatif dari fungsi *loss* seluruh *ensemble*. Setiap *base learner* dibangun untuk memperbaiki kesalahan (residual) yang dihasilkan oleh model sebelumnya. Teknik *linear search* memastikan bahwa kontribusi *base learner* optimal untuk mengurangi kesalahan prediksi tanpa *overfitting*. *Pseudocode* untuk algoritma GBM terdapat pada Algoritma 3.

Algoritma 2.3 GBM

```
Input: Data latih S = (x1, y1), ..., (xm, ym)
    Fungsi loss L(y,F(x))
    Jumlah iterasi T
    Prosedur:
    Inisialisasi model awal dengan loss kuadrat
    for t = 1, ..., T:
       1. Hitung residual untuk setiap sampel
       2. Latih base learner menggunakan nilai residual
       3. Cari skalar linear search yang meminimalkan fungsi
          loss
       4. Perbarui model
    end for
Output: kombinasi prediksi semua model sebelumnya
```

c. *Extreme Gradient Boosting* (XGBoost) merupakan pengembangan dari algoritma *gradient boosting* yang ditingkatkan dengan penambahan teknik regularisasi L1 (Lasso) dan L2 (Ridge) pada fungsi kerugian (*loss function*) untuk membantu mengurangi risiko *overfitting*. XGBoost menggunakan pendekatan *Taylor approximation*, yaitu memanfaatkan turunan pertama (*gradien*) dan turunan kedua (*hessian*), untuk mempercepat dan menstabilkan proses optimasi. *Pseudocode* untuk algoritma XGBoost terdapat pada Algoritma 4.

Algoritma 2.4 XGBoost

```
Input: Data latih S = (x1, y1), ..., (xm, ym)
    Fungsi loss L(y,F(x))
    Jumlah iterasi T
    Learning rate η
    Prosedur:
    Inisialisasi model awal dengan loss kuadrat
    for t = 1, ..., T:
       1. Hitung pseudo-residual untuk setiap sampel
       2. Latih base learner menggunakan nilai residual
       3. Hitung nilai optimal leaf
       4. Perbarui model
    end for
Output: kombinasi prediksi semua model
```

d. *Light Gradient Boosting Machine* (LightGBM) merupakan implementasi yang lebih efisien dari GBM. LightGBM menggunakan strategi *Gradient-based One-Side Sampling* (GOSS) dan *Eksklusif Feature Bundling* (EFB) untuk meningkatkan kecepatan dan akurasi. Teknis GOSS melibatkan pengecualian sejumlah besar sampel dengan gradien kecil dan hanya menggunakan sampel yang tersisa untuk menghitung *information gain* (IG). IG dihitung berdasarkan pengurangan kesalahan setelah data dibagi menjadi beberapa subset, yang digunakan untuk menentukan pemilihan terbaik pada pohon keputusan. Teknik FEB melakukan pemilihan fitur dengan menggabungkan atribut yang saling eksklusif, sehingga mengurangi jumlah atribut. Kedua teknik tersebut membuat kompleksitas data berkurang tanpa kehilangan informasi penting. *Pseudocode* untuk algoritma LightGBM terdapat pada Algoritma 5.

Algoritma 2.5 LightGBM

```
Input: Data latih S = (x1, y1), ..., (xm, ym)
    Fungsi loss L(y,F(x))
    Jumlah iterasi T
    Rasio sampling gradien besar a, dan gradien kecil b
    Prosedur:
    Gabungkan fitur yang saling ekslusif menggunakan FEB
    Inisialisasi model awal
```

```
    for t = 1, ..., T:
       1. Hitung gradien absolut setiap data
       2. Resampling dataset menggunakan GOSS
       3. Hitung nilai informasi gain untuk setiap fitur
       4. Dapatkan tree baru
       5. Perbarui model
    end for
Output: kombinasi dari semua tree yang dibangun
```

e. *Categorical Boosting* (CatBoost) merupakan algoritma *boosting* yang dirancang untuk menangani data dengan banyak fitur kategorikal. Algoritma ini menghindari penggunaan sampel yang sama pada setiap iterasi dan mengurangi *overfitting*. CatBoost menggunakan *unbiased gradient estimation*, yaitu pendekatan estimasi gradien yang tidak bias untuk meningkatkan stabilitas dan akurasi model. CatBoost secara otomatis mengubah fitur kategorikal menjadi numerik berbasis permutasi acak, sehingga tidak memerlukan *pre-processing*. *Pseudocode* untuk algoritma CatBoost terdapat pada Algoritma 6.

Algoritma 2.6 CatBoost

```
Input: Data latih S = (x1, y1), ..., (xm, ym)
    Fungsi loss L(y,F(x))
    Jumlah iterasi T
    Learning rate η
    Fitur kategorikal C
    Prosedur:
    Encode C menggunakan target statistics dengan ordered
sampling
    Inisialisasi model awal
    for t = 1, ..., T:
       1. Hitung gradien absolut setiap data
       2. Latih base learner dengan ordered boosting
       3. Hitung nilai optimal untuk setiap lead
       4. Perbarui modeldengan learning rate η
    end for
Output: kombinasi dari semua tree yang dibangun
```

2.1.10 *Grid Search*

Grid Search adalah pendekatan *hyperparameter tuning* yang digunakan untuk menemukan kombinasi *hyperparameter* optimal pada model. *Hyperparameter* merupakan parameter yang harus ditentukan sebelumnya untuk melatih model dan tidak dapat dipelajari dari data. *Grid search* bekerja dengan mencari secara ekstensif melalui semua kombinasi *hyperparameter* yang potensial dalam rentang atau kumpulan nilai tertentu (Shams dkk., 2024). Terdapat beberapa *hyperparameter* yang dapat dioptimalisasi sebagai berikut (Tarwidi dkk., 2023).

a. ‘max_depth’, yaitu kedalaman maksimum setiap pohon.
b. ‘n_estimators’, yaitu jumlah *decision tree* pada *forest*.
c. ‘learning_rate’, yaitu ukuran langkah pada setiap iterasi untuk memperbarui bobot model selama pelatihan.
d. ‘subsample’, yaitu proporsi data yang akan dijadikan sampel untuk setiap pohon.
e. ‘colsample_bytree’, yaitu rasio subsampel kolom saat membangun setiap pohon.

2.1.11 *K-Fold Cross Validation*

K-Fold Cross Validation adalah salah satu metode *cross-validation* yang membagi dataset menjadi *k* subset dengan ukuran hampir sama (*folds*). Proses validasi dilakukan dengan mengambil satu *fold* sebagai data uji dan sisa *fold* lainnya sebagai data latih. Proses ini diulang *k* kali, sehingga setiap *fold* digunakan sebanyak satu kali sebagai data uji. Hasil evaluasi dari setiap iterasi dirata-ratakan untuk mendapatkan estimasi performa model (Yates dkk., 2023).

2.1.12 Metrik Evaluasi

2.1.12.1 C-Index

Concordance Index (C-Index) adalah metrik yang digunakan untuk mengevaluasi kemampuan model prediksi dalam membedakan subjek dengan risiko lebih tinggi dari subjek dengan risiko lebih rendah. C-Index menilai kesesuaian urutan kejadian pada subjek. Subjek yang mengalami kejadian di awal diberi skor prediksi yang lebih tinggi daripada subjek yang mengalami kejadian tersebut setelahnya, atau tidak mengalami kejadian sama sekali dalam periode penelitian. Formula C-Index terdapat pada Persamaan 2.5 (Hartman dkk., 2023).

$$C - Index = \frac{Jml\ Pasangan\ Corcondance + 0.5(Jml\ Pasangan\ Tak\ Tentu)}{Jml\ Pasangan\ yang\ Dapat\ Dibandingkan} \tag{2.5}$$

2.1.12.2 *Time-dependent* ROC (AUC(t))

Time-dependent Receiver Operating Characteristic (*Time-dependent* ROC) merupakan analisis ROC untuk setiap titik waktu pengamatan pada *time-to-event data* (Park dkk., 2021). Perhitungan *time-dependent* ROC mencakup sensitivitas dan spesifisitas. Sensitivitas (*SensCD*) adalah probabilitas individu dengan nilai melebihi ambang batas tertentu telah mengalami kejadian sebelum waktu *t*. Spesifisitas (*SpecCD*) adalah probabilitas individu dengan nilai di bawah ambang batas belum mengalami kejadian sebelum waktu *t*. AUC sebagai luas di bawah kurva ROC dibentuk dengan menggambarkan sensitivitas vs. 1-spesifisitas. Estimasi sensitivitas dan spesifisitas dilakukan pada berbagai nilai ambang batas *k*, sehingga nilai AUC dihitung sebagai integral karena luas di bawah kurva ROC. Formula AUC terdapat pada Persamaan 2.6 (Nuño & Gillen, 2021).

$$AUC(t) = \int_{-\infty}^{\infty} SensCD(t,k)d(1 - SpecCD(t,k)) \quad (2.6)$$

AUC juga dikenal sebagai AUROC, *c-statistic* untuk hasil biner, dan *C-Index* untuk hasil *time-to-event*. Nilai yang mendekati 1 menunjukkan model sangat baik membedakan risiko subjek, sedangkan nilai 0,5 mengartikan model serupa prediksi acak. Kualitas kinerja model melalui kualifikasi rentang nilai terdapat pada Tabel 2.1 (White dkk., 2023).

Tabel 2. 1 Rentang Kualitas Kinerja Model untuk C-Index dan AUC

Skor	**Keterangan**
> 0,9	Luar Biasa (*Excellent*)
0,8 – 0,9	Baik (*Good*)
0,7 – 0,8	Cukup (*Fair*)
0,6 – 0,7	Dapat Diterima (*Acceptable*)
0,5 – 0,6	Gagal (*Failed*)

2.1.12.3 *Statistical Parity (SP)*

Statistical Parity (SP) merupakan konsep *fairness* yang memastikan bahwa probabilitas prediksi positif suatu model sama untuk semua kelompok (H. Zhang dkk., 2021). Pengukuran SP dapat menggunakan *Pearson Correlation Coefficient* (PCC) dan *Mutual Information* (MI). PCC adalah tingkat korelasi atribut sensitif dan prediksi, sementara MI adalah ukuran tingkat ketergantungan informasi antara atribut sensitif dan prediksi (Hashim & Yassin, 2023). Nilai Mutual Information (MI) tidak memiliki batas atas tetap dan selalu bernilai ≥ 0, dimana MI yang mendekati 0 menunjukkan ketergantungan informasi yang rendah antara atribut sensitif dan prediksi.

Rentang nilai PCC berkisar antara −1 hingga +1, di mana nilai 0 menunjukkan tidak adanya hubungan linear antara atribut sensitif dan prediksi. Nilai mendekati +1 atau −1 menandakan hubungan linear yang semakin kuat, baik positif maupun negatif. Nilai absolut PCC dan interpretasinya ditunjukkan pada Tabel 2.3 (Schober & Schwarte, 2018).

Tabel 2. 2 Interpretasi Nilai Absolut Koefisien Korelasi

Nilai Absolut Koefisien Korelasi	**Interpretasi Korelasi**
0,00 – 0,10	Korelasi dapat diabaikan (*Negligible correlation*)
0,10 – 0,39	Korelasi lemah (*Weak correlation*)
0,40 – 0,69	Korelasi sedang (*Moderate correlation*)
0,70 – 0,89	Korelasi kuat (*Strong correlation*)
0,90 – 1,00	Korelasi sangat kuat (*Very strong correlation*)

2.2 Penelitian Terkait

2.2.1 *State of The Art*

Tabel 2. 3 *State of The Art*

No	Peneliti	Judul	Objek Penelitian	Algoritma	Hasil Penelitian
1	(Iwasaki dkk., 2022)	*Establishment of a predictive model for GVHD-free, relapse-free survival after allogeneic HSCT using ensemble learning*	Penanganan *right-censored data* dan *competing risks* untuk prediksi GRFS pasca allo-HCT	*Stacked Ensemble*	Hasil penelitian menunjukkan bahwa *stack ensemble* dari model Cox-PH dan 7 algoritma ML (RSF, Dynamic DeepHit, ADABoost, XGBoost, Extra Tree, *Bagging*, dan GB) dengan validasi menggunakan 5-fold CV menunjukkan C-Index untuk GRFS sebesar 0,670.
2	(E. J. Choi dkk., 2022)	*Predicting Long-term Survival After Allogeneic Hematopoietic Cell Transplantation in Patients with Hematologic Malignancies: ML–Based Model Development and Validation*	Prediksi kelangsungan hidup pasien pasca allo-HCT dalam 5 tahun	GBM, RF, DNN, LR, AdaBoost	Penelitian ini menggunakan *Recursive Feature Elimination* (RFE) untuk seleksi fitur, validasi 10-fold CV menemukan bahwa GBM memiliki performa terbaik dengan AUC-ROC 0,75 dan menghasilkan AUC-ROC final sebesar 0,788.

No	Peneliti	Judul	Objek Penelitian	Algoritma	Hasil Penelitian
3	(Hernandez Boluda dkk., 2024)	*Prediction of Poor Survival after Hematopoietic Cell Transplantation in Myelofibrosis Using Machine Learning Techniques*	Prediksi OS dan NRM pasca HSCT untuk pasien *myelofibrosis*	RSF	Penelitian ini menggunakan *dimensionality reduction*, RSF menunjukkan performa yang lebih baik dibanding skor CIBMTR dengan Harrel's C-index 0,626 vs. 0,581.
4	(Nadiminti dkk., 2021)	*A novel Iowa-Mayo validated composite risk assessment tool for allogeneic stem cell transplantation survival outcome prediction*	Prediksi *Disease-Free Survival* (DFS) dan OS dalam 2 tahun pasca allo-HCT	Regresi Cox Penalized LASSO	Penelitian ini mengusulkan metode Cox-PH dengan LASSO sebagai teknik regularisasi, validasi dengan 10-fold CV menunjukkan C-Index 0,61 untuk DFS dan 0,62 untuk OS.
5	(Y. Zhou dkk., 2024)	*Longitudinal clinical data improve survival prediction after hematopoietic cell transplantation using machine learning*	Melibatkan variabel *baseline* dan data longitudinal dalam prediksi kelangsungan hidup pasca HCT	Naïve Bayes	Penggunaan Naïve Bayes pada prediksi 100 hari dengan kombinasi variabel *baseline* + data longitudinal menghasilkan AUC tertinggi sebesar 0,883

No	Peneliti	Judul	Objek Penelitian	Algoritma	Hasil Penelitian
6	(Shourabizadeh dkk., 2023)	*Machine Learning for the Prediction of Survival Post-Allogeneic Hematopoietic Cell Transplantation: A Single-Center Experience*	Prediksi kelangsungan hidup 100 hari pasca allo-HCT	RF, KNN, SVM, DT, LR, XGB	Penelitian ini menggunakan 5 dataset, performa terbaik didapatkan oleh RF dengan dataset PMCC menggunakan 45 variabel, mencapai AUC 0,71.
7	(Rouzbahani dkk., 2025)	*Predictive modeling of outcomes in acute leukemia patients undergoing allogeneic hematopoietic stem cell transplantation using machine learning techniques*	Eksplorasi kombinasi algoritma ML dengan seleksi fitur untuk prediksi OS, *relapse*, dan GVHD pada pasien leukimia akut pasca allo-HCT	ST, GB, GLMB, GLMN, CB, RSF, CoxPH	Penelitian ini mengembangkan 28 model ML, yaitu kombinasi 7 algoritma ML dengan 4 seleksi fitur (UCI, Boruta, MI, IBMA), menemukan C-Index terbaik pada OS, *relapse* dan GVHD berada pada rentang 0,61–0,68.

No	Peneliti	Judul	Objek Penelitian	Algoritma	Hasil Penelitian
8	(Meyer dkk., 2024)	*Machine Learning Based Prediction of One Year Mortality after Allogeneic Hematopoietic Cell Transplantation (alloHCT) Highlights Importance of Pre-Transplant Immunocompetence*	Prediksi risiko mortalitas 1 tahun pasca allo-HCT untuk mengganti skor HCT-CI	DT, AdaBoost, GB dan RF	Penelitian ini menggunakan 28 fitur, validasi model menggunakan 10× pengulangan 5-Fold *nested* CV, performa terbaik dicapai menggunakan RF dengan AUC 0,77 pada data latih dan 0,79 pada data uji.
9	(E.-J. Choi dkk., 2020)	*Machine Learning-Based Approach to Predict Survival after Allogeneic Hematopoietic Cell Transplantation in Hematologic Malignancies*	Prediksi kelangsungan hidup pasien pasca allo-HCT	RF, SVM, LT, *feed forward* NN	Penelitian ini menggunakan RFE sebagai seleksi fitur, validasi model menggunakan 10-fold CV menunjukkan bahwa RF mencapai kinerja paling baik dengan AUC-ROC 0,812 dan akurasi 0,73.
10	(Wang dkk., 2025)	*Machine learning algorithms to predict heart failure with preserved ejection fraction among patients with premature myocardial infarction*	Prediksi risiko Heart Failure with Preserved Ejection Fraction (HFpEF) selama rawat inap	Lasso-Logistic, XGBoost, RF, KNN, SVM	Penelitian ini menggunakan Lasso sebagai seleksi fitur dan mengkategorikan variabel kontinu berdasarkan referensi klinis, hasil dari 5-fold CV menemukan bahwa XGBoost memberikan performa terbaik dengan AUC 0.854.

No	Peneliti	Judul	Objek Penelitian	Algoritma	Hasil Penelitian
11	(Gong dkk., 2021)	*Application of machine learning approaches to predict the 5-year survival status of patients with esophageal cancer*	Prediksi status kelangsungan hidup pasien kanker esofagus	GBM, XGBoost, CatBoost, LightGBM, GBDT, RF, ANN, NB, SVM	Penelitian ini menggunakan uji chi-square untuk seleksi fitur dan *hyperparameter tuning* dengan Bayesian Optimization, hasil penelitian menunjukkan fitur lengkap lebih baik, dengan XGBoost pada 5-fold CV mencapai AUC 0,852.
12	(Azar dkk., 2022)	*Application of machine learning techniques for predicting survival in ovarian cancer*	Memprediksi kelangsungan hidup pasien kanker ovarium	KNN, SVM, DT, RF, AdaBoost, XGBoost	Penelitian ini menggunakan *Grid Search* untuk pemilihan *hyperparameter*, hasil 5-fold CV menunjukkan bahwa RF mencapai AUC tertinggi sebesar 82,38.
13	(Abdollahzade dkk., 2025)	*Predictive models for post-liver transplant survival using machine learning techniques in three critical time intervals*	Prediksi kelangsungan hidup pasien pasca transplantasi hati	DT, RF, LR, GaussianNB, dan LDA	Penelitian ini menemukan bahwa DT memberikan hasil terbaik dalam seleksi fitur, dimana GaussianNB mencapai performa terbaik pada survival 1 tahun dengan AUC 0,61, sensitivitas 0.98, dan F1-score 0.89

No	Peneliti	Judul	Objek Penelitian	Algoritma	Hasil Penelitian
14	(Ayers dkk., 2021)	*Using machine learning to improve survival prediction afterheart transplantation*	Prediksi kelangsungan hidup 1 tahun pasca *orthotopic heart transplantation* (OHT)	RF, DNN, LR dan AdaBoost	Hasil penelitian menunjukkan bahwa model LR konvensional memiliki AUC-ROC 0,649, sementara *ensamble model* mencapai AUROC 0,764
15	(Alkhadar dkk., 2021)	*Comparison of machine learning algorithms for the prediction of five-year survival in oral squamous cell carcinoma*	Komparasi algoritma ML dalam prediksi kelangsungan hidup 5 tahun pasien OSCC	LR, KNN, Naïve Bayes, DT, RF	Penelitian ini menggunakan 9 variabel dengan k-fold CV, mendapatkan performa terbaik pada DT dengan akurasi 76% dan AUC-ROC 0,77, model kedua terbaik pada LR dengan akurasi 60% dan ROC-AUC 0,69.

Tabel 2.3 merangkum berbagai penelitian terkait dengan penelitian yang dilakukan. *Ensemble learning* telah banyak digunakan oleh penelitian sebelumnya untuk prediksi kelangsungan hidup, meliputi model ML tunggal, metode *bagging* (seperti *Random Forest*), metode *boosting* (seperti GBM, LightGBM, XGBoost, dan AdaBoost) serta metode *stacking* dengan kombinasi berbagai model. Naïve Bayes mencapai AUC tertinggi pada penelitian terkait dengan AUC 0,883 (Y. Zhou dkk., 2024), diikuti oleh XGBoost - AUC 0,854 pada prediksi HFpEF (Wang dkk., 2025), dan AUC 0,852 pada prediksi untuk kanker esofagus (Gong dkk., 2021). Evaluasi dengan C-Index menunjukkan penggunaan *ensemble* untuk model Cox-PH dan 7 algoritma ML menghasilkan nilai tertinggi 0,670 (Iwasaki dkk., 2022).

Mengacu pada temuan dalam penelitian sebelumnya, penelitian ini akan membuat model berbasis *bagging* dan *boosting* untuk prediksi kelangsungan hidup pasien pasca allo-HCT. Hyperparameter tuning digunakan untuk mendapatkan kombinasi *hyperparameter* yang optimal. Selain mengupayakan performa kompetitif, penelitian ini juga memberikan perhatian khusus terhadap aspek *fairness* melalui penerapan teknik *fairness regression*.

2.2.2 *Matriks Penelitian*

Tabel 2. 4 Matriks Penelitian

No	Penulis	Ruang Lingkup										
		Seleksi Fitur	*Hyperparameter Tuning*	FaiReg	Algoritma				Cross Validation	Evaluasi		
					Bagging	Boosting	Stacking	Lainnya		C-Index	AUC-ROC	Akurasi
1	(Iwasaki dkk., 2022)	-	-	-	✓	✓	✓	✓	✓	✓	-	-
2	(E. J. Choi dkk., 2022)	✓	-	-	✓	✓	-	✓	✓	-	✓	-
3	(Hernandez Boluda dkk., 2024)	✓	-	-	✓	-	-	-	-	✓	-	-
4	(Nadiminti dkk., 2021)	✓	-	-	-	-	-	✓	✓	✓	-	-
5	(Y. Zhou dkk., 2024)	✓	-	-	-	-	-	✓	-	-	✓	-
6	(Shourabizadeh dkk., 2023)	✓	-	-	✓	✓	-	✓	✓	-	✓	-
7	(Rouzbahani dkk., 2025)	✓	-	-	✓	✓	-	✓	✓	✓	-	-
8	(Meyer dkk., 2024)	-	-	-	✓	✓	-	-	✓	-	✓	-
9	(E.-J. Choi dkk., 2020)	✓	-	-	✓	-	-	✓	✓	-	✓	✓
10	(Wang dkk., 2025)	✓	-	-	✓	✓	-	✓	✓	-	✓	-
11	(Gong dkk., 2021)	✓	✓	-	✓	✓	-	✓	✓	-	✓	-
12	(Sorayaie Azar dkk., 2022)	-	✓	-	✓	✓	-	✓	✓	-	✓	-
13	(Abdollahzade dkk., 2025)	✓	-	-	✓	-	-	✓	✓	-	✓	-
14	(Ayers dkk., 2021)	✓	-	-	✓	✓	✓	✓	-	-	✓	-
15	(Alkhadar dkk., 2021)	✓	-	-	✓	-	-	✓	✓	-	✓	✓
16	*Our Research*	-	✓	✓	✓	✓	-	-	✓	✓	✓	-

Tabel 2.4 merupakan matriks penelitian yang terkait dengan penelitian yang akan dilakukan. Penelitian ini berfokus pada penggunaan *ensemble learning* untuk prediksi survival dan teknik *fairness regression* untuk kinerja model yang lebih adil. Matriks penelitian digunakan untuk memberikan rincian mengenai perbedaan antara penelitian yang dilakukan, dengan penelitian sebelumnya.

Berdasarkan matriks penelitian pada Tabel 2.2, penelitian ini akan melakukan prediksi kelangsungan hidup pasien pasca allo-HCT dengan *ensemble learning* serta menerapkan teknik *fairness regression*. Metode *bagging* dan *boosting* diambil sebagai metode pengembangan model dengan merujuk pada penelitian sebelumnya yang hampir selalu menyertakan kedua metode tersebut. Hasil dari penelitian sebelumnya menunjukkan bahwa meskipun akurasi model cukup tinggi, masih terdapat potensi ketidakadilan dalam prediksi terhadap kelompok ras. Hal ini menunjukkan bahwa tanpa penerapan *teknik fairness regression*, model dapat berperforma baik secara teknis tetapi tetap menyisakan isu *fairness* yang belum terselesaikan. Perbedaan antara penelitian yang dilakukan dengan penelitian sebelumnya terletak pada penerapan teknik *fairness regression* dalam mencapai keadilan model. *Hyperparameter tuning* dilakukan untuk mendapatkan kombinasi *hyperparameter* yang optimal. Penggunaan *cross-validation* juga diterapkan untuk memastikan performa model pada data yang belum pernah dilihat sebelumnya.