

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Berbagai penelitian mengenai sistem reservasi barbershop telah dilakukan sebelumnya.

Penelitian oleh (Fauzi et al., 2021) mengembangkan aplikasi online booking berbasis website untuk Dankie Barbershop menggunakan PHP, CodeIgniter, MySQL, dan integrasi Midtrans. Penelitian tersebut berhasil mengurangi waktu tunggu pelanggan dengan menyediakan sistem pemesanan layanan secara online yang efisien dan terintegrasi.

Penelitian oleh (Tya Septiani Nurfausia Koeswara, 2021) mengembangkan sistem informasi berbasis web untuk Barbershop Shavr menggunakan metode waterfall dengan teknologi PHP dan MySQL. Sistem ini fokus pada penanganan booking dan pencatatan transaksi yang sebelumnya masih manual.

Penelitian oleh (Lutvi, 2023) membahas tentang pengembangan Sistem Informasi Reservasi Wisata Camping berbasis web menggunakan metode pengembangan Waterfall. Pembangunan sistem dilakukan dengan menggunakan bahasa pemrograman PHP, basis data MySQL, dan XAMPP sebagai server lokal.

Penelitian oleh (Fambudi et al., 2020) membahas tentang Pengembangan Aplikasi Wedding.co berbasis Web (Studi Kasus: Kota Malang) Metode pengembangan yang digunakan meliputi analisis kebutuhan, perancangan

sistem, implementasi, dan evaluasi aplikasi. Dengan menggunakan teknologi PHP, MySQL, dan Bootstrap, aplikasi Wedding.co berhasil dirancang dan diimplementasikan dengan baik, memberikan solusi yang efektif dan efisien bagi calon pengantin dalam merencanakan pernikahan mereka.

2.2. *First Come First Served* (FCFS)

First Come First Served (FCFS) merupakan salah satu algoritma penjadwalan proses yang paling sederhana dan mendasar dalam ilmu komputer. Algoritma ini bekerja berdasarkan prinsip antrian linear (*queue*) di mana entitas yang tiba lebih awal akan dilayani terlebih dahulu. Setiap permintaan layanan dimasukkan ke dalam antrian sesuai urutan waktu kedatangannya, dan pemrosesan dilakukan secara sekuensial tanpa adanya mekanisme preemption atau prioritas dinamis (Abraham Silberschatz, Peter B. Galvin, n.d.).

Dalam konteks sistem pemesanan barbershop, FCFS diimplementasikan untuk menjamin keadilan (*fairness*) pelayanan, sehingga pelanggan yang melakukan reservasi terlebih dahulu memiliki hak prioritas yang lebih tinggi dibandingkan pelanggan yang melakukan reservasi setelahnya.

2.2.1. Rumus-Rumus Matematis Algoritma FCFS

a. Waktu Penyelesaian (Completion Time)

Waiting Time (WT) adalah selisih antara waktu pelanggan tiba di sistem dengan waktu pelanggan mulai dilayani. Rumus perhitungannya adalah:

$$CT(i) = CT(i - 1) + BT(i) \text{ dengan } CT(0) = AT(1) \text{ (kondisi awal)}$$

Keterangan notasi:

Simbol:	Keterangan:
$CT(i)$	Completion Time proses ke-i (waktu selesai proses ke-i)
$CT(i - 1)$	Completion Time proses ke-(i-1) (waktu selesai proses sebelumnya)
$BT(i)$	Burst Time proses ke-i (durasi waktu layanan yang dibutuhkan)
$CT(0)$	Kondisi awal, diinisialisasi dengan waktu kedatangan proses pertama
$AT(1)$	Arrival Time proses pertama (waktu kedatangan pelanggan pertama)

b. Waktu Tunggu (Waiting Time)

Waiting Time (WT) adalah selisih antara waktu pelanggan tiba di sistem dengan waktu pelanggan mulai dilayani. Rumus perhitungannya adalah:

$$WT(i) = CT(i - 1) - AT(i)$$

atau secara ekuivalen:

$$WT(i) = ST(i) - AT(i)$$

Simbol:	Keterangan:
$WT(i)$	Waiting Time proses ke-i (total waktu tunggu pelanggan ke-i)
$CT(i - 1)$	Completion Time proses ke-(i-1), yaitu waktu mulai proses ke-i dilayani
$AT(i)$	Arrival Time proses ke-i (waktu kedatangan/pemesanan pelanggan ke-i)
$ST(i)$	Start Time proses ke-i (waktu pelanggan ke-i mulai

	mendapatkan layanan)
--	----------------------

c. Turnaround Time

Turnaround Time (TAT) merupakan total waktu yang dihabiskan oleh suatu proses dalam sistem, mulai dari saat proses tiba hingga proses tersebut selesai sepenuhnya. Rumus Turnaround Time adalah:

$$TAT(i) = CT(i) - AT(i)$$

atau secara ekuivalen:

$$TAT(i) = WT(i) + BT(i)$$

Simbol:	Keterangan:
$TAT(i)$	Turnaround Time proses ke-i
$CT(i)$	Completion Time proses ke-i
$AT(i)$	Arrival Time proses ke-i
$WT(i)$	Waiting Time proses ke-i
$BT(i)$	Burst Time proses ke-i (durasi layanan)

d. Rata-Rata Waktu Tunggu (Average Waiting Time)

Average Waiting Time (AWT) digunakan sebagai metrik utama untuk mengukur efisiensi dan performa keseluruhan sistem antrian FCFS. Nilai AWT yang rendah mengindikasikan sistem yang lebih responsif terhadap pelanggan. Rumus AWT adalah:

$$AWT = \frac{1}{n} \sum_{i=1}^n WT(i)$$

Simbol:	Keterangan:
AWT	Average Waiting Time (rata-rata waktu tunggu seluruh

	pelanggan)
n	Jumlah total proses/pelanggan dalam antrian
$WT(i)$	Waiting Time proses ke-i
Σ	Notasi penjumlahan dari $i = 1$ sampai dengan $i = n$

e. Rata-Rata Turnaround Time (Average Turnaround Time)

Average Turnaround Time (ATAT) mengukur rata-rata total waktu yang dibutuhkan oleh setiap pelanggan sejak melakukan pemesanan hingga layanan selesai diterima:

$$ATAT = \frac{1}{n} \sum_{i=1}^n TAT(i)$$

Simbol:	Keterangan:
$ATAT$	Average Turnaround Time (rata-rata turnaround time seluruh pelanggan)
n	Jumlah total proses/pelanggan
$TAT(i)$	Turnaround Time proses ke-i

f. Throughput Sistem

Throughput adalah jumlah proses yang berhasil diselesaikan oleh sistem dalam satu satuan waktu tertentu. Nilai throughput yang tinggi mengindikasikan kapasitas sistem yang optimal:

$$Throughput = \frac{n}{CT(n) - T_{mulai}}$$

Simbol:	Keterangan:
$Throughput$	Jumlah proses yang selesai per satuan waktu (pelanggan/menit)
n	Jumlah total pelanggan yang dilayani
$CT(n)$	Completion Time pelanggan terakhir (ke-n)

T_{mulai}	Waktu operasional sistem dimulai
-------------	----------------------------------

g. Utilisasi Barber (Resource Utilization)

Utilisasi merupakan persentase waktu produktif (waktu melayani pelanggan) terhadap total waktu operasional sistem.

Semakin tinggi utilisasi, semakin efisien penggunaan sumber daya:

$$Utilisasi = \frac{\sum BT(i)}{T_{operasional}} \times 100\%$$

Simbol:	Keterangan:
$Utilisasi$	Persentase tingkat kegunaan barber terhadap waktu operasional
$\sum BT(i)$	Total akumulasi waktu layanan dari seluruh pelanggan
$T_{operasional}$	Total waktu operasional barbershop (jam buka hingga tutup)

2.2.2. Time Slot Scheduling

Time Slot Scheduling adalah metode penjadwalan berbasis segmentasi waktu, di mana total waktu operasional dibagi menjadi sejumlah slot (segmen) waktu dengan durasi yang telah ditentukan. Setiap slot merepresentasikan satu sesi layanan yang dapat dipesan oleh pelanggan (*bookable time unit*). Metode ini memberikan kepastian waktu layanan kepada pelanggan serta memungkinkan sistem untuk mengelola kapasitas secara terstruktur dan terukur (Pinedo, n.d.).

Dalam sistem pemesanan barbershop, Time Slot Scheduling dikombinasikan dengan FCFS untuk menciptakan mekanisme reservasi

yang adil sekaligus terorganisasi, di mana setiap slot waktu hanya dapat dipesan oleh satu pelanggan pada satu waktu tertentu.

a. Jumlah Slot Waktu yang Tersedia

Rumus untuk menghitung jumlah total slot waktu yang dapat dialokasikan dalam satu periode operasional adalah:

$$N_{slot} = \frac{T_{operasional}}{d_{layanan}}$$

Simbol:	Keterangan:
N_{slot}	Jumlah total slot waktu yang tersedia dalam satu hari operasional
$T_{operasional}$	Total durasi operasional barbershop (dalam menit): $T_{tutup} - T_{buka}$
$d_{layanan}$	Durasi standar satu sesi layanan per pelanggan (dalam menit)

Apabila terdapat lebih dari satu barber (b) yang beroperasi secara paralel, maka jumlah slot diperluas menjadi:

$$N_{slot} = \frac{T_{operasional}}{d_{layanan}} \times b$$

Simbol:	Keterangan:
b	Jumlah barber yang beroperasi secara bersamaan (paralel)

b. Penentuan Waktu Mulai Setiap Slot

Waktu mulai (start time) dari slot ke- k dihitung secara berurutan berdasarkan urutan slot yang telah dialokasikan:

$$S(k) = T_{buka} + (k - 1) \times d_{layanan}$$

$$\text{untuk } k = 1, 2, 3, \dots, N_{slot}$$

Simbol:	Keterangan:
$S(k)$	Waktu mulai slot ke-k (dalam satuan waktu, misalnya menit sejak 00:00)
T_{buka}	Waktu pembukaan operasional barbershop
k	Indeks slot ke-k (dimulai dari 1)
$d_{layanan}$	Durasi satu sesi layanan (menit)

c. Waktu Berakhir Setiap Slot

Waktu berakhir (end time) dari slot ke-k dihitung sebagai:

$$E(k) = S(k) + d_{layanan}$$

atau secara langsung:

$$E(k) = T_{buka} + k \times d_{layanan}$$

Simbol:	Keterangan:
$E(k)$	Waktu berakhirnya layanan pada slot ke-k
$S(k)$	Waktu mulai slot ke-k
$d_{layanan}$	Durasi standar satu sesi layanan

d. Fungsi Status Ketersediaan Slot

Setiap slot memiliki status ketersediaan yang direpresentasikan sebagai fungsi biner. Secara formal, fungsi status slot ke-k pada waktu t didefinisikan sebagai:

$\{ 1, \text{ jika slot ke } - k \text{ belum dipesan (tersedia)} \}$

$$A(k) = \{$$

$\{ 0, \text{ jika slot ke } - k \text{ sudah dipesan (tidak tersedia)} \}$

Simbol:	Keterangan:
$A(k)$	Fungsi status ketersediaan slot ke-k (Availability Function)

1	Nilai biner yang menyatakan slot TERSEDIA untuk dipesan
0	Nilai biner yang menyatakan slot SUDAH TERISI / TIDAK TERSEDIA

e. Validasi Pemesanan (Booking Constraint)

Sebuah pemesanan pada slot ke-k oleh pelanggan ke-i dinyatakan valid apabila memenuhi seluruh batasan (constraint) berikut:

$$Valid(i, k) = A(k) = 1 \text{ AND } AT(i) \leq S(k) \text{ AND } S(k) < T_{tutup}$$

Simbol:	Keterangan:
$Valid(I, k)$	Fungsi kevalidan pemesanan pelanggan ke-i pada slot ke-k
$A(k) = 1$	Slot ke-k harus dalam kondisi tersedia
$AT(i) \leq S(k)$	Waktu pemesanan pelanggan tidak boleh melebihi waktu mulai slot
$S(k) < T_{tutup}$	Slot yang dipesan harus berada dalam rentang operasional barbershop

f. Indeks Antrian FCFS pada Time Slot

Dalam integrasi FCFS dengan Time Slot Scheduling, posisi antrian pelanggan ke-i dihitung berdasarkan urutan kedatangan (pemesanan) mereka secara relatif:

$$Q(i) = |\{j : AT(j) < AT(i)\}| + 1 \text{ dengan syarat: } A(k_j) =$$

0 untuk setiap j yang telah memesan slot

Simbol:	Keterangan:
$Q(i)$	Posisi antrian pelanggan ke-i dalam sistem (queue)

	position)
$AT(j)$	Arrival Time (waktu pemesanan) pelanggan ke-j
$AT(i)$	Arrival Time (waktu pemesanan) pelanggan ke-i
$ \{j : \dots\} $	Kardinalitas himpunan, yaitu banyaknya pelanggan j yang memenuhi kondisi
k_j	Slot yang telah dipesan oleh pelanggan ke-j

g. Tingkat Pengisian Slot (Slot Occupancy Rate)

Slot Occupancy Rate (SOR) mengukur proporsi slot waktu yang telah terisi dari total slot yang tersedia, dan berfungsi sebagai indikator kinerja sistem pemesanan:

$$SOR = \frac{N_{slot} \sum_{k=1} (1 - A(k))}{N_{slot}} \times 100\%$$

Simbol:	Keterangan:
SOR	Slot Occupancy Rate (tingkat pengisian slot, dalam persen)
N_{slot}	Jumlah total slot waktu yang tersedia
$A(k)$	Fungsi status ketersediaan slot ke-k (0 = terisi, 1 = tersedia)
$1 - A(k)$	Bernilai 1 jika slot ke-k sudah terpesan, 0 jika masih kosong

h. Waktu Tunggu Efektif dalam Time Slot

Waktu tunggu efektif pelanggan ke-i dalam sistem Time Slot Scheduling didefinisikan sebagai selisih antara waktu mulai slot yang dipesan dengan waktu pemesanan dilakukan:

$$WT_{eff(i)} = S(k_i) - AT(i)$$

Simbol:	Keterangan:
---------	-------------

$WT_{eff(i)}$	Waktu tunggu efektif pelanggan ke-i sejak memesan hingga dilayani
$S(k_i)$	Waktu mulai slot yang dipesan oleh pelanggan ke-i
$AT(i)$	Waktu pemesanan (booking time) pelanggan ke-i

2.2.3. FCFS dan Time Slot Scheduling

Dalam sistem pemesanan barbershop yang diusulkan, kedua algoritma dikombinasikan melalui mekanisme berikut: Time Slot Scheduling berfungsi sebagai kerangka struktural yang membagi waktu operasional ke dalam slot-slot layanan, sedangkan FCFS bertindak sebagai aturan prioritas yang menentukan urutan hak pemesanan slot. Integrasi ini dapat diformulasikan sebagai:

$$Assign(i) = \min\{k : A(k) = 1 \text{ AND } k \geq Q(i)\}$$

$$AWT_{sistem} = \left(\frac{1}{n}\right) \times \sum WT_{eff(i)}$$

Simbol:	Keterangan:
$Assign(i)$	Fungsi penugasan slot minimum yang tersedia untuk pelanggan ke-i
$\min\{k : \dots\}$	Slot dengan nomor terkecil yang memenuhi kondisi ketersediaan
AWT_{sistem}	Rata-rata waktu tunggu efektif seluruh pelanggan dalam sistem terintegrasi

Dengan implementasi integrasi tersebut, kompleksitas komputasi algoritma secara keseluruhan adalah $O(n \times N_{slot})$, di mana n adalah jumlah pelanggan dan N_{slot} adalah jumlah slot waktu yang tersedia dalam satu periode operasional.

Mekanisme integrasi tersebut bekerja dengan cara memvalidasi setiap permintaan booking yang masuk terhadap status ketersediaan slot secara real-time. Ketika sebuah slot waktu berstatus tersedia (available), maka permintaan booking yang masuk akan langsung diterima dan status slot tersebut diubah menjadi terisi (booked). Sebaliknya, apabila slot yang diminta telah berstatus terisi, sistem akan menolak permintaan booking tersebut dan mengarahkan pelanggan untuk memilih slot waktu lain yang masih tersedia. Penentuan pelanggan mana yang berhak atas suatu slot dilakukan berdasarkan prinsip FCFS, yaitu berdasarkan urutan waktu kedatangan permintaan (timestamp) pada server, bukan berdasarkan waktu layanan yang dipilih.

Sebagai ilustrasi, apabila terdapat dua pelanggan, yaitu Fajar dan Al, yang sama-sama hendak memesan slot layanan pada pukul 10.00, maka sistem akan memproses kedua permintaan tersebut berdasarkan urutan kedatangannya di server. Apabila permintaan booking Fajar tercatat masuk lebih awal dibandingkan permintaan Al, meskipun selisih waktunya hanya beberapa detik, maka slot pukul 10.00 akan dialokasikan kepada Fajar dan status slot tersebut berubah menjadi terisi. Ketika permintaan Al diproses setelahnya, sistem akan mendeteksi bahwa slot pukul 10.00 sudah tidak tersedia, sehingga booking Al ditolak dan sistem menampilkan notifikasi agar Al memilih slot waktu alternatif, misalnya pukul 10.30 atau 11.00. Mekanisme ini menjamin bahwa tidak akan terjadi bentrok jadwal antar pelanggan

pada slot dan barber yang sama, sekaligus memastikan keadilan dalam antrian karena penentuan hak atas slot murni didasarkan pada urutan waktu pemesanan.

2.3. *Hypertext Preprocessor (PHP)*

PHP merupakan salah satu bahasa pemrograman yang dirancang untuk dijalankan dalam konteks halaman web, khususnya dalam pengolahan data berbasis internet. Secara terminologis, PHP merupakan akronim dari Hypertext Preprocessor, yaitu bahasa pemrograman sisi server (server-side) yang bersifat open source dan dapat digunakan secara bebas. PHP terintegrasi langsung dengan HTML dan dijalankan pada sisi server, menjadikannya pilihan yang efisien dalam pengembangan aplikasi web dinamis. Bahasa ini dikembangkan secara khusus oleh dan untuk para pengembang web, dengan kemampuan mendukung berbagai jenis basis data secara langsung tanpa memerlukan instalasi konektor tambahan, berbeda dengan beberapa bahasa pemrograman lain seperti Java (Scrum et al., 2019).

2.4. *HyperText Markup Language (HTML)*

HyperText Markup Language (HTML) merupakan bahasa markup yang digunakan untuk merancang struktur dan elemen-elemen dalam sebuah halaman web. Fungsinya adalah untuk memungkinkan penyajian dokumen secara daring melalui jaringan internet. Dalam praktiknya, setiap elemen dalam dokumen HTML umumnya ditulis menggunakan pasangan tag, yaitu tag pembuka dan tag penutup, di mana tag penutup dibedakan dengan penambahan garis miring (/) sebelum nama tag (Khairunnisa Raihani, 2025)

2.5. *My Structure Query Language (MySQL)*

MySQL merupakan salah satu sistem manajemen basis data yang banyak dimanfaatkan dalam pengembangan aplikasi web dinamis. MySQL mendukung integrasi dengan berbagai bahasa pemrograman, khususnya PHP, yang menjadikannya populer di kalangan pengembang web. MySQL menggunakan *Structured Query Language (SQL)* sebagai bahasa query utamanya, dengan sintaks yang sederhana serta penggunaan escape character yang selaras dengan PHP. (Khairunnisa Raihani, 2025).

2.6. *Javascript*

Pengembangan website modern umumnya menggunakan kombinasi HTML, CSS, dan *JavaScript*. HTML berfungsi sebagai struktur dasar halaman, CSS bertanggung jawab dalam mengatur tampilan visual, sedangkan *JavaScript* digunakan untuk menambahkan interaktivitas serta logika pada aplikasi web. *JavaScript* adalah bahasa script dinamis yang dapat disisipkan pada hampir seluruh bagian halaman web untuk menambahkan fungsionalitas dan membuat halaman HTML statis menjadi interaktif (Polgan et al., 2025).

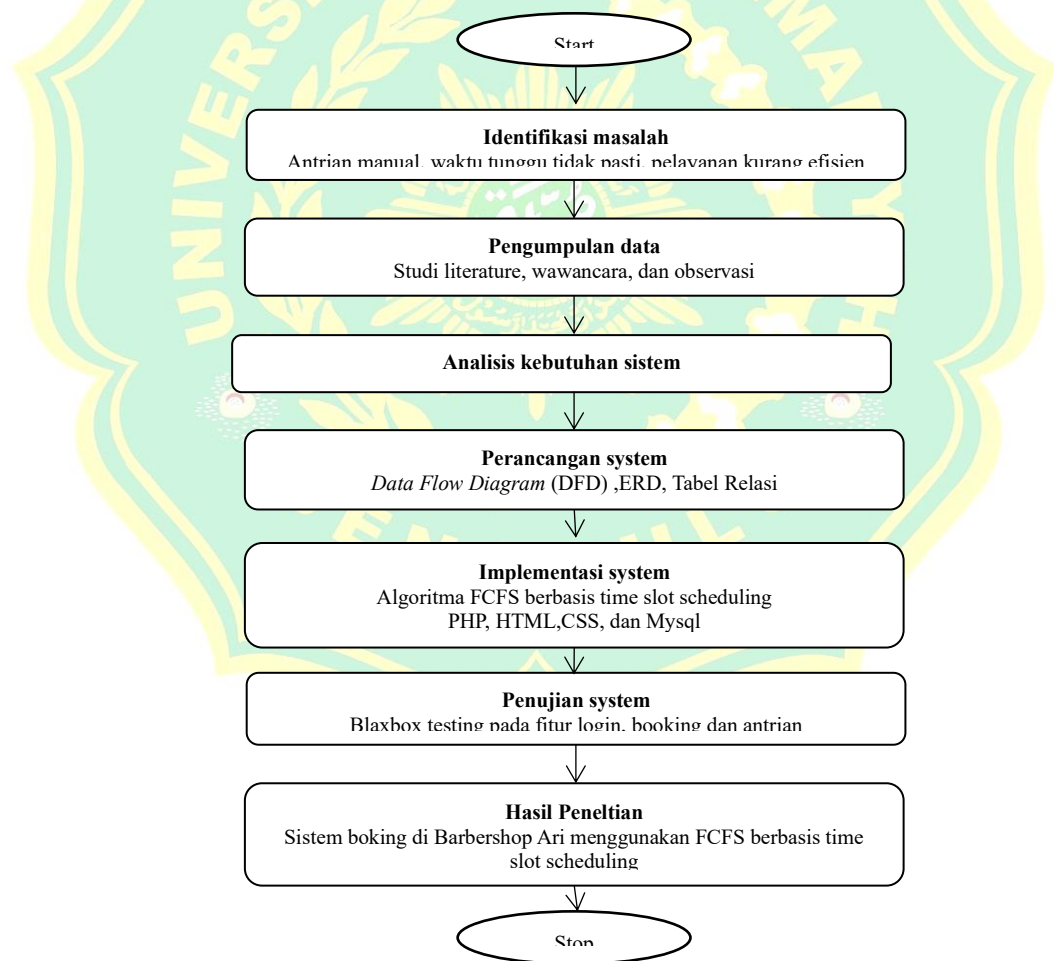
2.7. *Metode Waterfall*

Waterfall adalah model pengembangan perangkat lunak tradisional yang bersifat sekuensial. Prosesnya berjalan berurutan mulai dari analisis kebutuhan sistem, perancangan, implementasi, pengujian, hingga tahap perawatan. Model ini cocok digunakan untuk proyek dengan kebutuhan yang sudah jelas dan terdefinisi sejak awal (Jazuli et al., 2026).

2.8. Data Flow Diagram (DFD)

DFD digunakan untuk menggambarkan aliran data secara logis dalam sistem. Diagram ini memperlihatkan bagaimana data berpindah dari input, melalui proses pengolahan, penyimpanan di database, hingga menghasilkan output berupa laporan kehadiran yang dapat digunakan oleh pihak manajemen. Pembuatan DFD bertujuan untuk mempermudah pemahaman alur data serta memastikan semua proses bisnis dalam sistem terdokumentasi dengan baik.

2.9. Framework Penelitian



Gambar 2.1 Framework Penelitian