URANUS : Jurnal Ilmiah Teknik Elektro, Sains dan Informatika
Volume 2 No. 4 Desember 2024
e-ISSN : 3031-996X, p-ISSN 3031-9951 Hal 41-70
DOI: https://doi.org/10.61132/uranus.v2i4.469
Available online at: https://journal.arteii.or.id/index.php/Uranus

Penerapan Model *Incremental Delivery* pada Pengembangan Aplikasi Sistem Informasi Hotel Cipta Jakarta

Vilianty Rafida
Teknik Informatika, STMIK Widya Cipta Dharma, Indonesia

Alamat: Jl. Prof. M. Yamin No. 25 Samarinda Kalimantan Timur 75123, Indonesia
Korespondensi penulis: vilianty@wicida.ac.id

Abstract*. Along with the development of the times, we see many hotels experiencing changes. A change that is rarely reviewed as a news topic is the software overhaul that occurs in hotels from offline to online systems. Lay people only see changes in hotels from a visible perspective, without having to know whether the hotel's system and system applications have changed. One of the factors in system development at Hotel Cipta is the need for hotel management to prioritize the development of system applications from offline to online systems, even web-based. It is not easy to overhaul an existing system application compared to creating a new system application. There are several software engineering methods and models that are commonly used to analyze system needs and design systems in the software development process. One of them is the incremental delivery model. This makes it interesting to try to apply the incremental delivery model to the development of the Hotel Cipta information system application by taking place at the Hotel Cipta on KH. Wachid Hasyim Street, Central Jakarta. In this study, the object of research is the Front Office department.*

Keywords*: Development; Hotel; Incremental.*

Abstrak. Seiring dengan perkembangan zaman, kita banyak melihat hotel mengalami perubahan. Perubahan yang jarang sekali diulas menjadi topik berita adalah perombakan perangkat lunak yang terjadi di hotel dari sistem *off line* menjadi *on line*. Orang awam hanya melihat perubahan hotel dari sudut pandang kasat mata saja, tanpa harus mengetahui apakah sistem dan aplikasi sistem hotelnya berubah. Salah satu faktor pengembangan sistem di Hotel Cipta adalah kebutuhan manajemen hotel menuntut prioritas pada pengembangan aplikasi sistem dari sistem *off line* menjadi menjadi sistem *on line*, bahkan berbasis web. Sungguh tidak mudah merombak suatu aplikasi sistem yang sudah ada dibanding dengan membuat aplikasi sistem yang baru. Ada beberapa metode dan model rekayasa perangkat lunak yang umum digunakan untuk menganalisis kebutuhan-kebutuhan sistem dan merancang sistem dalam proses pembuatan perangkat lunak. Salah satunya adalah model *incremental delivery*. Hal ini menjadi menarik guna mencoba untuk menerapkan model *incremental delivery* pada pengembangan aplikasi sistem informasi Hotel Cipta dengan mengambil tempat di Hotel Cipta jalan KH. Wachid Hasyim Jakarta Pusat. Dalam penelitian ini objek penelitiannya departemen *Front Office*.

Kata kunci: Incremental; Pengembangan; Hotel.

1. LATAR BELAKANG

Hotel Cipta didirikan pada bulan Maret 1991, terletak di jalan KH. Wachid Hasyim Jakarta Pusat, dengan kap asitas 48 kamar dan termasuk kategori hotel bintang tiga. Pada awal hotel dibuka sampai dengan Agustus tahun 2023, sistem informasi yang digunakan pada hotel ini masih merupakan sistem pemrosesan transaksi, yaitu sistem informasi yang terkomputerisasi guna mencatat dan memproses data dalam jumlah besar untuk transaksi bisnis rutin seperti memasukkan daftar tamu dan inventarisasi. Selain itu, sistem masih belum terhubung dengan sistem jaringan.

Received: Oktober 03, 2024; Revised: Oktober 20, 2024; Accepted: November 05, 2024; Online Available: November 08, 2024

Akhir tahun 2023, kebutuhan manajemen hotel menuntut prioritas pada pengembangan aplikasi sistem dari sistem *off line* menjadi sistem *on line* bahkan berbasis web agar setiap transaksi tamu yang memakai fasilitas hotel dapat dihitung dan dicetak dengan cepat secara otomatis. Hal ini, menyebabkan aplikasi sistem informasi yang ada harus diperbaiki dan dikembangkan. Berhubung aplikasi sistem yang ada tidak dapat digunakan lagi karena tidak mendukung sistem jaringan, otomatis aplikasi sistem dirombak total menjadi suatu sistem yang baru.

Perlunya pengembangan sistem disebabkan oleh beberapa faktor, antara lain: sistem yang lama tidak dapat beroperasi sesuai dengan yang diharapkan, instruksi dari pemilik hotel dan manajemen hotel yang menginginkan perubahan sistem, serta pertumbuhan organisasi hotel meningkatkan volume pengolahan data dan kebutuhan informasi.

Fasilitas hotel yang pada awalnya hanya menangani kamar, laundry, drug store dan restoran, berkembang seperti penambahan *outlet spa*, sehingga harus ada perubahan fungsionalitas sistem pada aplikasi yang disesuaikan dengan penambahan fasilitas tersebut.

Sungguh tidak mudah merombak suatu aplikasi sistem yang sudah ada dibanding dengan membuat aplikasi sistem yang baru. Ada beberapa model dan metode rekayasa perangkat lunak yang umum digunakan untuk menganalisis kebutuhan-kebutuhan sistem dan merancang sistem dalam proses pembuatan perangkat lunak. Salah satunya adalah model *incremental delivery*.

Hal ini menjadi menarik guna mencoba untuk menerapkan model incremental delivery pada pengembangan aplikasi sistem informasi Hotel Cipta dengan objek penelitian khususnya pada departemen *Front Office*.

2. KAJIAN TEORITIS

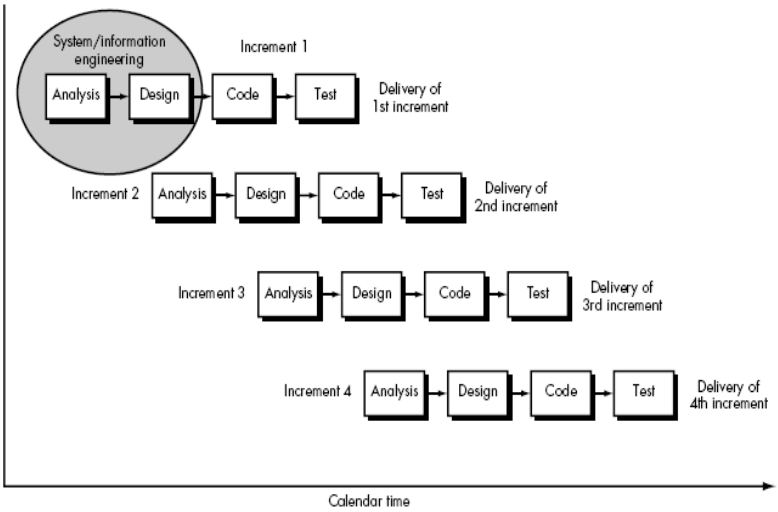
Model *Incremental*

Model *incremental* adalah kombinasi dari model linier (model air terjun) dan model prototipe iterative. Pada tahap inkremental, jika inkremen pertama masih tidak sesuai dengan apa yang diinginkan pengguna, maka langkah-langkah iteratif dilakukan secara berurutan sehingga fungsi sistem seperti yang diinginkan oleh pengguna (Rachman et al., 2020).

Pengguna pada model ini mengutamakan persyaratan sistem yang akan dibuat sehingga harus dilakukan beberapa iterasi dalam perkembangannya agar fungsinya sesuai dengan yang diinginkan oleh pengguna (Rachman et al., 2020).

Tahap Kebutuhan adalah tahap persyaratan awal yang digunakan peneliti untuk mendapatkan kebutuhan fungsional dari aplikasi yang sedang dikembangkan. Tahap Desain adalah tahap desain sistem aplikasi. Fase Coding adalah fase pengembangan berdasarkan hasil tahap desain. Tahap Pengujian adalah tahap dimana modul program, program kelas diuji dan sesuai dengan kebutuhan atau tidak, kami menggunakan model pengujian kotak hitam. Tahap Implementasi adalah tahap implementasi dari hasil pembuatan seluruh aplikasi (Rachman et al., 2020).

Model *incremental* merupakan pengembangan sistem model air terjun di mana menuntut user untuk membuat seperangkat persyaratan sebelum perancangan dimulai dan perancangan sistem untuk membuat strategi perancangan khusus sebelum implementasi. Perubahan persyaratan selama pengembangan menuntut pengerjaan ulang dari persyaratan, perancangan dan implementasi. Namun demikian keuntungan dari model incremental adalah bahwa model tersebut merupakan model manajemen yang sederhana dan pemisahan perancangan dan implementasi harus menghasilkan sistem yang berbobot yang dapat berubah (Pressman, 2010).



Gambar 1. Model *Incremental Delivery* Metode Pressman

Pengembangan incremental dikenal sebagai suatu alternatif dari metoda pengembangan perangkat lunak *waterfall* dengan urutan yang tersusun seksama dari kebutuhan, analisis, perancangan dan tahap-tahap pengembangan lainnya. Bagaimanapun, pendekatan *incremental* bervariasi aspek seperti yang direkomendasikan seperti panjang iterasi, sejumlah pekerjaan spesifikasi, atau penekanan pada umpan balik dan pengembangan *adaptation-driven*. Ada banyak istilah yang membingungkan antara

pengembangan iterative, pengembangan incremental dan pengembangan berversi yang digunakan secara tidak konsisten di dalam literatur (Mohagheghi, 2004:29).

Inkremen

Definisi inkremen '*increment*' dapat diartikan sebagai berikut:

a. *Feature increments*: fungsi dan ciri khas ditambahkan pada setiap inkremen.
b. *Normal/error increments*: Kasus normal sederhana dikembangkan lebih dulu. Sedangkan yang lebih kompleks ditambahkan ke kasus –kasus dalam iterasi yang berturut-turut, misalnya penanganan kesalahan.
c. *Separate system function increments*: Contohnya dalam area telekomunikasi dimulai dikembangkan dulu. Perintah, penanganan lalu lintas dan fungsionalitas lain pengguna ditambahkan.
d. *Component-oriented increments*: Cobra menetapkan komponen dan bagiannya untuk inkremen agar menyerahkan versi exe dan secara bertahap dalam pada realisasi komponen (Atkitson et al, 2002).

Perbedaan yang utama antara pendekatan *component-oriented increments* dan *feature increments* adalah kedua-duanya dikombinasikan dengan *Normal/error increments*. *System function increments* dipertimbangkan sebagai suatu *feature increments*.

Atkinson et al. menyatakan bahwa arsitektur perangkat lunak itu tidak sama dengan pengembangan *incremental* sebab arsitektur seharusnya menahan kebutuhan yang tidak fungsional, dengan begitu suatu pandangan sistem harus dikembangkan. Sebagai perbaikan, mereka mengusulkan *component-oriented increments*. Kerugian dari pendekatan ini bahkan lebih tergantung pada kebutuhan dan pekerjaan desain berlebihan di awal pengembangan. Selain itu, tidak semua kebutuhan non fungsional mungkin untuk menentukan komponen tunggal dan tidak ada fungsi yang lengkap dibangun pada awal inkremen.

Incremental Delivery

Incremental delivery dalam proses pengembangan sistem rekayasa perangkat lunak dinyatakan sebagai berikut:

a. Dibandingkan dengan sistem sebagai delivery tunggal, pengembangan dan *delivery* dipecah ke dalam bagian delivery tambahan fungsi.
b. Kebutuhan user diprioritaskan dan kebutuhan prioritas yang paling tinggi adalah termasuk tahap awal proses pengembangan ini.

c. Pada saat pengembangan suatu inkremen fungsi dimulai, kebutuhan-kebutuhan lain dihentikan sementara dan dapat dilanjutkan bertahap.

Sistem Informasi dan Sistem Informasi Hotel

Sistem informasi merupakan sistem yang berisi jaringan sistem pengolahan data, yang dilengkapi dengan kanal-kanal komunikasi yang digunakan dalam sistem organisasi data. Elemen proses dari sistem informasi antara lain mengumpulkan data, mengelola data yang tersimpan, menyebarkan informasi (Humris, n.d.).

Di dalam sistem informasi, manusia berinteraksi dengan manusia, manusia, manusia berinteraksi dengan komputer, dan komputer berintera(Humris, n.d.)ksi dengan komputer lain. Di dalam sistem informasi, data, informasi dan/atau pengetahuan mengalir dibawa oleh dokumen atau media komunikasi elektronik, seperti telepon atau jaringan komputer.

Sistem informasi dalam suatu pemahaman yang sederhana dapat didefinisikan sebagai satu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa. Sistem informasi memuat berbagai informasi penting mengenai orang, tempat, dan segala sesuatu yang ada di dalam atau di lingkungan sekitar organisasi. Informasi sendiri mengandung suatu arti yaitu data yang telah diolah ke dalam suatu bentuk yang lebih memiliki arti dan dapat digunakan untuk pengambilan keputusan. Data sendiri merupakan fakta-fakta yang mewakili suatu keadaan, kondisi, atau peristiwa yang terjadi atau ada di dalam atau di lingkungan fisik organisasi. Data tidak dapat langsung digunakan untuk pengambilan keputusan, melainkan harus diolah lebih dahulu agar dapat dipahami, lalu dimanfaatkan dalam pengambilan keputusan. Informasi harus dikelola dengan baik dan memadai agar memberikan manfaat yang maksimal. Penerapan sistem informasi di dalam suatu organisasi dimaksudkan untuk memberikan dukungan informasi yang dibutuhkan, khususnya oleh para pengguna informasi dari berbagai tingkatan manajemen. Sistem informasi yang digunakan oleh para pengguna dari berbagai tingkatan manajemen ini biasa disebut sebagai Sistem Informasi Manajemen .

Sistem informasi mengandung tiga aktivitas dasar di dalamnya, yaitu: aktivitas masukan (*input*), pemrosesan (*processing*), dan keluaran (*output*). Tiga aktivitas dasar ini menghasilkan informasi yang dibutuhkan organisasi untuk pengambilan keputusan, pengendalian operasi, analisis permasalahan, dan menciptakan produk atau jasa baru. Masukan berperan di dalam pengumpulan bahan mentah (*raw data*), baik yang diperoleh dari dalam maupun dari lingkungan sekitar organisasi. Pemrosesan berperan untuk

mengkonversi bahan mentah menjadi bentuk yang lebih memiliki arti. Sedangkan, keluaran dimaksudkan untuk men*transfer* informasi yang diproses kepada pihak-pihak atau aktivitas-aktivitas yang akan menggunakan. Sistem informasi juga membutuhkan umpan balik (*feedback*), yaitu untuk dasar evaluasi dan perbaikan di tahap input berikutnya (Djoko Sutono, 2007:7-8).

Sistem informasi hotel adalah suatu sistem yang ada di dalam hotel yang menangani semua informasi kamar, tamu dan pengunjung hotel serta membuat laporanlaporan untuk pengambilan keputusan yang dilakukan oleh manusia yang dibantu suatu alat berupa mesin komputer.

Sistem informasi hotel dapat diartikan juga sebagai sistem pengolahan data hotel meliputi pengolahan data di antaranya data tamu, data kamar, data *check in*, data *checkout*, juga sebagai manipulasi data, penyimpanan dan data persiapan dokumen untuk pengambilan keputusan yang dilakukan manusia dengan dibantu suatu alat yang berupa mesin komputer (Jehan Marlini Humris, 2010:4).

Analisis Kebutuhan

Sistem informasi merupakan sistem yang Penelitian ini menggunakan model *incremental delivery* dengan objek penelitian terbatas pada lingkungan gedung Hotel Cipta saja terutama sistem transaksi tamu hotel di departemen *Front Office* dan memiliki hasil akhir berupa suatu produk sistem informasi Hotel Cipta. Agar mendapatkan pemahaman tentang pengelolaan data dan informasi pada departemen *Front Office* Hotel Cipta, maka penulis melakukan wawancara terhadap programer yang membuat program sistem informasi Hotel Cipta serta melakukan observasi pada aplikasi sistem informasi Hotel Cipta.

Pengembangan aplikasi sistem informasi hotel dibuat dengan model *incremental delivery* karena dilakukan secara bertahap dalam beberapa modul seperti modul reservasi tamu, modul registrasi tamu, modul transaksi tamu, serta modul laporan akhir tamu hotel. Modul-modul tersebut dapat dibuat dalam bentuk tabel-tabel dengan menggunakan database Paradox atau MySQL sebagai *imbedded database.* Source program dapat dibuat dalam pemrograman Borland Delphi dan sejenisnya.

Data yang dibutuhkan dalam pengembangan sistem informasi hotel adalah:

a. data login admin/user: nama user ID, password, shift;
b. data tamu: nama tamu, alamat, telepon, jenis kelamin, tanggal kedatangan, tanggal kepergian, tipe kamar yang dipilih, tanggal deposit, jumlah deposit.

c. data kamar: nomor kamar, tipe kamar, harga kamar, keterangan

Fungsi dari sistem informasi hotel ini adalah:

a. proses login untuk admin dan user;
b. proses pengelolaan data tamu baik tamu reservasi maupun tamu registrasi, meliputi input, add, edit dan save;
c. proses pengelolaan transaksi tamu;
d. proses pencarian data tamu dan transaksi.

Program sistem yang telah jadi tersebut dijalankan *on line* secara intranet antar departemen dalam satu gedung hotel, dan tidak dipublikasikan ke jaringan internet karena sesuai dengan permintaan pemilik hotel.

3. METODE PENELITIAN

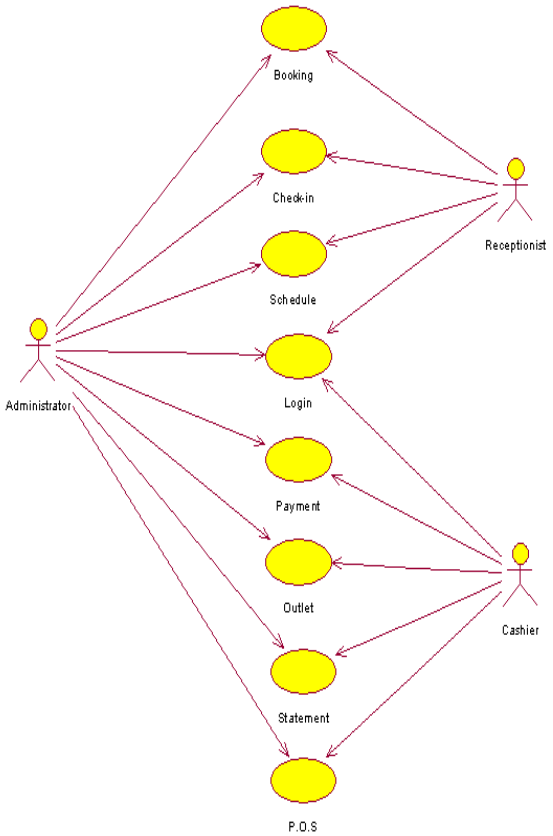
Penelitian ini dirancang menggunakan metode pemodelan UML dan pemodelan user interface.

Pemodelan UML (Unified Modeling Language)

Teknik analisis dan rancangan sistem berorientasi objek yang paling terkenal di seluruh industi Teknologi dan Informasi adalah *Unified Modeling Language (UML)* yang dikembangkan oleh Grady Booch, Ivar Jacobson, dan James Rumbaugh (Kendall&Kendall, 2003:430). UML telah diterima luas saat ini sebagai standar untuk pemodelan objek (Jeffery L. Whitten, dkk, 2004:408). Perangkat lunak yang dirancang untuk penelitian ini dapat dimodelkan ke sebuah diagram tingkah laku dan stuktur, yang terdiri atas diagram model *use case (use case diagram)*, diagram aktivitas *(activity diagram)*, diagram kelas *(class diagram)*, dan *sequence diagram*.

Use Case Diagram

Perangkat lunak sistem informasi Hotel Cipta pada departemen *Front Office* dapat dimodelkan ke dalam sebuah diagram use case berikut:



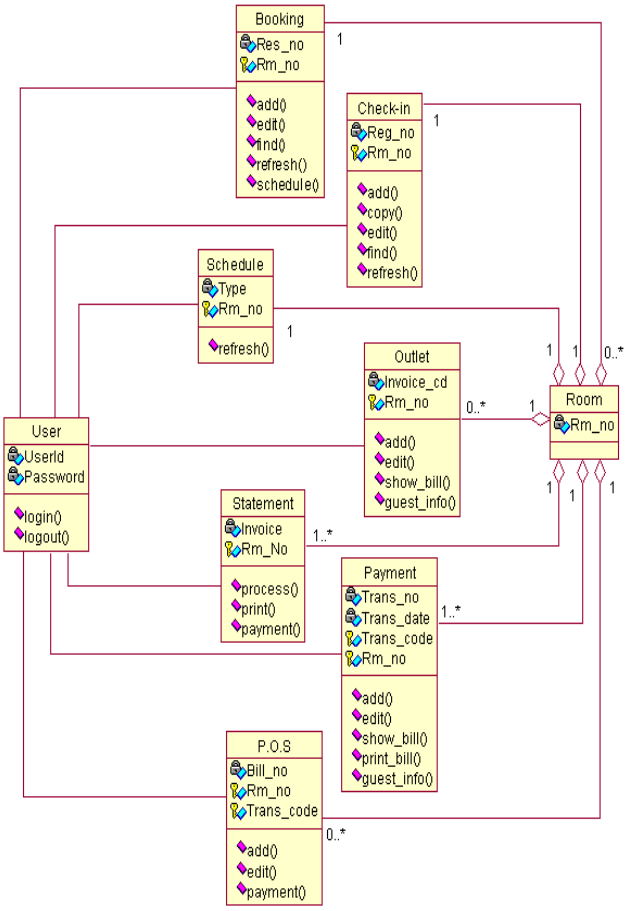
Gambar 2. *Use Case Diagram* pada Menu *Front Office* Hotel Cipta

Pengguna perangkat lunak sistem informasi Hotel Cipta di departemen *Front Office* dapat didefinisikan menjadi tiga aktor yaitu: *Administrator*, *Receptionist* dan *Cashier*.

Diagram *use case* perangkat lunak sistem informasi Hotel Cipta menggambarkan 8 buah *use case*. Penjelasan masing-masing *use case* tersebut dapat dilihat pada tabel berikut ini:

Class Diagram

Class diagram menggambarkan struktur dan deskripsi *class, package, dan object* beserta hubungannya satu sama lain di sistem informasi Hotel Cipta.

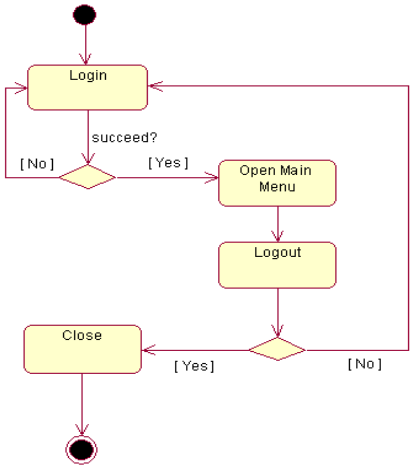


Gambar 3. *Class Diagram* pada Menu *Front Office* Hotel Cipta

Activity Diagram

Pada bagian ini akan dijabarkan skenario use case pada perangkat lunak sistem informasi perhotelan di departemen *Front Office* Hotel Cipta, di mana skenario use case digambarkan secara umum dalam bentuk *Activity Diagram* yaitu diagram yang menjelaskan rincian bagaimana sebuah use case berjalan.

Activity diagram yang dibuat seperti *activity diagram login, activity diagram booking, activity diagram check-in, activity diagram schedule activity diagram payment, activity diagram outlet, activity diagram statement, activity diagram P.O.S.* Salah satu contoh *activity diagram login* sebagai berikut.

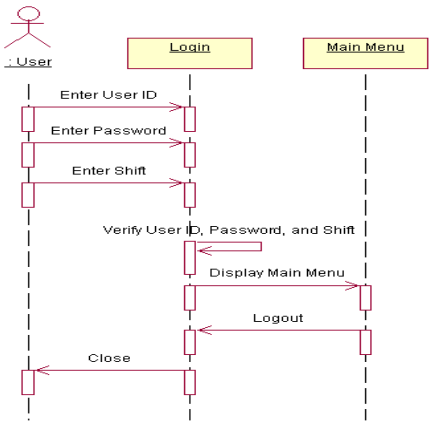


Gambar 4 ***Activity Diagram Login***

Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Dalam membuat *sequence diagram* ini, pengguna (*user*) yang dimaksudkan untuk *Administrator*, *Receptionist*, dan *Cashier*.

Sequence diagram yang dibuat seperti sequence *diagram login, sequence diagram booking, sequence diagram check-in, sequence diagram schedule sequence diagram payment, sequence diagram outlet, sequence diagram statement, sequence diagram P.O.S.* Salah satu contoh *Sequence diagram login sebagai berikut.*



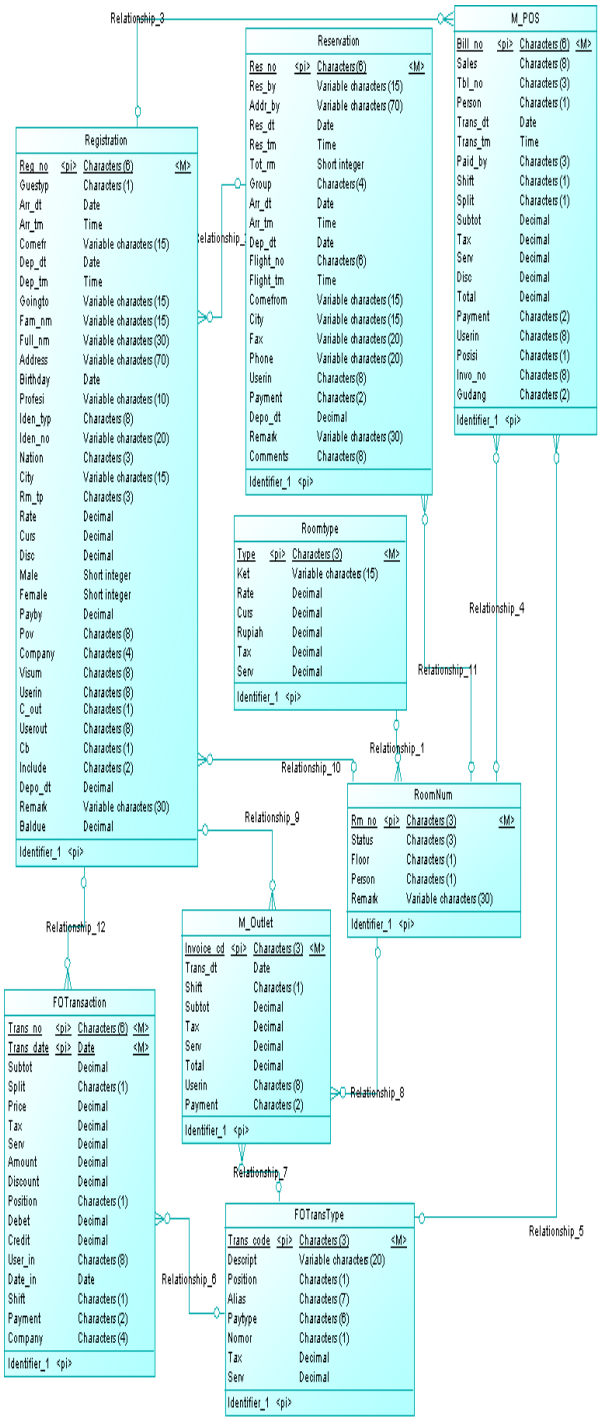
Gambar 5. ***Sequence Diagram Login***

Sequence diagram pada Gambar 5 menunjukkan pengguna (*user*) untuk dapat melakukan akses (*login*) ke dalam *Main Menu.*

***Entity Relationship Diagram* (ERD)**

Salah satu cara penganalisis sistem dengan menggunakan model hubungan-entitas yang diimplementasikan sebagai sebuah database dalam bentuk diagram hubungan-entitas *(entity relationship diagram)* atau ERD (Ibrahim, 2020)

Untuk menggambarkan ERD misalnya dengan program *Sybase Power Designer* versi 15.2 dalam bentuk *Conceptual Data Model* yang dapat dilihat pada Gambar 6.,



Gambar 6. *Conceptual Data Model* pada Menu *Front Office* Hotel Cipta

4. HASIL DAN PEMBAHASAN

Implementasi Sistem

Hasil dari desain yang telah didefinisikan sebelumnya akan ditampilkan pada pembahasan ini. Agar sistem dapat dijalankan dan diinstalasi, maka hasil tersebut terbagi atas spesifikasi perangkat keras dan spesifikasi perangkat lunak. Selanjutnya akan dibahas hasil dari pengujian *white box* dan *black box*. Dalam bab ini juga akan dibahas tentang tempat, waktu dan teknik implementasi.

1. Tempat dan Waktu Implementasi

Setelah perangkat lunak dibuat, maka langkah selanjutnya adalah penerapan hasil aplikasi tersebut terhadap obyek penelitian yaitu pada departemen *Front Office* di Hotel Cipta Jakarta. Pengembangan aplikasi sistem informasi Hotel Cipta pada departemen *Front Office* diimplementasikan secara terintegrasi, sedangkan waktu implementasi memerlukan waktu enam bulan.

2. Teknik Implementasi

Sesuai dengan batasan obyek penelitian dalam ruang lingkup masalah, dianalisis model *incremental delivery* pada pengembangan aplikasi sistem informasi hotel di departemen *Front Office* Hotel Cipta tersebut.

Beberapa tahapan implementasi ini sebagai berikut.

a. Melakukan pengenalan aplikasi ke *user* pada departemen terkait.
b. Menetapkan item-item wawancara yang nantinya dijadikan parameter penilaian penelitian.
c. Melakukan wawancara terhadap programer, serta melakukan observasi pada pengembangan aplikasi sistem informasi Hotel Cipta yang ada khususnya aplikasi yang berhubungan departemen *Front Office*.
d. Implementasi pengembangan aplikasi sistem informasi hotel yang berhubungan dengan departemen *Front Office* untuk pengolahan transaksi tamu hotel.
e. Melakukan survei akhir untuk mendapatkan informasi sistem Hotel Cipta.
f. Melakukan analisis terhadap hasil penelitian.

Spesifikasi Sistem

Spesifikasi sistem untuk menjalankan aplikasi sistem informasi Hotel Cipta khususnya yang berhubungan dengan departemen *Front Office*, sebagai berikut.

1. **Perangkat Keras**

Spesifikasi perangkat keras yang disarankan untuk menjalankan aplikasi ini adalah sebagai berikut:

a. Perangkat Komputer *Server*

Processor Intel

Memori 1 Gb

Harddisk 80 Gb

Sistem Operasi Microsoft Windows

Database Paradox

b. Perangkat Komputer *Client*

Processor Intel

Memori 512 Mb

Harddisk 40 Gb

Sistem Operasi Microsoft Windows

c. Perangkat Jaringan

Switch Hub (Jumlah port tergantung jumlah *Client* yang akan mengakses aplikasi) dan kabel jaringan UTP minimal kategori 5.

2. **Perangkat Lunak**

Spesifikasi perangkat lunak yang dibutuhkan sebagai berikut:

a. Sistem Operasi

Aplikasi sistem informasi Hotel Cipta ini di-*install* pada *PC*/laptop yang memiliki sistem operasi Microsoft Windows.

b. Database

Database yang digunakan untuk aplikasi sistem informasi Hotel Cipta ini adalah Paradoks minimal versi 5 atau bisa dikembangkan MySQL yang di-*install* pada server dengan sistem operasi MS WindowsNT.

3. **Instalasi Sistem**

Untuk menjalankan aplikasi sistem informasi Hotel Cipta ini perlu dilakukan langkah-langkah sebagai berikut:

a. Instalasi Aplikasi

Instalasi aplikasi dilakukan pada perangkat *PC*/laptop dan juga *Server* bila diperlukan.

b. Instalasi Database

Instalasi database Paradox/MySQL database dilakukan pada perangkat *Server*. Setelah proses instalasi database selesai, langkah berikutnya adalah *restore* data ke dalam database Paradox yang telah terinstalasi.

Setelah proses instalasi di atas telah selesai dilakukan, langkah selanjutnya adalah membuka aplikasi dan melakukan setting koneksi aplikasi dari perangkat *PC/*laptop ke *Server*. Setelah setting koneksi aplikasi telah berhasil, maka aplikasi sistem Hotel Cipta telah dapat dilakukan.

Menjalankan Sistem

Untuk dapat menjalankan aplikasi ini dengan cara dimulai dari tampilan antarmuka sebagai berikut.

1. Antarmuka *Login*

Memasukkan sistem otentikasi *login* dengan mengisi *User ID*, *password* dan *shift* (default *shift* bernilai 1). Antarmuka login dapat dilihat pada gambar di bawah ini.

Gambar 7. Tampilan Antarmuka Login

Setelah itu, *initial login User ID*, *password* dan *shift* akan diverifikasi ke tabel pengguna dan juga group pengguna untuk menentukan otorisasi pengguna terhadap sistem. Kemudian sesuai dengan otorisasi pengguna, akan tampil menu Utama sebagai berikut:

Gambar 8. Tampilan Antarmuka Menu Utama

Menu Utama berfungsi untuk menampilkan menu *Front Office*, menu *Food/Beverage*, dan menu *Marketing*. Menu utama juga menampilkan foto-foto yang berkenaan dengan Hotel Cipta beserta ulasannya. Kemudian memilih menu dengan melakukan klik mouse menu yang akan dipilih.

Pilihan menu yang sesuai dengan batasan lingkup masalah adalah menu *Front Office*. Setelah menu dipilih, akan tampil menu *Front Office* seperti yang terlihat pada Gambar 9 sebagai berikut:

Gambar 9. Tampilan Antarmuka Menu Front Office

Pada menu Front Office menampilkan menu-menu seperti menu *Booking*, menu *Check-in*, menu *Payment*, menu *Outlet*, menu *Statement*, menu *Schedule* dan menu *P.O.S*.

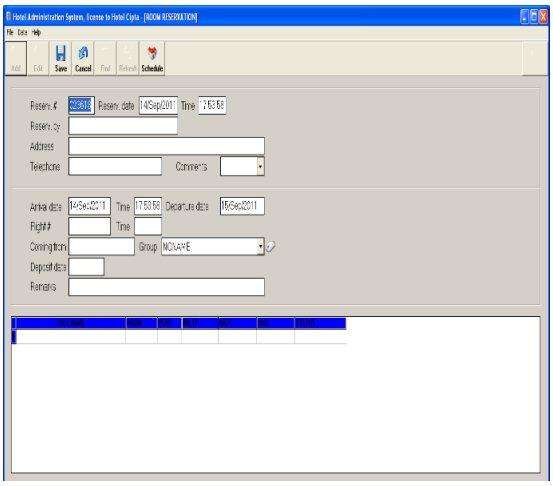
Sebelum memilih menu aplikasi yang ada pada menu Front Office, maka diperlukan untuk:

a. Melakukan *setup* seperti nomor kamar, jenis kamar, status kamar, jenis tujuan tamu
b. Melakukan *default* dalam persentase pajak, persentase pelayanan, batas akhir *waktu* check-out.
c. Melakukan *setting* jenis transaksi dan jenis pembayaran.
d. Laporan hasil transaksi pembayaran tamu

Masing-masing menu tersebut memiliki antarmuka dan akan dibahas secara lebih detail.

2. Antarmuka Menu *Booking*

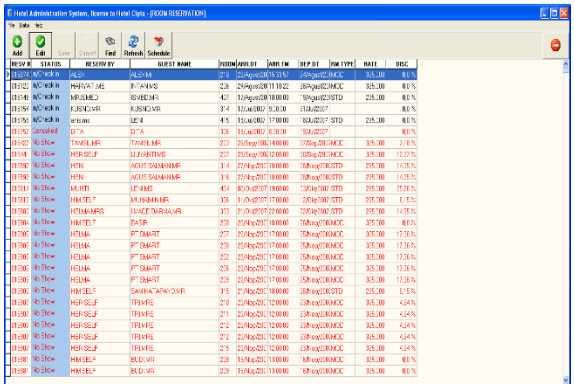
Antarmuka menu *Booking* atau *room reservation* ditunjukkan pada Gambar 10 yang menampilkan tombol *Add* untuk menginput data pemesanan kamar pertama kali atau menambah data, *Edit* untuk merubah data, *Save* untuk menyimpan data reservasi, *Cancel* untuk membatalkan data reservasi yang telah diinput, *Find* untuk menemukan data *record* pada kamar tertentu, *Refresh* untuk kembali menempatkan data pada posisi semula dan *Schedule* untuk melihat jadwal kamar yang ditempati para tamu hotel dalam bulan tertentu. Tombol *Save* dan *Cancel* akan aktif, saat tombol *Add* dan *Edit* mulai diaktifkan.



Gambar 10. Tampilan Antarmuka *Add/Edit* Menu *Booking*

Data reservasi tamu yang ditampilkan secara umum setelah tombol *Add/Edit* aktif dan data tamu diisi adalah nomor reservasi *(RESV. #)*, status tamu *(STATUS)*,

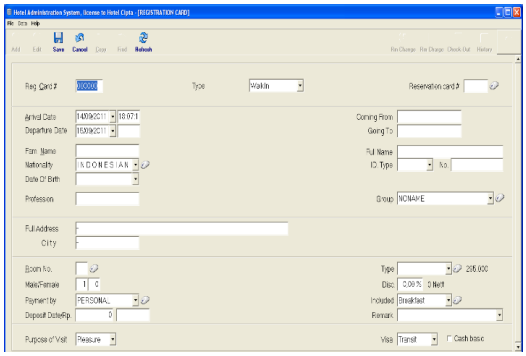
nama pemesan *(RESERV. BY)*, nama tamu *(GUEST NAME)*, kamar yang akan ditempati (*ROOM)*, tanggal kedatangan tamu *(ARR. DT)*, jam kedatangan tamu *(ARR. TM)*, tanggal kepergian tamu *(DEP. DT)*, tipe kamar *(RM. TYPE)*, rate kamar *(RATE)*, diskon kamar *(DISC)*. Antarmuka menu *Booking* akan ditunjukkan pada Gambar 11 sebagai berikut.



Gambar 11. Tampilan Antarmuka menu *Booking*

3. Antarmuka Menu *Check=in*

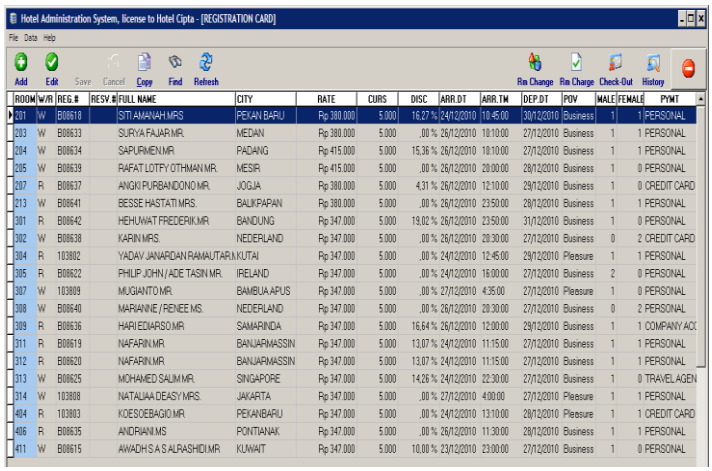
Antarmuka menu *Check-in* atau *room registration* menampilkan tombol *Add* untuk menginput data registrasi tamu pertama kali atau menambah data, *Edit* untuk merubah data, *Save* untuk menyimpan data registrasi tamu, *Cancel* untuk membatalkan data reservasi yang telah diinput, *Copy* untuk menyalin data *record* tertentu, *Find* untuk menemukan data *record* pada kamar tertentu, dan *Refresh* untuk kembali menempatkan data pada posisi semula. Tombol *Save* dan *Cancel* akan aktif, saat tombol *Add* dan *Edit* mulai diaktifkan. Antarmuka menu *Check-in* akan ditunjukkan pada Gambar 12 sebagai berikut.



Gambar 12. Tampilan Antarmuka *Add/Edit* Menu *Check-in*

Setelah *user* mengisi kolom yang telah disediakan pada *Add/Edit*, maka akan ditampilkan data registrasi tamu seperti nomor kamar (ROOM), jenis tamu dengan pilihan *walkin guest* atau *reservation guest(W/R)*, nomor registrasi *(REG. #)*, nomor

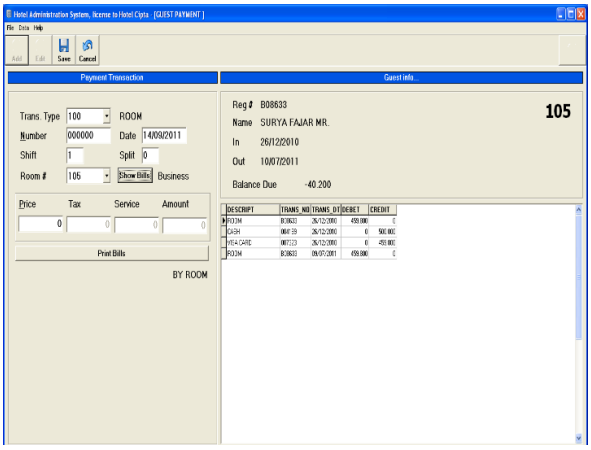
reservasi bila ada *(RESV. #)*, nama lengkap tamu (*FULL NAME)*, kota *(CITY)*, rate kamar *(RATE)*, kurs saat ini *(CURS)*, diskon *(DISC)*, tanggal kedatangan tamu *(ARR. DT)*, jam kedatangan *tamu (ARR. TM)*, tanggal kepergian tamu *(DEP. DT)*, tujuan kunjungan tamu *(Purpose of Visit)*, jumlah orang *(MALE/FEMALE)*, dibayarkan dengan *(PYMT)*, grup *(GROUP)*, dan nama user yang mendata tamu *(USERIN)*. Antarmuka akan ditunjukkan pada Gambar 13 berikut.



Gambar 13. Tampilan Antarmuka Menu *Check-in*

4. Antarmuka Menu *Payment*

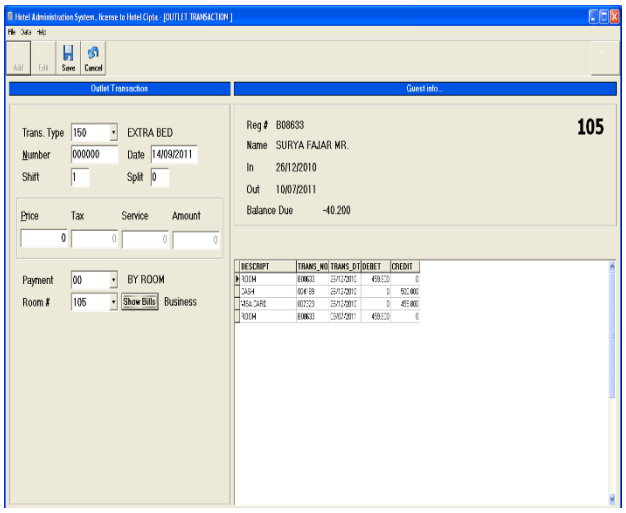
Antarmuka menu *Payment* menampilkan tombol *Add* untuk menginput transaksi pembayaran fasilitas hotel atau menambah data transaksi, *Edit* untuk merubah data transaksi, *Save* untuk menyimpan data transaksi, *Cancel* untuk membatalkan data transaksi yang telah diinput. Tombol *Save* dan *Cancel* akan aktif, saat tombol *Add* dan *Edit* mulai diaktifkan. Contoh tampilan akan ditunjukkan pada Gambar 14 berikut ini.



Gambar 14 Tampilan Antarmuka Menu *Payment*

5. Antarmuka Menu *Outlet*

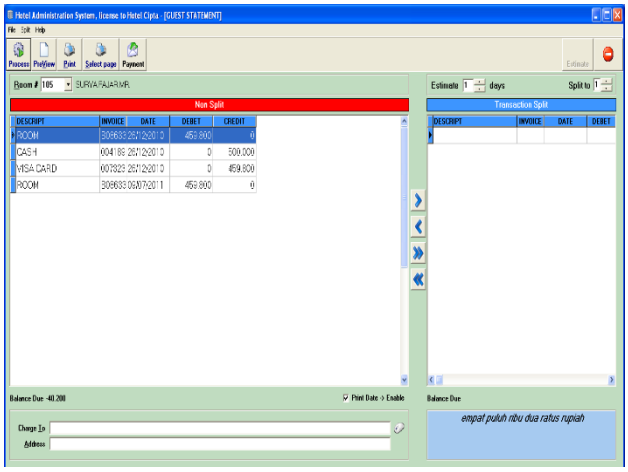
Antarmuka menu *Outlet* menampilkan tombol *Add* untuk menginput transaksi pembayaran fasilitas hotel atau menambah data transaksi, *Edit* untuk merubah data transaksi, *Save* untuk menyimpan data transaksi, *Cancel* untuk membatalkan data transaksi yang telah diinput. Tombol *Save* dan *Cancel* akan aktif, saat tombol *Add* dan *Edit* mulai diaktifkan. Contoh tampilan akan ditunjukkan pada Gambar 15 berikut ini.



Gambar 15. Tampilan Antarmuka Menu *Outlet*

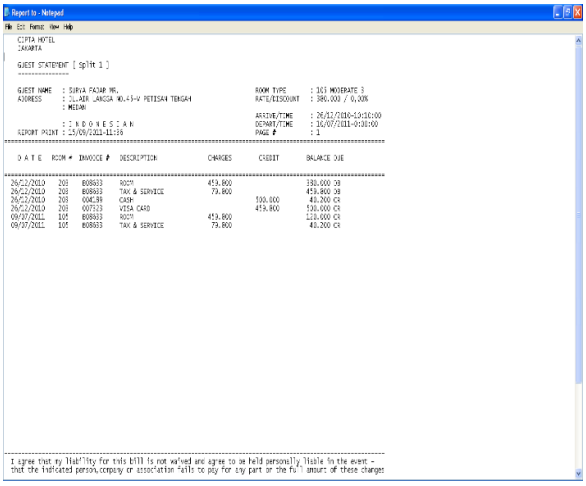
6. Antarmuka Menu *Statement*

Antarmuka menu Statement menampilkan laporan transaksi tamu hotel selama menginap. Dimulai dengan mengisi nomor kamar yang dipilih (Room #), kemudian otomatis menampilkan transaksi pembayaran fasilitas hotel. Pada menu Statement, user dapat memroses semua transaksi tamu pada kamar yang dipilih (Process) ditunjukkan pada Gambar berikut ini.



Gambar 16. Tampilan *Process* dari Antarmuka Menu *Statement*

Sedangkan menampilkan laporan di layar monitor sebelum dicetak di printer *(Preview)* dapat dilihat pada Gambar 17 berikut ini



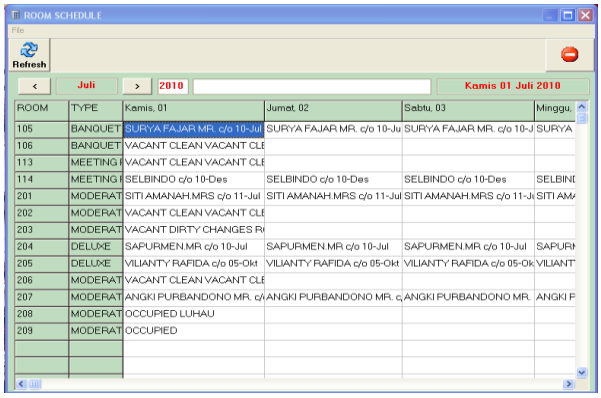
Gambar 17. Tampilan *Preview* dari Antarmuka Menu *Statement*

Untuk mencetak laporan semua transaksi *(Print)*, menyeleksi halaman yang akan dicetak *(Select Page)*, menampilkan tagihan pembayaran *(Payment)*, dan memisahkan laporan sesuai dengan keinginan tamu hotel jika diperlukan *(Split)*. Untuk mengakhiri akses *Statement*, *user* dapat menutup file dan melakukan *logout* agar dapat keluar dari aplikasi *Main Menu* tersebut.

7. **Antarmuka Menu *Schedule***

Desain antarmuka Menu *Schedule* dirancang untuk melihat kamar yang sedang dipakai dalam satu bulan sesuai dengan bulan dan tahun yang diinginkan. Tombol *Refresh* untuk menempatkan data kembali ke posisi semula. Untuk mengakhiri akses *Schedule*, *user* dapat menutup file dan melakukan *logout* agar dapat keluar dari aplikasi *Main Menu* tersebut.

Desain antarmuka menu *Schedule* akan ditunjukkan pada Gambar 18 berikut ini.

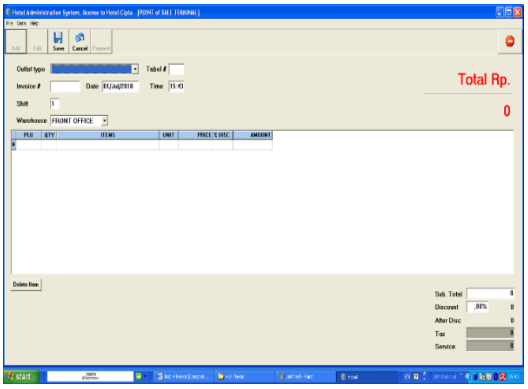


Gambar 18. Tampilan Antarmuka Menu *Schedule*

8. Antarmuka Menu *P.O.S*

Antarmuka menu *P.O.S* menampilkan tombol *Add* untuk menginput transaksi pembayaran fasilitas hotel atau menambah data transaksi, *Edit* untuk merubah data transaksi, *Save* untuk menyimpan data transaksi, *Cancel* untuk membatalkan data transaksi yang telah diinput. Tombol *Save* dan *Cancel* akan aktif, saat tombol *Add* dan *Edit* mulai diaktifkan.

Antarmuka menu *P.O.S* lebih terperinci dalam menginput data transaksi dibandingkan dengan Antarmuka menu *Outlet.* Menu *P.O.S* ini bersifat optional. Tampilan antarmuka menu *P.O.S* akan ditunjukkan pada Gambar 19 berikut ini.



Gambar 19. Tampilan Antarmuka Menu *P.O.S*

Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan salah satu proses rangkaian dari pengembangan perangkat lunak. Pengujian dilakukan dengan dua metode yaitu: dengan pengujian dengan metode *Black box* dan metode *White box*.

1. Pengujian *Black Box*

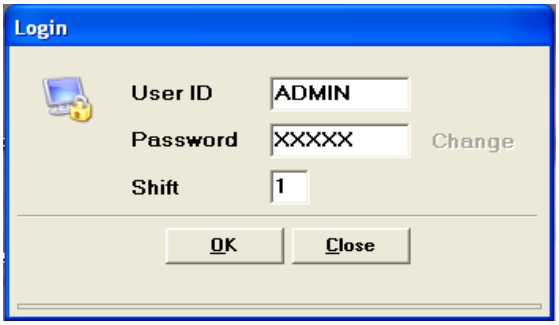
Menurut Sommerville (2003:p87), pengujian fungsional disebut juga pengujian kotak hitam '*black box'* merupakan pendekatan pengujian yang ujinya diturunkan dari spesifikasi program atau komponen. Sistem merupakan kotak hitam yang perilakunya hanya dapat ditentukan dengan mempelajari input dan output yang berkaitan. Penguji hanya berkepentingan dengan fungsionalitas dan bukan implementasi perangkat lunak.

Metode pengujiannya adalah dengan memberikan input terhadap sistem untuk menguji apakah setelah diberi input yang benar tersebut dapat menghasilkan output sesuai yang diharapkan, maka sistem dapat dikatakan bahwa sistem lolos dari pengujian *black box*.

Pengujian unit dimulai pada perincian antar muka:

a. Menu *login*

Saat modul login diinput dengan data *User ID* dan *Password* yang sesuai seperti pada gambar di bawah ini.

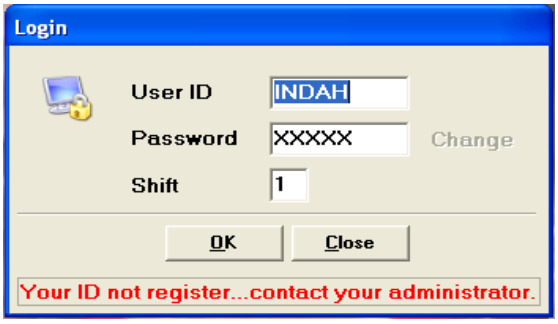


Gambar 20

Maka aplikasi akan menampilkan Menu Utama yang terlihat pada Gambar 2 Tampilan Menu Utama. Sebaliknya, maka akan tampil pesan kesalahan ”invalid password” sebagai berikut.

Gambar 21

Jika User ID tidak ada dalam database, maka muncul pesan kesalahan ”Your ID not register...contact your administrator” sebagai berikut.

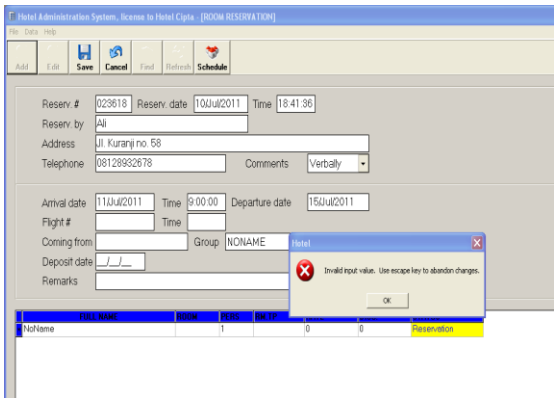


Gambar 22

b. Menu *Booking*

Saat Menu *Booking* diklik, muncul operasi Add dan Edit. Bila input salah, maka akan muncul pesan kesalahan ”Invalid input value. Use escape key to abandon

changes”seperti gambar berikut.

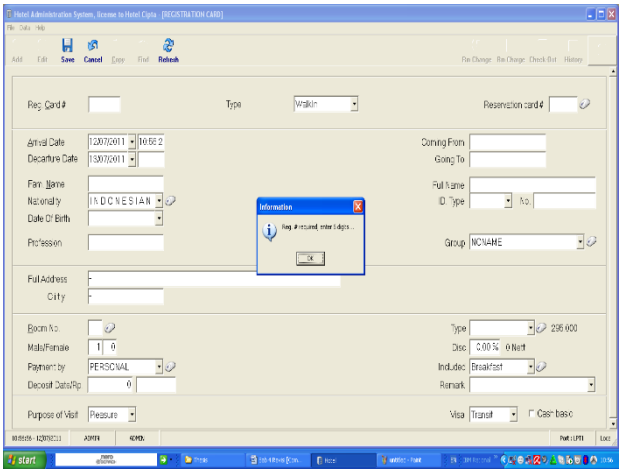


Gambar 23

Input data tamu Reservasi ada yang bersifat optional atau hanya tambahan saja dapat diisi input, bisa juga tidak. Saat pengujian input data yang memang seharusnya diisi malah dilewatkan dengan menekan tombol ”Enter” atau ”Tab”, maka data salah tersebut masih bisa disimpan dengan operasi *save*. Seharusnya ada muncul pesan kesalahan atau posisi kursor tidak akan bergerak ke komponen lain, jika data tidak diinput secara benar.

c. Menu *Check-in*

Saat input nomor registrasi tidak diisi dan dilewati, maka muncul peringatan “Reg # required, enter 6 digits” seperti gambar di bawah ini:



Gambar 24

Beberapa modul yang telah diuji ditampilkan pada tabel di bawah ini.

Tabel 1. Hasil Uji *Black Box*

Input (*Event*)	**Proses** (Form aplikasi)	**Output** (Hasil yang diharapkan)	**Hasil Uji** (sesuai/ tidak sesuai)
Input User ID, password yang telah terdaftar	Otentikasi	Menampilkan Menu Utama aplikasi sistem informasi hotel	Sesuai
Input User ID, Password yang belum terdaftar	Otentikasi	Menampilkan pesan kesalahan.	Sesuai
Klik Menu Front Office	*Menu Front Office*	Menampilkan menu-menu yang dibutuhkan di departemen *Front Office* yang sesuai dengan otorisasi dari *user* yang telah terdaftar	Sesuai
Klik menu *Booking*	*Room Reservation*	Menampilkan form pendataan tamu reservasi	Sesuai
Klik menu *Check-in*	*Room Registration*	Menampilkan form pendataan tamu registrasi	Sesuai
Klik menu *Payment*	*Payment Transaction*	Menampilkan form transaksi pembayaran dari tamu yang telah terdaftar pada input data tamu registrasi	Sesuai
Klik menu *Outlet*	*Outlet Transaction*	Menampilkan form transaksi outlet yang digunakan oleh tamu hotel	Sesuai
Klik menu *Statement*	*Guest Statement*	Menampilkan laporan dari semua transaksi yang digunakan tamu hotel, baik transaksi pembayaran dan transaksi outlet	Sesuai
Klik File, pilih *log out*	Proses keluar dari aplikasi	User keluar dari Aplikasi sistem	Sesuai

2. Pengujian *White Box*

Metode *white box* ini adalah pengujian struktural yang merupakan pendekatan terhadap pengujian yang diturunkan dari pengetahuan struktur dan implementasi perangkat lunak. Nama lainnya pengujian ‘kotak putih’, pengujian ‘kotak kaca’, atau pengujian ‘kotak jernih’ untuk membedakannya dari pengujian kotak hitam.

Pengujian ini biasanya diterapkan untuk unit program yang relatif kecil seperti subrutin atau operasi yang terkait dengan suatu objek. Penguji dapat menganalisis kode dan menggunakan pengetahuan mengenai struktur komponen untuk menurunkan data uji (Sommerville, 2003:91-93)

Pendekatan sistematis yang dipakai adalah partisi ekuivalensi. Partisi ekuivalensi input merupakan himpunan data dengan semua anggota harus diproses secara ekuivalen. Partisi ekuivalen output merupakan output program yang memiliki karakteristik yang sama sehingga dapat dianggap satu kelas yang berbeda.

Sesuai dengan teori yang disebutkan oleh Sommerville, maka penulis akan melakukan pengujian pada salah satu komponen pada nomor registrasi dari program. Contoh kasus pengujian *white box* sebagai berikut:

a. Dilakukan pengujian pada komponen dari program untuk input nomor registrasi tamu. DBEdit1.text merujuk pada nomor registrasi tamu.

b.

```
procedure
TFRegistration.DBEdit1Exit(Sender:
TObject);
begin
if (DBEdit1.Text='') or
(Length(DBEdit1.Text)<6) or
(DBEdit1.Text='000000') then
begin
MessageDlg('Reg. #
required, enter 6 digits...',
mtInformation,[mbOK],0);
DBEdit1.SetFocus;
end;
end;
```

Gambar 25. Spesifikasi Komponen Program

Contoh data input dan output partisi ekuivalensi untuk komponen DBEdit1.Text sebagai berikut.

Tabel 2. Data Input dan Output dari DBEdit1.Text

Input	Proses	Output
'' artinya nilai DBEdit1.Text adalah blank	Menampilkan pesan dialog "Reg. # required, enter 6 digits…"	Output **tidak sesuai**, yang diharapkan
A99999 artinya (jumlah panjang DBEdit1.Text=6)	Melangkah ke proses berikutnya	Output **sesuai** dengan yang diharapkan
7888 artinya (jumlah panjang DBEdit1.Text<6)	Menampilkan pesan dialog "Reg. # required, enter 6 digits…"	Output tidak sesuai yang diharapkan
000000 (DBExit1.Text='000000')	Menampilkan pesan dialog "Reg. # required, enter 6 digits…"	Output tidak sesuai yang diharapkan

Dari hasil pengujian *white box*, komponen program tersebut di atas sesuai dengan yang diharapkan. Jika komponen DBEdit1 yang menunjukkan nomor registrasi tidak ada input atau jumlah panjang input lebih kecil dari enam atau nilai input '000000', maka kursor tidak akan berpindah ke komponen objek yang lain dan akan menampilkan pesan kesalahan "Reg. # required, enter 6 digits…". Ini berarti sub rutin lolos dari pengujian *white box*.

Hasil Pengujian

Dari pengujian metode black box dengan melihat fungsionalitas sistem dan bukan implementasi aplikasi sistem diperoleh hasil bahwa:

a. Menu login sudah sesuai dengan yang diharapkan.
b. Masih perlu perbaikan pada input data tamu reservasi. Saat pengujian input data tamu reservasi, input data yang memang seharusnya diisi bisa lolos ke komponen lain, bahkan data masih bisa disimpan dengan operasi *save*. Seharusnya muncul pesan kesalahan atau posisi kursor tidak akan bergerak ke komponen lain.
c. Input data tamu registrasi dan menu-menu lainnya sudah lolos pengujian white box artinya sudah sesuai dengan yang diharapkan.

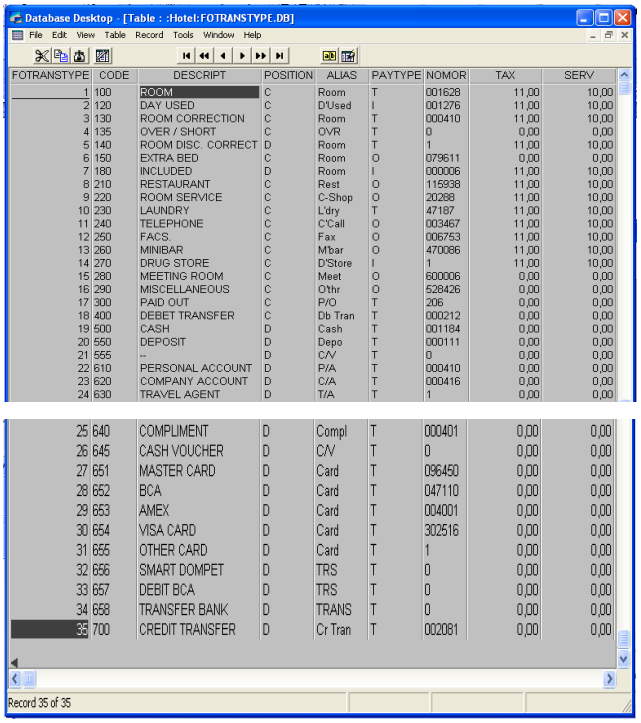
Pengujian metode *white box* diperoleh hasil adalah bahwa sub rutin untuk input nomor registrasi yang diuji lolos atau berhasil dari pengujian white box adalah input yang bertipe karakter dengan panjang karakter 6 digit.

Hasil Penerapan Model *Incremental Delivery*

Dalam pengembangan aplikasi sistem informasi Hotel Cipta khususnya di departemen *Front Office*, menggunakan model *incremental delivery* dan membagi kebutuhan sistem menjadi beberapa inkremen atau bagian sehingga menjadi suatu sistem informasi hotel yang lengkap.

Pembagian terhadap empat inkremen pada model *incremental delivery* sesuai metode Pressman terhadap sistem informasi hotel khususnya sistem transaksi hotel pada departemen *Front Office*m yaitu:

a. Kebutuhan *user* yang menjadi prioritas tertinggi di awal inkremen adalah membuat menu *Booking* dan menu *Check-in*.
b. Setelah inkremen pertama selesai, maka akan menjadi dasar pengembangan untuk inkremen kedua ini, yaitu membuat menu menu *Outlet*, menu *Payment*, menu *Statement*.
c. Setelah inkremen kedua selesai, masuk pada inkremen ketiga yaitu membuat menu *Schedule* dan menu *P.O.S*. Menu *Schedule* dibuat untuk melihat status kamar dalam jadwal satu bulan seperti terisi atau tidak oleh tamu, kamar dalam kondisi kosong dan masih kotor, masih dalam pemesanan, kamar masih dalam kondisi rusak dan lain-lain. Sedangkan menu *P.O.S* bersifat optional maksudnya bisa digunakan atau tidak karena sebagai alternatif lain dari menu *Outlet.*
d. Inkremen terakhir yaitu membuat Menu *Login* dan *setup,default* dan *setting*. Melakukan *setup* seperti nomor kamar, jenis kamar, status kamar, jenis tujuan tamu, sedangkan *default* dalam persentase pajak, persentase pelayanan, batas akhir waktu *check-out. Setting* jenis transaksi dan jenis pembayaran.



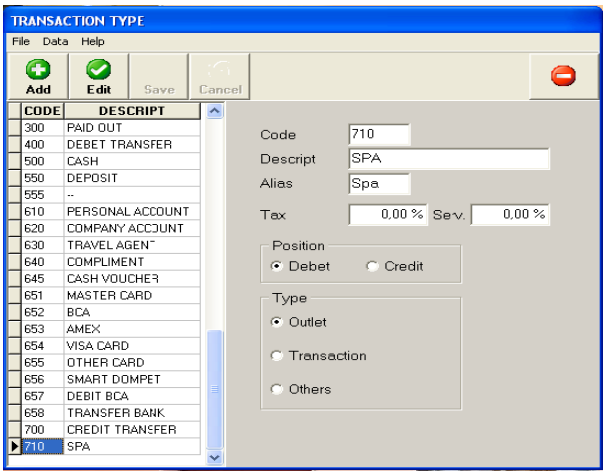
FOTRANSTYPE	CODE	DESCRIPT	POSITION	ALIAS	PAYTYPE	NOMOR	TAX	SERV
1	100	ROOM	C	Room	T	001628	11,00	10,00
2	120	DAY USED	C	D'Used	I	001276	11,00	10,00
3	130	ROOM CORRECTION	C	Room	T	000410	11,00	10,00
4	135	OVER / SHORT	C	OVR	T	0	0,00	0,00
5	140	ROOM DISC. CORRECT	D	Room	T	1	11,00	10,00
6	150	EXTRA BED	C	Room	O	079611	0,00	0,00
7	180	INCLUDED	D	Room	I	000006	11,00	10,00
8	210	RESTAURANT	C	Rest	O	115938	11,00	10,00
9	220	ROOM SERVICE	C	C-Shop	O	20268	11,00	10,00
10	230	LAUNDRY	C	L'dry	T	47187	11,00	10,00
11	240	TELEPHONE	C	C'Call	O	003467	11,00	10,00
12	250	FACS.	C	Fax	O	006753	11,00	10,00
13	260	MINIBAR	C	M'bar	O	470086	11,00	10,00
14	270	DRUG STORE	C	D'Store	I	1	11,00	10,00
15	280	MEETING ROOM	C	Meet	O	600006	0,00	0,00
16	290	MISCELLANEOUS	C	O'thr	O	528426	0,00	0,00
17	300	PAID OUT	C	P/O	T	206	0,00	0,00
18	400	DEBET TRANSFER	C	Db Tran	T	000212	0,00	0,00
19	500	CASH	D	Cash	T	001184	0,00	0,00
20	550	DEPOSIT	D	Depo	T	000111	0,00	0,00
21	555	--	D	C/V	T	0	0,00	0,00
22	610	PERSONAL ACCOUNT	D	P/A	T	000410	0,00	0,00
23	620	COMPANY ACCOUNT	D	C/A	T	000416	0,00	0,00
24	630	TRAVEL AGENT	D	T/A	T	1	0,00	0,00
25	640	COMPLIMENT	D	Compl	T	000401	0,00	0,00
26	645	CASH VOUCHER	D	C/V	T	0	0,00	0,00
27	651	MASTER CARD	D	Card	T	096450	0,00	0,00
28	652	BCA	D	Card	T	047110	0,00	0,00
29	653	AMEX	D	Card	T	004001	0,00	0,00
30	654	VISA CARD	D	Card	T	302516	0,00	0,00
31	655	OTHER CARD	D	Card	T	1	0,00	0,00
32	656	SMART DOMPET	D	TRS	T	0	0,00	0,00
33	657	DEBIT BCA	D	TRS	T	0	0,00	0,00
34	658	TRANSFER BANK	D	TRANS	T	0	0,00	0,00
35	700	CREDIT TRANSFER	D	Cr Tran	T	002081	0,00	0,00

Gambar 26. Tampilan Awal Tabel FOTransType.db

Dengan menggunakan model Incremental, hal ini dapat diatasi dengan menambahkan *field* di bawah dari urutan *record* terakhir ke-35. Misalnya membuat kode transaksi masing-masing secara berurut 710 menempati record ke 36. Desain tabel untuk file FOTransType.db sebagai berikut.

Tabel 3. Desain Tabel File FOTransType.db

Record	*Code*	*Descript*
1	100	ROOM
2	120	DAY USED
3	130	ROOM CORRECTION
4	135	OVER / SHORT
5	140	ROOM DISC.CORRECT
6	150	EXTRA BED
7	180	INCLUDED
8	210	RESTAURANT
9	220	ROOM SERVICE
	...	....
34	658	TRANSFER BANK
35	700	CREDIT TRANSFER
36	710	SPA (record ditambah)



Gambar 27. Tampilan Akhir Program Eksekusi dengan Tambahan Outlet Spa

5. KESIMPULAN DAN SARAN

Dari hasil penelitian yang dilakukan mulai dari tahap awal hingga proses pengujian, maka penerapan model *incremental delivery* pada pengembangan aplikasi sistem informasi Hotel Cipta Jakarta dapat disimpulkan bahwa Modul dibagi dalam empat bagian/inkremen, yaitu (1) inkremen pertama, membuat menu *Booking*, dan menu *Check-in*; (2) inkremen kedua, membuat menu Outlet, menu *Payment*, menu *Statement;* (3) inkremen ketiga, membuat menu *Schedule* dan menu *P.O.S.*, dan; (4) inkremen keempat, membuat menu *Login* dan melakukan *setup* kamar, serta *setting* paramater jenis transaksi dan jenis pembayaran. Inkremen pertama, persyaratan didefinisikan dengan rinci dan menjadi prioritas, kemudian diserahkan ke analis/pembuat modul menu pertama sampai sesuai dengan yang diinginkan pemilik/manajemen hotel. Selanjutnya pemilik/manajemen hotel mencoba prototipe. Jika sesuai, maka mereka memberikan penjelasan persyaratan inkremen kedua diserahkan ke analis/pembuat modul prototipe, dan analis mengirimkannya kembali ke pemilik/manajemen hotel. Bila inkremen terakhir selesai. Inkremen-inkremen yang sudah ada itu diintegrasikan sehingga fungsionalitas sistem bertambah baik dan implementasi software diterapkan.

Berdasarkan hasil penelitian, ada beberapa hal yang disarankan penulis, yaitu Input data tamu reservasi tidak hanya melalui telepon, fax, email atau datang langsung, tetapi dapat dilakukan melalui web, dan metode penerapan model *incremental delivery* pada pengembangan aplikasi sistem informasi Hotel Cipta ini, dapat berguna bagi

peneliti lain untuk memperbaiki dan mengembangkannya di kemudian hari, misalnya berbasis web.

DAFTAR REFERENSI

Bagyono (2020), Teori dan Praktik Hotel Front Office, Alfabeta.

Ian Sommerville (2003), Software Engineering (Rekayasa Perangkat Lunak) Edisi 6 Jilid 1 versi Bahasa Indonesia. Alih Bahasa Dra. Yuhilza Hanum, M.Eng. Penyunting H.M. Wibi Hardani, S.T. Jakarta: Penerbit Erlangga, Judul Asli: Software Engineering/Sixth Edition, by Ian Sommerville.

Ian Sommerville (2003), Software Engineering (Rekayasa Perangkat Lunak) Edisi 6 Jilid 2 versi Bahasa Indonesia. Alih Bahasa Dra. Yuhilza Hanum, M.Eng. Penyunting H.M. Wibi Hardani, S.T. Jakarta: Penerbit Erlangga, Judul Asli: Software Engineering/Sixth Edition, by Ian Sommerville.

Ibrahim, I. M. (2020). Iterative and Incremental Development Analysis Study of Vocational Career Information Systems. International Journal of Software Engineering & Applications, 11(5), 13–24. https://doi.org/10.5121/ijsea.2020.11502.

Kendall, Kenneth E. & Kendall, Julie E. (2016). Analisis dan Perancangan Sistem Edisi Kelima Jilid 1 versi Bahasa Indonesia. Alih Bahasa Tim Thamir Abdul Hafedh Al-Hamdany, B.Sc., M.Sc., Penyunting: M. Azahari. Jakarta: Penerbit PT Indeks Kelompok Gramedia, Judul Asli: Systems Analysis and Design/Fifth Edition, by Kenneth E. Kendall dan Julie E. Kendall.

Parastoo Mohagheghi (2004), The Impact of Software Reuse and Incremental Development on Quality of Large Systems, Doctoral Thesis, Norwegian University of Science and Technology.

Pressman, Roger S. (2023), Alih Bahasa: Rekayasa Perangkat Lunak Pendekatan Praktis Edisi 7 Buku 2, Penerbit Andi, Yogyakarta.

Rachman, A., Andreansyah, & Rahmi. (2020). Implementation of Incremental Models on Development of Web-Based Loan Cooperative Applications. International Journal of Education, Science, Technology, and Engineering, 3(1), 26–34.

Sari, Nurullita, Isnurani, & Sastro, G. (2023), Sistem Informasi Manajemen, Unpam Press.

Sari, Shinta Wulan (2021), Akuntansi Perhotelan Teori dan Aplikasi, Polimedia: Jakarta.