

BAB II LANDASAN TEORI

2.1 Mobile Reservasi

Mobile reservasi merupakan suatu perjanjian pemesanan tempat antara 2 pihak atau lebih dilakukan dengan via *mobile* baik saat ini atau hari sebelumnya. Perjanjian atau pemesanan tempat tersebut dapat berupa perjanjian atas pemesanan suatu ruangan, kamar, tempat duduk dan lainnya pada waktu/ periode tertentu dan disertai dengan produk jasanya. (Ningtias, 2017)

2.2 Shortest Job First

Algoritma *Shortest Job First Scheduling* sangat optimal, karena memberikan rata-rata waktu lebih kecil dibandingkan algoritma penjadwalan yang lain dengan cara memindahkan *job-job* pendek di depan *job-job* yang panjang, sehingga akan mengurangi waktu tunggu. (Bambang, 1997)

Untuk memperjelasnya dapat dilihat di contoh berikut.

Misalkan ada empat job yaitu A, B, C, D masing-masing waktu kedatangan sama, yaitu pada $t = 0$, dan lama proses *job* berturut-turut : 8, 4, 4, 4.

Tabel 2.1. Contoh Penjadwalan *Shortest Job First* Jika $t = 0$

Proses	Waktu
A	8
B	4
C	4
D	4

Jika urutan pengerjaannya :

- a. *Job* A, B, C, D

b. *Job* B, C, D, A

Maka proses pengerjaannya adalah sebagai berikut:

(a)

8	4	4	4
A	B	C	D

(b)

4	4	4	8
B	C	D	A

Dengan pengerjaan *Job* berdasarkan urutan (a), maka berturut-turut waktu yang dibutuhkan untuk proses A, B, C, D adalah 8, 12, 16, 20 sehingga dapat dihitung waktu rata-rata = $(8+12+16+20) / 4 = 14$.

Bila *job* yang dikerjakan berdasarkan (b), yaitu dengan *shortest job first*, maka waktu yang dibutuhkan untuk proses B, C, D, A adalah 4, 8, 12, 20 atau rata-rata = $(4+8+12+20) / 4 = 11$.

Pada algoritma ini setiap proses yang ada di *ready queue* akan diesksekusi berdasarkan *burst time* yang pendek untuk setiap proses dan karena hal tersebut maka *waiting time* rata-ratanya juga menjadi pendek, sehingga dapat dikatakan bahawa algoritma ini adalah algoritma yang optimal.

2.3 Hemodialisa

2.3.1 Pengertian Hemodialisa

Hemodialisa adalah dialisis yang dilakukan diluar tubuh, darah dikeluarkan dari tubuh melalui sebuah kateterarteri, kemudian masuk ke dalam sebuah mesin besar, di dalam mesin tersebut terdapat dua ruang yang dipisahkan oleh sebuah membran semipermeabel. Darah dimasukkan ke salah satu ruang, sedangkan ruang yang lain diisi oleh cairan per dialisis dan diantara keduanya akan terjadi difusi. Darah dikembalikan ke tubuh melalui sebuah pirau vena. Hemodialisa memerlukan waktu selama 3 – 5 jam dan dilakukan sekitar 3x dalam seminggu. Pada akhir interval 2 – 3 hari diantara terapi, keseimbangan garam, air dan pangkat hidrogen (PH) sudah tidak normal lagi dan penderita biasanya merasa

tidak sehat. hemodialisa adalah proses dimana terjadi difusi partikel terlarut (solut) dan air secara pasif melalui satu kompartemen cair yaitu darah dan menuju kompartemenlainnya yaitu cairan dyalisan melalui membran semipermeabel dalam dialiser. (Mardyaningsih, 2014)

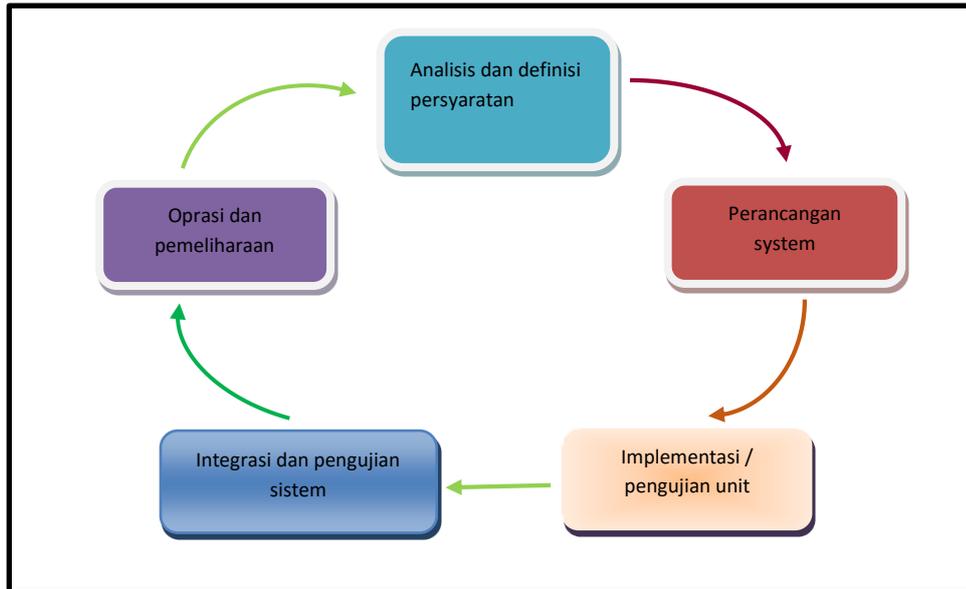
2.3.2 Tujuan Aktivitas Hemodialisa

Tujuan dilaksanakannya aktivitas hemodialisa adalah untuk menghilangkan gejala yaitu mengendalikan uremia, kelebihan cairan dan ketidakseimbangan elektrolit yang terjadi pada pasien penyakit ginjal tahap akhir. Selain itu, memungkinkan kehidupan untuk dijalani dan memberikan kehidupan yang layak untuk dijalani, tidak hanya menjaga pasien agar tetap hidup dengan dialisis.

2.4 Metode Pengembangan Perangkat Lunak

Prototipe bertindak sebagai mekanisme untuk mengidentifikasi spesifikasi-spesifikasi kebutuhan perangkat lunak. Jika suatu prototipe yang dapat digunakan akan dikembangkan, program yang sudah ada sebelumnya dapat digunakan. Dengan menggunakan metode pengembangan perangkat lunak prototipe, pengguna dapat langsung melihat sistem yang kelak akan mereka pakai dan para pengembang dapat mengembangkan sistem atau perangkat lunak dengan segera. (Pressman, 2012)

Metode penelitian ini melakukan rekayasa perangkat lunak, yang digunakan adalah model prototipe seperti pada gambar 2.1 berikut ini :



Gambar 2.1 Sistem Model *Prototype*

Penjelasan dari gambar 2.1 adalah sebagai berikut :

1) Analisis dan Definisi Persyaratan

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mem-spesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2) Perancangan Sistem

Proses multistep yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3) Implementasi dan Pengujian Unit

Perancangan sistem dan perangkat lunak ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4) Integrasi dan Pengujian Sistem

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5) Operasi dan Pemeliharaan

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.5 UML (Unified Modeling Language)

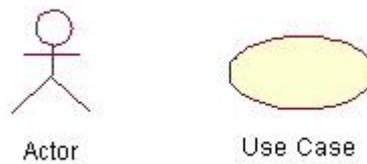
Unified Metodologi Language (UML) adalah salah satu standar bahasa yang digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek. (Rosa & Shalahuddin, 2016)

2.5.1 Use Case Diagram

Use Case Diagram didefinisikan sebagai gambaran bagaimana seorang user berinteraksi dengan sistem dan mendefinisikan langkah-langkah yang diperlukan untuk mendapatkan hasil yang spesifik. (Rosa & Shalahuddin, 2016)

Ada dua element utama pada *use case Diagram* yaitu pendefinisian actor dan *use case*.

- a. *Actor* yaitu sebagai figur utama yang direpresentasikan sebagai seorang aktor yang dasosiasikan dengan satu kategori dari user atau elemen interaksi yang lain.
- b. *Use Case* digambarkan dengan bentuk oval. Dan digunakan untuk bertukar pesan yang disediakan oleh sistem.



Gambar 2.2 Elemen Utama Pada *Use Case Diagram*

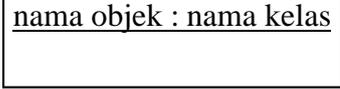
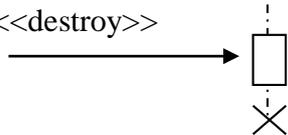
2.5.2 Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. (Rosa & Shalahuddin, 2016)

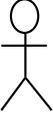
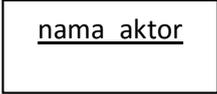
Dalam membuat *Sequence Diagram* harus memiliki acuan untuk melihat scenario yang ada pada *use case*. Banyaknya diagram *sequen* yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang dimiliki pada setiap proses. Sehingga semakin banyak *use case* yang didefinisikan maka *Sequence Diagram* yang harus dibuat juga semakin banyak.

Simbol-simbol *Sequence Diagram* dapat dilihat pada Tabel 2.2 berikut :

Tabel 2.2 Simbol *Sequence* Diagram

Simbol	Deskripsi
Garis hidup/ <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
Objek 	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
Pesan tipe create <code><<create>></code> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
Pesan tipe call 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan tipe send 1 : masukan 	Merupakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
Pesan tipe return 1 : keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe destroy <code><<destroy>></code> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri

Tabel 2.2 Simbol *Sequence* Diagram (Lanjutan)

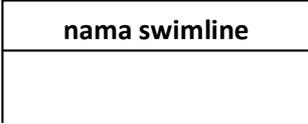
Simbol	Deskripsi
	sebaliknya jika ada create maka ada destroy.
<p>Aktor</p>  <p>atau</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>

2.5.3 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (alir kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa & Shalahuddin, 2016)

Simbol-simbol yang ada pada diagram aktivitas dapat dilihat pada Tabel 2.3 berikut ini :

Tabel 2.3 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah sistem awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.

2.6 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Kebutuhan basis data dalam sistem informasi meliputi :

1. Memasukkan, menyimpan, dan mengambil data.
2. Membuat laporan berdasarkan data yang telah disimpan.

2.7 DBMS

DBMS (Database Management System) atau dalam bahasa Indonesia sering disebut sebagai sistem Manajemen Basis Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi syarat minimal sebagai berikut:

1. Menyediakan fasilitas untuk mengelola akses data
2. Mampu menangani integritas data
3. Mampu menangani akses data yang dilakukan
4. Mampu menangani backup data

DBMS sudah mulai berkembang sejak tahun 1960-an. Kemudian sekitar tahun 1970-an mulai berkembang teknologi relational DBMS yaitu DBMS berbasis model. (Rosa & Shalahuddin, 2016)

Berikut ini adalah contoh DBMS versi open source yang cukup berkembang dan paling banyak digunakan saat ini adalah sebagai berikut:

- a. MySQL
- b. PostgreSQL
- c. Firebird
- d. SQLite

2.8 Normalisasi

Istilah Normalisasi berasal dari E. F. Codd, salah seorang perintis teknologi basis data. Selain dipakai sebagai metodologi tersendiri untuk menciptakan struktur tabel relasi dalam basis data (dengan tujuan untuk mengurangi kemubaziran data), normalisasi terkadang hanya dipakai sebagai perangkat verifikasi terhadap tabel-tabel yang dihasilkan oleh metodologi lain (misalnya E-R). Normalisasi memberikan panduan yang sangat membantu bagi pengembang untuk mencegah penciptaan struktur tabel yang kurang fleksibel atau mengurangi kefleksibelan. (Rosa & Shalahuddin, 2016)

Normalisasi adalah suatu teknik dalam mendesain suatu database yang dimulai dengan memeriksa hubungan (disebut *functional dependency*) antar atribut.

2.9 SQL (Structured Query Language)

(S. Rosa & M. Shalahuddin, 2016) SQL (Struktur Query Language) adalah bahasa yang digunakan untuk mengelola data RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. SQL mulai berkembang pada tahun 1970an. SQL mulai digunakan sebagai standar yang resmi pada tahun 1986 oleh ANSI (American National Standards Institute) dan pada tahun 1987 oleh ISO (International Organization for Standardization) dan disebut sebagai SQL-86. (Rosa & Shalahuddin, 2016)

Pada perkembangannya, SQL beberapa kali dilakukan revisi. Berikut ini sejarah perkembangan SQL sampai saat ini:

Tabel 2.4 SQL Tipe

No	Tahun	Nama
1	1986	SQL-86
2	1989	SQL-89
3	1992	SQL-92
4	1999	SQL:1999
5	2003	SQL:2003
6	2006	SQL:2006
7	2008	SQL:2008
8	2011	SQL:2011

Meskipun SQL diadopsi dan diacu sebagai bahasa standar oleh hampir sebagian RDBMS yang beredar saat ini, tetapi tidak semua standar yang tercantum dalam SQL diimplementasikan oleh seluruh DBMS tersebut. Sehingga kadang-kadang ada perbedaan perilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal query yang dimasukkan sama.

2.10 Penelitian Terdahulu

Tabel 2.4 berikut ini merupakan penelitian terdahulu yang menerapkan sistem *M-Reservation*.

Tabel 2.5 Penelitian Terdahulu

No	Nama	Judul	Terbit/ Tahun	Uraian
1.	Timotius Witono, Raphael Susanto	Aplikasi Pemesanan Tiket Bioskop Berbasis Mobile.	Sistem Informasi Universitas Kristen Maranatha / 2012	Semakin banyaknya masyarakat yang mencari hiburan dengan menonton film di bioskop saat ini, maka pemesanan tiket film di bioskop menjadi masalah yang penting. Ada sebagian masyarakat yang rela mengantri dari pagi untuk mendapat tiket film di sebuah bioskop. Dengan pemesanan tiket bioskop yang semakin hari semakin banyak maka diperlukan sistem untuk dapat mempermudah masyarakat dalam memesan tiket film di bioskop. Saat ini pemesanan tiket bioskop sudah dapat dilakukan secara <i>online</i> dengan membuka <i>mobile</i> yang mereka punya.

2.	Adi Putra Nugraha, Kodrat Iman Satoto, Kurniawan Teguh Martono.	Aplikasi Pemesanan Makanan Berbasis Mobile pada Rumah Makan Lek Nonong.	Sistem Komputer Universitas Diponegoro / 2014	Selama ini Rumah Makan Lek Nonong masih menggunakan cara pengerjaan manual dalam pelayanannya, belum menggunakan aplikasi terkomputeriasi atau aplikasi berbasis <i>mobile</i> . Oleh sebab itu, perlu dikembangkan suatu aplikasi berbasis <i>mobile</i> yang dapat membantu proses pelayanan maupun proses rekap data untuk laporan transaksi yang dapat digunakan pada Rumah Makan Lek Nonong. Aplikasi untuk Rumah Makan ini dapat menjadi salah satu solusi untuk memudahkan proses kerja dilingkungan Rumah Makan Lek Nonong, khususnya dalam proses pemesanan menu dan laporan transaksi. Dengan adanya suatu aplikasi yang dapat melakukan manajemen laporan dari

				transaksi transaksi terjadi pada Rumah Makan.
3.	Monika Santika, Seng Hansun	Implementasi Algoritma Shortest Job First dan Round Robin pada Sistem Penjadwalan Pengiriman Barang	Teknik Informatika Universitas Multimedia Nusantara / 2014	Masalah utama pada penelitian ini adalah antrian pada pengiriman barang di sebuah e-commerce yang dilakukan sesuai antrian pemesanan. Tetapi hal tersebut akan menjadi tidak efisien dan akan membuat orang yang hanya memesan barang yang sedikit, menunggu terlalu lama karena pembeli yang memesan barang cukup banyak. Pengiriman barang juga terkait dengan perusahaan-perusahaan besar yang memerlukan distributor.
4.	Ayu Ningtias	Membangun <i>M-Reservation</i> Pada Pecel Lele Haji Fuad Bandar Lampung.	Teknik Informatika IBI Darmajaya / 2017	Aplikasi reservasi dan pembayaran secara online pada rumah makan Haji Fuad Bandar Lampung. Masalah utamanya adalah antrian panjang pelanggan yang sering kali harus mengantri lama untuk memesan dan membayar

				makanan mereka. Selain itu aplikasi ini juga menjadi sarana promosi dan pencatatan bagi rumah makan Haji Fuad.
--	--	--	--	--