

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Waktu tunggu di Instalasi Gawat Darurat (IGD) yang berkepanjangan mengurangi kualitas perawatan dan meningkatkan efek samping dan terkadang tidak dapat diubah (Mahmoodian dkk., 2014). Studi cross-sectional analitik ini dilakukan dalam dua fase dari Desember 2012 hingga Mei 2013. Peneliti mendatangi IGD dari dua rumah sakit dan mendapat informasi dari 900 pasien. Data yang diambil ialah waktu kedatangan, tingkat triase, dan waktu kunjungan pertama oleh dokter. Waktu tunggu rata-rata dari kedatangan hingga kunjungan pertama oleh dokter untuk 900 pasien yang termasuk dalam penelitian ini adalah 8 (5-14) menit [median (kisaran interkuartil)]. Untuk pasien yang dirawat di unit rujukan, waktu tunggu adalah 84 (43-145) menit untuk pesanan dokter dan 85 (45-147) menit untuk dimulainya intervensi klinis pertama; 75% dari pasien dalam triase level I, 84,6% pada triase level II, dan 95,6% pada triase level III dikunjungi dalam batas waktu target.

Kepuasan pasien akan meningkat jika mampu memprediksi waktu tunggu secara akurat dan penundaan janji temu yang dijadwalkan dan memungkinkan pegawai/perawat untuk melayani antrean pasien lebih akurat. Curtis dkk. (2018) mempelajari penerapan model *machine learning* untuk memprediksi waktu tunggu di fasilitas radiologi *walk-in* dan fasilitas radiologi terjadwal (CT, MRI, dan *ultrasound*). Data yang tersedia dalam sistem informasi radiologi digunakan untuk memprediksi waktu tunggu. Beberapa algoritma *machine learning*, seperti *neural network*, *random forest*, *support vector machine*, *elastic net*, *multivariate adaptive regression splines*, *k-th nearest neighbour*, *gradient boosting machine*, *bagging*, *classification and regression tree*, and *linear regression* dievaluasi untuk menemukan metode yang paling akurat. Dalam penelitian ini, peneliti menggunakan model *elastic net* yang menghasilkan simpulan bahwa model ini memiliki kinerja yang baik untuk memprediksi waktu tunggu.

Kedatangan yang tak tentu dari pasien *walk-in* secara signifikan mempengaruhi operasi harian fasilitas kesehatan. Untuk meningkatkan kinerja departemen rawat jalan, Pan dkk. (2019) mencoba untuk membuat jadwal janji temu dengan mempertimbangkan *walk-in* dan target waktu tunggu untuk pasien janji temu. Model pemrograman stokastik diusulkan untuk menyelesaikan masalah ini dengan tujuan meminimalkan pasien yang menunggu dan biaya perawatan. Model ini diformulasikan untuk menetapkan kebijakan penjadwalan real-time optimal di bawah jadwal janji yang diberikan. Jadwal janji ditentukan oleh pendekatan pemrograman stokastik dua tahap dan peningkatan pencarian lokal.

Menurut Babashov dkk. (2017) berdasarkan penelitian yang telah dilakukan, pasien kanker di Kanada mengalami waktu tunggu yang lama sehingga memungkinkan memiliki dampak klinis negative, seperti meningkatkan resiko kambuh lokal atau kelangsungan hidup yang kurang baik. Beberapa studi sudah menyelidiki waktu tunggu radioterapi untuk memahami sumber daya mana yang paling penting untuk mengurangi waktu tunggu dan untuk menentukan prediksi waktu tunggu yang akan ditempuh oleh pasien. Mereka mengembangkan simulasi kejadian-diskrit dari perjalanan pasien dari tempat rujukan ke ahli onkologi radiasi hingga awal radioterapi, dengan mempertimbangkan langkah-langkah yang berurutan.

Sebagian besar rumah sakit, antrian pasien yang mendorong dan menentukan setiap tempo alur kerja, maka manajemen waktu tunggu pasien adalah suatu hal yang penting (Pianykh dan Rosenthal, 2015). Mereka mempelajari prediktabilitas waktu tunggu pasien dan mengidentifikasi prediktor yang paling berpengaruh dengan mengembangkan 25 parameter yang berhubungan dengan menunggu yang disarankan dalam penelitian sebelumnya dan diamati dalam penelitian ini. Semua parameter dipilih sebagai turunan dari dataset Sistem Informasi Rumah Sakit. Faktor dan model prediksi waktu tunggu yang paling efisien sudah ditemukan, seperti model *line-size*. Model-model lain pun terbukti sama-sama akurat dan efisien secara komputasi.

Waktu tunggu yang tidak perlu dan waktu yang lama mengakibatkan pemborosan sumber daya manusia dan meningkatkan rasa frustrasi yang dialami

pasien. Lebih baik jika pasien dapat mengetahui prediksi waktu tunggu dan menerima urutan pelayanan yang paling efisien melalui web yang akan terus diperbarui secara waktu nyata. Oleh karena itu, diusulkan Algoritma Prediksi Waktu Pelayanan Pasien untuk memprediksi waktu tunggu untuk setiap tugas perawatan untuk pasien dengan menggunakan data pasien yang realistis dari berbagai rumah sakit untuk mendapatkan model waktu perawatan pasien untuk setiap pelayanan. Berdasarkan skala besar ini, dataset realistis, waktu perawatan untuk setiap pasien dalam antrean saat ini dari setiap tugas diprediksi. Berdasarkan prediksi waktu tunggu, sistem menghitung dan memberi urutan pelayanan yang efisien dan nyaman untuk pasien. Karena skala besar, dataset realistis dan persyaratan untuk respon real-time, Algoritma Prediksi Waktu Pelayanan Pasien dan sistem mengamanatkan efisiensi dan respon latensi rendah. Hasil eksperimen dan simulasi ekstensif menunjukkan efektivitas penerapan model yang diusulkan untuk merekomendasikan rencana pelayanan yang efektif bagi pasien untuk meminimalkan waktu tunggu di rumah sakit (Chen dkk., 2016).

2.2 Dasar Teori

Waktu tunggu pasien terhadap pelayanan di rumah sakit merupakan masalah yang masih banyak dijumpai dalam praktik pelayanan kesehatan dan juga merupakan penyebab ketidakpuasan pasien sehingga menentukan kualitas pelayanan kesehatan. Salah satu dari enam prinsip adalah dapat mengurangi waktu tunggu yang terkadang penundaan tersebut dapat berdampak negatif bagi mereka yang menerima maupun memberi perawatan (Bleustein dkk., 2014). Menurut Bleustein dkk. (2014), waktu tunggu adalah salah satu komponen yang penting dari perawatan berkualitas dalam ekonomi layanan kesehatan. Usaha untuk meminimalkan waktu tunggu harus ditanggapi dengan serius untuk bersaing, mengelola biaya, dan mempertahankan pelanggan.

Antrean dapat ditemukan di mana saja yang terdapat permintaan untuk barang atau jasa yang secara temporer dan lokal melebihi kemampuan penyedia untuk memenuhi permintaan. Antrean terjadi karena variabilitas dalam pola kedatangan

dan waktu layanan, dan antrean panjang dapat menggagalkan pelanggan dan menyebabkan mereka mencari layanan di tempat lain. Manajer harus secara berkala meninjau proses-proses dalam kegiatannya untuk mengidentifikasi antrean pelanggan dan mengevaluasi pengalaman pelanggan saat menunggu. Sebuah perusahaan harus sadar akan kebutuhan pelanggannya dan menghormati waktu pelanggan (Weiss dan Tucker, 2018).

Manajemen antrean pasien dan prediksi waktu menunggu menjadi pekerjaan rumit bagi pihak rumah sakit. Setiap pelayanan dapat memiliki waktu yang berbeda untuk setiap pasien, yang membuat prediksi waktu dan rekomendasi pelayanan menjadi sangat rumit. Seorang pasien biasanya perlu untuk menjalani beberapa pelayanan sesuai dengan kondisinya (Chen dkk., 2016).

Waktu tunggu yang lama untuk pasien biasanya terlihat di luar fasilitas pasien, dan kesulitan ini berkontribusi pada berbagai masalah kesehatan masyarakat, termasuk gangguan akses ke perawatan, gangguan pola kerja rumah sakit, dan ketidakpuasan pasien (Xie dan Or, 2017).

2.2.1 Algoritma Prediksi Waktu Pelayanan Pasien

Chen dkk. (2016) juga mengusulkan sebuah algoritma yang dapat menghasilkan prediksi waktu tunggu, disebut dengan *Patient Treatment Time Prediction*. Algoritma Prediksi Waktu Pelayanan Pasien dihasilkan berdasarkan data historis rumah sakit. Waktu tunggu setiap pelayanan diprediksi oleh *Prediksi Waktu Pelayanan Pasien* diambil dari jumlah waktu tunggu pasien dalam antrean. Kemudian, Chen dkk. (2016) membangun sebuah sistem *Hospital Queuing-Recommendation* yang menampilkan rekomendasi sistem tersebut mengenai rencana pelayanan yang efisien dan nyaman dengan waktu tunggu yang paling sedikit bagi pasien.

Waktu tunggu pasien dapat dihitung apabila durasi waktu pelayanan setiap pasien sudah diketahui. Durasi waktu pelayanan setiap pasien akan bervariasi sesuai dengan kondisi pasien. Oleh karena itu, Algoritma *Random Forest* digunakan untuk melatih data durasi waktu berdasarkan karakteristik pasien untuk membangun

Patient Treatment Time Prediction. Algoritma *Random Forest* ditingkatkan dalam 4 aspek untuk membangun *Patient Treatment Time Prediction*.

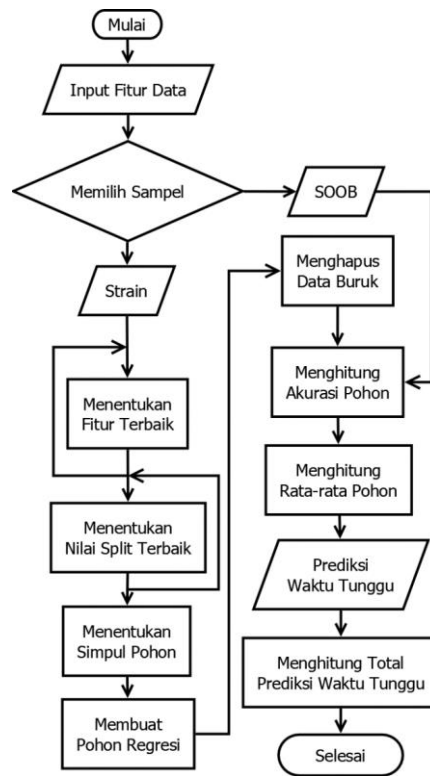
Pertama, semua variabel data yang dipilih sudah dipilah dari fitur yang tidak perlu, seperti nama pasien, alamat, dan nomor telepon. Kedua, Durasi waktu pelayanan adalah variabel yang menjadi sasaran dalam proses ini yang merupakan variabel kontinu, sehingga model CART akan digunakan. Namun, ada pula variabel independent yang merupakan data nominal, seperti rentang waktu (0 - 23) dan hari (Senin - Minggu). Dalam kasus ini, model pohon *two-fork* dari CART tradisional tidak bisa sepenuhnya menghasilkan analisis yang baik. Oleh karena itu, untuk membangun pohon regresi yang tepat diperlukan model multi-cabang dalam proses konstruksi. Selanjutnya, proses pembuatan pohon regresi harus melalui identifikasi data yang cacat kemudian menghapusnya agar mengurangi pengaruh terhadap akurasi. Terakhir, algoritma *Random Forest* menggunakan pemungutan suara tradisional dalam memilih sampel data yang memungkinkan untuk mengambil data yang salah. *Prediksi Waktu Pelayanan Pasien* meningkatkan cara memilih data yang akan digunakan. Oleh karena itu, pengklasifikasi meningkatkan keakuratan klasifikasi dari algoritma *Random Forest* dan mengurangi kesalahan. Sehingga *Prediksi Waktu Pelayanan Pasien* lebih unggul dalam hal akurasi dan kinerja daripada algoritma *Random Forest*. Perincian proses *Prediksi Waktu Pelayanan Pasien* untuk menghitung total waktu tunggu pasien dipresentasikan dalam bentuk *flowchart* pada Gambar 2.1.

Data pra proses

Pada tahap ini, data-data yang dibutuhkan dari rumah sakit dikumpulkan, seperti data pasien dan data pelayanan.

$$S = \{s_1, s_2, \dots, s_N\} \quad (2.1)$$

S adalah himpunan data pasien di rumah sakit yang telah terdaftar, dan s_i data setiap pasien, seperti no. rekam medis, nama, umur, dan jenis kelamin. N notasi untuk jumlah pasien.



Gambar 2.1 *Flowchart* Algoritma Prediksi Waktu Pelayanan Pasien

Rumah sakit menyediakan beberapa pelayanan yang saling berhubungan maupun tidak berhubungan dalam satu hari kunjungan, seperti yang ditunjukkan pada Tabel 2.1 dan setiap pasien bisa mendapatkan beberapa pelayanan, seperti yang ditunjukkan pada Tabel 2.2.

$$X|s_i = \{x_1, x_2, \dots, x_K\} \quad (2.2)$$

Persamaan 2.2 menyatakan himpunan data pelayanan setiap pasien. x_i adalah data setiap pelayanan, seperti pendaftaran, pemeriksaan, pengambilan obat, dan lainnya. Jumlah data X dinotasikan dengan K .

$$Y|x_i = \{y_1, y_2, \dots, y_M\} \quad (2.3)$$

Himpunan data pasien yang dengan rencana perawatan terdiri dari y_j yaitu data setiap pasien yang melakukan beberapa pelayanan dan M untuk jumlah data Y .

Tabel 2.1 Daftar pelayanan yang tersedia

No.	Nama Pelayanan	Data yang dibutuhkan
1.	Pendaftaran	No. antrean pasien, data pasien(no. RM, kategori pasien BPJS atau umum, nama, umur, jenis kelamin) nama poliklinik, dan nama dokter
2.	Pemeriksaan Awal	No. antrean pasien, data pasien, nama poliklinik, nama perawat, data waktu (waktu mulai, waktu selesai)
3.	Pemeriksaan oleh Dokter	No. antrean pasien, data pasien, nama poliklinik, nama dokter, dan data waktu

Data-data pada Tabel 2.2 akan diproses untuk mendapatkan durasi waktu setiap layanan, terutama data yang penting seperti data pasien (no. antrean pasien, umur, jenis kelamin), data layanan (nama poliklinik, nama dokter, nama petugas), dan data waktu (waktu mulai dan waktu selesai). Data-data lainnya tidak dipilih, seperti nama pasien, no. telepon, dan alamat pasien tidak akan digunakan karena tidak dibutuhkan untuk Algoritma Prediksi Waktu Pelayanan Pasien.

Tabel 2.2 Data pasien dengan beberapa layanan

No. Antrean	Umur	Gender	Jenis layanan	Waktu Mulai	Waktu Selesai
A1	24	P	Pendaftaran	20-03-2019 14:00:00	20-03-2019 14:03:00
A1	24	P	<i>Check-up</i>	14:05:00	14:08:00
A1	24	P	Pemeriksaan oleh Dokter	14:10:00	14:35:00

Prediksi Waktu Pelayanan Pasien perlu dilatih dengan berbagai fitur penting dari data yang harus dihitung, seperti durasi waktu pasien dari setiap pelayanan, hari saat pelayanan, dan rentang waktu setiap pelayanan. Fitur-fitur variabel dari data pelayanan yang akan digunakan dalam proses *Prediksi Waktu Pelayanan Pasien* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Fitur-fitur variabel dari data pelayanan untuk PWPP

No.	Nama Variabel Fitur	Keterangan Nilai Data
1	Jenis Kelamin Pasien	“Wanita”, “Pria”
2	Umur Pasien	Umur pasien
3	Nama Poliklinik	Poliklinik Mata
4	Nama Dokter	Nama dokter yang dipilih pasien
5	Nama Layanan	Setiap layanan yang akan direncanakan
6	Waktu Mulai	Waktu mulai pelayanan
7	Waktu Selesai	Waktu berakhirnya pelayanan
8	Hari	Hari pelayanan dilakukan
9	Durasi Waktu Pelayanan	Waktu selesai – waktu mulai

a. Menyusun Data Pengujian untuk *Patient Treatment Time Prediction*

Data-data yang akan digunakan untuk proses *Prediksi Waktu Pelayanan Pasien* harus melalui pengujian. Durasi waktu pelayanan akan berbeda bergantung pada kondisi pasien, maka durasi waktu pelayanan pasien tidak dapat diukur dengan standar absolut. Algoritma *Random Forest* diimprovisasi untuk meningkatkan akurasi *Patient Treatment Time Prediction*.

Beberapa himpunan data pasien dari data asli diambil secara acak untuk dijadikan sampel.

$$S_{\text{train}} = \{S_{\text{train}1}, S_{\text{train}2}, \dots, S_{\text{train}k}\} \quad (2.4)$$

Pada Persamaan 2.4, k adalah jumlah data S_{train} . S_{train} merupakan himpunan data pasien yang dipilih untuk dijadikan sampel.

Data yang tidak dipilih sebagai sampel juga akan disusun menjadi data *Out-Of-Bag* (OOB).

$$S_{\text{OOB}} = \{S_{\text{OOB}1}, S_{\text{OOB}2}, \dots, S_{\text{OOB}k}\} \quad (2.5)$$

Persamaan 2.5 merupakan himpunan data pasien yang tidak dipilih untuk dijadikan sampel dan di dalam S_{OOB} terdapat variabel k yang diartikan sebagai jumlah data S_{OOB} .

Pengujian Pohon Regresi CART (Classification And Regression Tree)

Durasi waktu pelayanan adalah variabel yang menjadi sasaran dalam proses ini yang merupakan variabel kontinu, sehingga jenis pohon keputusan tunggal dalam model *Random Forest* yang digunakan adalah pohon regresi. Dengan demikian, model pohon regresi CART dibuat untuk setiap himpunan data pengujian $S_{\text{train}i}$.

Aspek optimasi pertama dari algoritma *Random Forest* adalah proses pembuatan setiap pohon CART. Semua fitur M dari setiap data pengujian $S_{\text{train}i}$ digunakan dalam proses pengujian. Setiap proses pemisahan simpul pohon, diperlukan setiap variabel fitur y_j dan setiap nilai titik pemisah potensial v_p dari y_j dipilih untuk menghitung *loss function* (y_j, v_p) , yang didefinisikan sebagai berikut:

$$(y_j, v_p) = \arg \min [\sum_{x \in R_L(y_j, v_p)} (y_i - c_L)^2 + \sum_{x \in R_R(y_j, v_p)} (y_i - c_R)^2] \quad (2.6)$$

y_j adalah setiap fitur dari data pengujian, $1 \leq j \leq M$. v_p adalah setiap nilai titik pemisah potensial dari y_j . $R_L(y_j, v_p)$ merupakan himpunan data pertama (kiri), sedangkan $R_R(y_j, v_p)$ himpunan data kedua (kanan). c_L diartikan sebagai nilai rata-rata dari himpunan $R_L(y_j, v_p)$, sedangkan c_R adalah nilai rata-rata dari himpunan $R_R(y_j, v_p)$.

Setelah mendapatkan nilai v_p , maka tahap selanjutnya membuat data pengujian tersebut dipisahkan berdasarkan himpunan kanan atau kiri, seperti yang terlihat pada notasi di bawah ini.

$$R_{L(y_j, v_p)} = \{x | (y_j \leq v_p)\} \quad (2.7)$$

$$R_{R(y_j, v_p)} = \{x | (y_j > v_p)\} \quad (2.8)$$

Beberapa variabel independen dari data pengujian adalah data nominal yang memiliki nilai yang berbeda, seperti hari dalam seminggu (Senin - Minggu). Oleh karena itu, untuk membangun model pohon regresi dengan tepat, yaitu model pohon regresi multi-cabang, yang merupakan aspek optimasi kedua dari algoritma *Random Forest*.

Setelah simpul pohon dibagi menjadi dua garpu dengan variabel y_j dan nilai v_p pada Langkah 2, variabel y_j yang sama terus dipilih untuk menghitung titik split v_{pL} terbaik untuk data di cabang kiri dan v_{pR} untuk data di cabang kanan. Mengambil cabang kiri sebagai contoh, titik pemisah terbaik yang dihitung untuk variabel fitur saat ini didefinisikan sebagai berikut:

$$\Phi(v_{pL} | y_j) = \max_i \Phi(v_i | y) \quad (2.9)$$

PL adalah rasio jumlah data di cabang kiri dari keseluruhan volume data pengujian, sedangkan PR rasio jumlah data di cabang kanan dari keseluruhan volume data pengujian. $p(c_j | y_L)$ merupakan rasio volume data yang dimiliki oleh kelas c_j pada cabang kiri dengan volume data pada cabang kiri.

Jika nilai pemisah $\Phi(v_{pL}|y_j) \geq \Phi(v_p|y_j)$, maka cabang kiri terus dipecah oleh variabel y_j dan nilai v_{pL} . Jika tidak, variabel fitur yang tersisa terus dihitung. Cabang kanan dihitung dengan cara yang sama. Kemudian, setiap simpul dan dua sub-simpulnya dihitung secara berturut-turut. Jika pemisahan variabel yang sama ada di kedua simpul induk dan simpul anak, operasi penggabungan simpul harus dilakukan. Ulangi Persamaan 2.6 sampai 2.10 hingga data pada masing-masing cabang diklasifikasikan dalam satu kelas sebagai simpul daun.

Data buruk mungkin masih ada walaupun sudah dihapus pada tahap pra-proses. Oleh karena itu, aspek optimisasi ketiga dari algoritma *Random Forest* adalah untuk mengurangi pengaruh data buruk terhadap akurasi algoritma. Metode penghilangan data ini dilakukan dalam perhitungan nilai setiap simpul daun CART. Data pada simpul daun saat ini diurutkan dalam urutan menaik. Kemudian, nilai dari tiga titik data Q1, Q2, Q3 dihitung.

Batas dalam dari data buruk ditentukan sebagai berikut:

$$IL = Q1 - 1.5(Q3 - Q1) \quad (2.11)$$

Batas luar dari data buruk ditentukan sebagai berikut:

$$OL = Q3 + 1.5(Q3 - Q1) \quad (2.12)$$

Dalam persamaan 2.11 dan 2.12, Q1 adalah titik data bagian bawah, Q2 adalah titik data median, dan Q3 adalah titik data bagian atas.

Data yang berada di luar kisaran $\{IL; OL\}$ akan dihapus sebagai data buruk. Setelah menghapus data tersebut, nilai rata-rata c_j dari data y_j dihitung di setiap simpul daun dari pohon regresi. Rumus perhitungan didefinisikan sebagai berikut:

$$c_j = \frac{1}{k} \sum y_j, (IL \leq y_j \leq OL) \quad (2.13)$$

Pada persamaan 2.13 terdapat variabel k yang merupakan jumlah item data dalam simpul daun.

Prediksi Waktu Tunggu Pasien untuk Serangkaian Pelayanan

Setiap pasien U_{ik} yang menunggu dalam antrian T_i , durasi waktu perawatan pasien diprediksi dengan *Prediksi Waktu Pelayanan Pasien* yang sudah diuji sesuai dengan karakteristik pasien (jenis kelamin dan umur), faktor hari (senin-minggu), dan faktor-faktor lain (poliklinik dan loket yang tersedia). Durasi waktu perawatan pasien T_{ik} setiap pasien dalam antrian didefinisikan sebagai berikut:

$$T_{ik} = c_j = \frac{1}{k} \sum y_j \quad (2.17)$$

Berikut ini adalah pseudocode dari Algoritma Prediksi Waktu Pelayanan Pasien:

Algoritma : Proses Algoritma Prediksi Waktu Pelayanan Pasien berdasarkan *Random Forest*

Masukan : himpunan data pengujian

Keluaran : pohon keputusan

membuat sampel data pengujian

membuat himpunan data pengujian yang tidak digunakan

membuat pohon kosong

for setiap variabel independent di sampel data pengujian **do**

 menghitung kandidat titik split

for setiap kandidat titik split **do**

 menghitung titik split terbaik

end for

 menambahkan simpul

 memisahkan data untuk cabang kiri

 memisahkan data untuk cabang kanan

for setiap data di cabang kanan dan kiri **do**

 menghitung nilai split terbaik cabang kiri

if nilai split terbaik cabang kiri atau kanan > titik split terbaik **then**

 menambahkan bagian simpul

 memisahkan data menjadi dua garpu/cabang kiri dan kanan

else

```
        mengumpulkan data untuk simpul daun
        menghitung nilai utama dari simpul daun
    end if
end for
end for
return 0
```

2.2.2 Model Framework Waterfall

Metode Model Waterfall membutuhkan kegiatan dasar untuk spesifikasi, pengembangan, validasi dan evolusi sebagai tahapan proses yang terpisah, mempunyai persyaratan spesifikasi sistem, desain sistem, implementasi, pengujian, dan pemeliharaan. (Sommerville, 2011). Metode Waterfall sering disebut dengan *classic life cycle* (Pressman, 2010).

Metode ini mempunyai tahapan-tahapan sebagai berikut:

1. *Requirement Analysis and Definition*

Tahapan ini adalah penetapan fitur, analisa kebutuhan, kendala pembuatan dan tujuan sistem melalui konsultasi dengan pengguna sistem.

2. *System and Software Design*

Tahapan ini adalah tahapan pembentukan arsitektur sistem dan desain berdasarkan persyaratan yang telah ditetapkan pada tahapan sebelumnya.

3. *Implementation and Unit Testing*

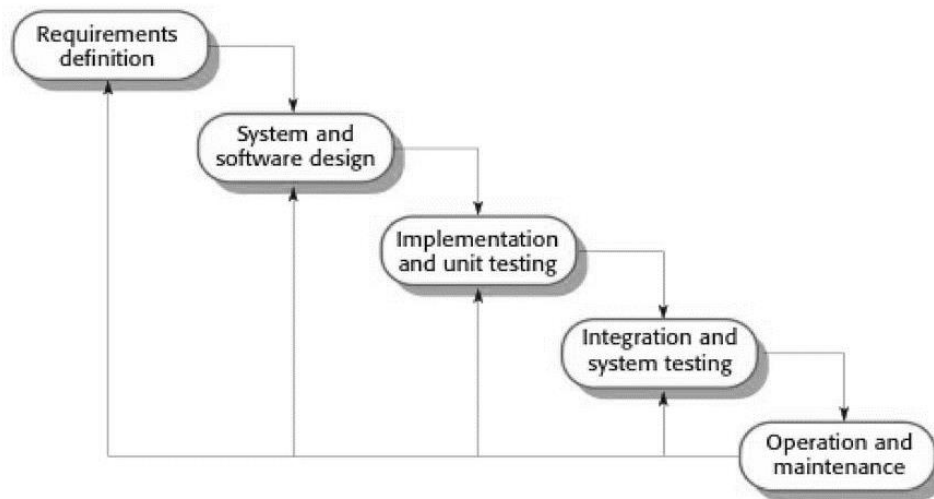
Tahapan ini adalah tahapan hasil dari desain perangkat lunak untuk direalisasikan sebagai satu set program atau unit program.

4. *Integration and System Testing*

Tahapan ini adalah tahapan mengintegrasikan setiap unit program satu sama lain dan dilakukan pengujian untuk memastikan sistem sudah memenuhi persyaratan yang sudah ditetapkan.

5. *Operation and Maintenance*

Tahapan ini adalah tahapan pemasangan sistem. Pada tahap ini juga dilakukan pengujian setelah sistem dijalankan untuk menemukan dan memperbaiki error yang tidak ditemukan pada tahap pengujian sebelumnya.



Gambar 2.2 Model Waterfall (Sommerville, 2011)

Tahapan pada model *waterfall* tidak dapat dimulai sebelum tahapan sebelumnya selesai. Namun, model ini bukan merupakan model linier yang sederhana karena juga melibatkan umpan balik dari satu tahapan ke tahapan lainnya. Maka, dokumen yang dihasilkan pada setiap tahapan ada kemungkinan harus diubah supaya sesuai dengan perubahan yang sudah dibuat (Sommerville, 2011).

2.2.3 Pengujian *Black-box*

Pengujian blackbox merupakan pendekatan komplementer dari Teknik White Box, karena pengujian black box diharapkan mampu mengungkapkan kelas kesalahan yang lebih luas dibandingkan dengan teknik white box. Pengujian black box berfokus pada pengujian persyaratan fungsional perangkat lunak, untuk mendapatkan serangkaian kondisi input yang sesuai dengan persyaratan fungsional suatu program (Laurie, 2006).

Pengujian blackbox berusaha menemukan kesalahan dalam kategori (Perry, 1995):

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi