

# Aplikasi *Monitoring Stock* Barang dengan Fitur Notifikasi Berbasis *Web*

Ivana Celesta, Sistem Informasi, Universitas Katolik Musi Charitas dan Maria Bellanier Ismiati, Sistem Informasi, Universitas Katolik Musi Charitas

**Abstract**— Hotel Santika Radial Palembang merupakan hotel yang terletak di pusat kota Palembang yakni di Jalan Radial. Hotel Santika Radial Palembang sudah beroperasi selama hampir 5 tahun. Pada Hotel Santika Radial Palembang memiliki beberapa bagian unit kerja, salah satunya bagian Accounting. Pada bagian unit kerja Accounting ini memiliki beberapa sub bagian lagi. Dari beberapa sub bagian ini, fokus penelitian yaitu pada sub bagian store (stock control). Pada sub bagian ini ditemukan permasalahan bahwa dalam proses perhitungan stok barang masih dilakukan secara manual. Maka dari itu dibuatlah aplikasi monitoring stock barang dengan fitur notifikasi berbasis web untuk membantu admin dapat lebih cepat dan mudah dalam proses perhitungan stok barang dan pencarian data sebelumnya. Aplikasi ini dibangun dengan menggunakan metode Rapid Application Development (RAD) dan analisis permasalahan menggunakan PIECES. Aplikasi monitoring stock barang dengan fitur notifikasi berbasis web ini diharapkan dapat membantu pihak hotel untuk me-monitor barang dan stok barang.

**Index Terms**— Aplikasi monitoring stock barang, Hotel Santika Radial Palembang, mySQL, PIECES, PHP, RAD method.

## I. PENDAHULUAN

Hotel Hotel & Resor Santika Indonesia, di bawah manajemen PT.Grahawita Santika didirikan pada tahun 1981[8]. Salah satu cabang Hotel Santika yang berada di kota Palembang yaitu Hotel Santika Radial yang terletak di pusat kota Palembang yakni di Jl. HM Dhani Effendi (Jl. Radial). Hotel Santika Radial Palembang sudah mulai beroperasi selama hampir 5 tahun, sejak 19 Agustus 2016. Hotel ini memiliki 16 lantai dengan berbagai fasilitas yang dimiliki seperti 149 unit kamar, lounge, ruang meeting, laksa restaurant, swimming pool, mini gym, dan Wi-Fi. Visi Hotel Santika Radial Palembang yaitu Menjadi Jaringan Hotel Pilihan Utama yang tersebar di Indonesia dan terbesar di Asia Tenggara”. Misi Hotel Santika Radial Palembang yaitu “Menciptakan Nilai Lebih bagi Stakeholders dengan Menyajikan Produk Bermutu disertai Pelayanan Professional yang Ramah dalam mewujudkan ‘Sentuhan Indonesia’ Sebagai Citra Santika”. Struktur organisasi pada Hotel Santika Radial terdiri dari *general manager, manager, assistant manager, room division,*

*F&B division, accounting, engineering, marketing, HRD, supervisor, captain, leader, staff/admin, daily worker, dan OJT (On The Job Training/Magang).*

## II. KEGIATAN KERJA PRAKTEK

Pada Hotel Santika Radial Palembang memiliki 9 bagian unit kerja dimulai dari *General Manager, Manager, Assistant Manager, Room Division, F&B (Food and Beverage) Division, Accounting Division, Engineering Division, Marketing Division, dan HRD Division*. Bagian-bagian ini saling berkaitan dalam menjalankan operasional bisnis pada Hotel Santika Radial Palembang.

Bagian unit kerja yang akan diteliti adalah bagian Accounting. Pada bagian unit kerja ini terdiri dari beberapa sub bagian yaitu *book keeper, cost control, account receiveable, account payable, general cashier, purchasing, receiving, income audit, store (stock control), dan assistant accounting manager*.

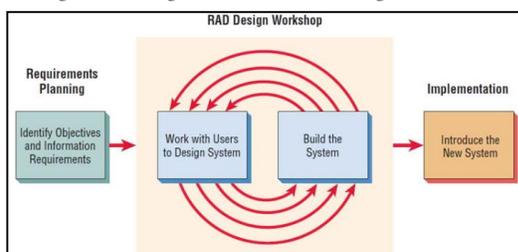
Dari beberapa sub bagian ini, fokus penelitian pada sub bagian *store (stock control)*. Pada sub bagian ini, bertugas untuk menyediakan stok barang yang dibutuhkan departemen lain, serta melakukan pengecekan atas stok barang yang ada di Hotel Santika Radial Palembang, dan memproses permintaan barang keluar.

## III. METODOLOGI PENGEMBANGAN SISTEM

Metodologi merupakan kesatuan metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan, dan postulat-postulat yang dapat digunakan oleh suatu ilmu pengetahuan, seni, atau disiplin lainnya[3]. Metode adalah suatu proses atau prosedur yang sistematis berdasarkan prinsip dan teknik ilmiah yang dipakai oleh disiplin (ilmu) untuk mencapai suatu tujuan[5].

Metodologi pengembangan sistem yang akan digunakan adalah *Rapid Application Development (RAD)* atau *rapid prototyping*. Model ini adalah model proses pembangunan perangkat lunak yang termasuk dalam teknik bertingkat (*incremental*). Model RAD menekankan pada siklus pembangunan sistem informasi yang pendek, singkat, dan tepat [1]. Batasan yang penting untuk model ini adalah waktu yang singkat. RAD menggunakan metode berulang (*iterative*) dalam pengembangan sistem, dimana model kerja (*working model*) sistem dikonstruksikan pada tahapan awal pengembangan dengan tujuan penetapan kebutuhan (*requirements*) user dan selanjutnya akan disingkirkan.

RAD adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus



perkembangan yang sangat pendek. Model ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier di mana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen[6]. Model RAD dapat dilihat pada gambar di bawah ini.

**.Gambar 1 Rapid Application Development (RAD) Model**  
**Sumber : Kendall, 2008**

Menurut Kendall, terdapat tiga fase besar dalam tahap RAD yang melibatkan pengguna dan analis dalam penilaian, desain, dan implementasi, yaitu sebagai berikut[4]:

a. *Requirements Planning*

Pada fase ini, pengguna dan analis melakukan pertemuan untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi pada fase ini yaitu menyelesaikan masalah-masalah perusahaan.

Pada penelitian ini, fase *Requirements Planning*, dilakukan dengan mengadakan pertemuan dengan pembimbing lapangan di Hotel Santika Radial Palembang. Untuk mengidentifikasi tujuan-tujuan aplikasi dan mengidentifikasi syarat-syarat informasinya, dilakukan proses diskusi, wawancara, maupun tanya jawab langsung dengan pembimbing lapangan dari Hotel Santika Radial Palembang. Didapatkan hasil yaitu Hotel Santika Radial Palembang membutuhkan sebuah aplikasi untuk membantu kegiatan stok barang di *Store*.

- Identifikasi Permasalahan

Permasalahan yang ada pada sistem lama menjadi suatu kendala dalam mencapai tujuan dari sebuah sistem, sehingga diperlukan pemecahan dari masalah ini agar sistem dapat berjalan sesuai dengan keinginan pengguna.

Untuk mengidentifikasi masalah yang ada, maka perlu dilakukan analisis terhadap kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan. Aspek ini dikenal dengan analisis *PIECES* (*Performance, Information, Economy, Control, Efficiency, dan Services*). Setelah melakukan analisis *PIECES*, akan melakukan analisis sebab akibat berdasarkan hasil dari analisis *PIECES*. Hasil analisis ini diharapkan menjadi landasan dalam membuat aplikasi yang dapat membantu staff store keeper dalam melakukan monitoring stock barang di Hotel Santika Radial Palembang.

- Analisis *PIECES*

- Analisis Kinerja (*Performance*)

Analisis kinerja (*performance*) merupakan peningkatan terhadap kinerja (hasil kerja) sistem yang baru sehingga menjadi lebih efektif. Kinerja dapat diukur dari *throughput* (*Tp*) dan *response time* (*Tr*). *Throughput* (*Tp*) ialah jumlah dari pekerjaan yang dapat dilakukan suatu sistem tertentu, sedangkan *response time* (*Tr*) adalah rata-rata waktu yang tertunda diantara dua pekerjaan ditambah dengan *response time* untuk menangani pekerjaan

tersebut.

Setelah dilakukan analisis terhadap sistem lama yang ada, terdapat beberapa kelemahan pada segi kinerja, yaitu harus mencatat semua secara manual, seperti stok barang yang dicatat pada *bin card*, serta transaksi stok barang masuk dan stok barang keluar yang dicatat pada *log book store*. Kegiatan ini memakan waktu, menggunakan banyak kertas dalam prosesnya, dan meningkatkan risiko kehilangan data.

Pada sistem lama juga tidak ada pencatatan secara khusus mengenai tanggal *expired*/tanggal kadaluwarsa barang, sehingga meningkatkan kelalaian dalam menggunakan barang yang sudah *expired* atau kadaluwarsa.

- Analisis Informasi (*Information*)

Merupakan peningkatan terhadap kualitas informasi yang disajikan. Setelah melakukan analisis terhadap sistem lama, terdapat kelemahan pada segi informasi, dimana belum ada informasi yang jelas terkait jumlah stok barang, transaksi barang masuk dan barang keluar, serta adanya kemungkinan kesalahan dalam penginputan jumlah barang masuk dan barang keluar (*human error*).

- Analisis Ekonomi (*Economy*)

Merupakan peningkatan terhadap manfaat-manfaat atau keuntungan-keuntungan maupun penurunan-penurunan biaya yang terjadi. Setelah melakukan analisis terhadap sistem lama, masih terdapat kelemahan dari segi ekonomi, yaitu sistem lama membutuhkan banyak kertas dan pena untuk mencatat barang masuk dan barang keluar pada *bin card*, serta membutuhkan buku dalam pencatatan transaksi barang masuk dan barang keluar pada *log book store*. Hal ini merupakan pemborosan dalam bidang keuangan.

- Analisis Pengendalian (*Control*)

Merupakan peningkatan terhadap pengendalian untuk mendeteksi dan memperbaiki kesalahan-kesalahan serta kecurangan-kecurangan yang terjadi. Setelah melakukan analisis terhadap sistem lama, masih terdapat kelemahan dari segi pengendalian, yaitu pada sistem lama terdapat kemungkinan kehilangan *bin card* dan buku *log book store*, dan pencatatan ini masih dilakukan secara manual, sehingga jika data tersebut sedang dibutuhkan, maka akan menjadi persoalan bagi *staff store keeper*, karena harus mengumpulkan data-data lama dan

merekap ulang data-data yang ada secara manual.

▪ Analisis Efisiensi (*Efficiency*)

Merupakan peningkatan terhadap efisiensi operasi. Efisiensi berbeda dengan ekonomis. Bila ekonomis berhubungan dengan jumlah sumber daya yang digunakan, efisiensi berhubungan dengan bagaimana sumber data digunakan dengan pemborosan paling minimum. Efisiensi dapat diukur dari output dibagi dengan inputnya.

Setelah melakukan analisis terhadap sistem lama, masih terdapat kelemahan pada segi efisiensi. Pada sistem yang lama, menyimpan seluruh data pada laci dan rak sehingga banyak memakan tempat, dan menjadi tidak efisien.

▪ Analisis Pelayanan (*Service*)

Merupakan peningkatan terhadap pelayanan yang diberikan oleh sistem. Setelah melakukan analisis terhadap sistem lama, masih terdapat kelemahan di bagian pelayanan. Pada sistem lama pelayanan terhadap bagian *Accounting* masih kurang, karena informasi masih lambat dan membutuhkan waktu yang cukup lama untuk penyampaian laporan mengenai stok barang.

○ Analisis Sebab Akibat

Berdasarkan hasil dari metode analisis *PIECES* pada poin diatas yang membahas mengenai permasalahan yang terdapat pada Hotel Santika Radial Palembang bagian *Store*, maka akan dijelaskan mengenai akibat yang dapat terjadi dari permasalahan yang ada pada **Tabel 1**.

**Tabel 1 Analisis Sebab Akibat**

No	Kriteria	Permasalahan	Sebab	Akibat
1.	Kinerja / <i>Performance</i>	Mencatat semua barang dan transaksi stok barang secara manual. Tidak ada pencatatan mengenai tanggal <i>expired</i> barang.	Pencatatan masih secara manual oleh <i>staff store keeper</i> .	Memakan waktu dan penggunaan kertas yang banyak, serta meningkatkan risiko kehilangan data.
2.	Informasi / <i>Information</i>	Tidak ada informasi terkait stok barang, laporan barang masuk dan barang keluar, dan <i>human error</i> .	Tidak ada pencatatan jelas mengenai stok keseluruhan barang di dalam satu buku atau laporan.	Kurangnya informasi jelas mengenai barang, stok barang, barang masuk dan barang keluar.
3.	Ekonomi / <i>Economy</i>	Masih menggunakan kertas, buku, dan pena.	Pencatatan menggunakan buku, kertas, dan pena tanpa menggunakan bantuan sistem.	Pemborosan biaya pembelian alat tulis kantor (ATK) dan tempat penyimpanan data.
4.	Pengendalian / <i>Control</i>	Belum ada sistem untuk menyimpan data barang dan <i>monitoring</i> stok barang.	Kehilangan data barang dan stok barang.	Data hilang, tidak tercatat, tidak tersimpan dengan baik.
5.	Efisiensi / <i>Efficiency</i>	Memakan banyak tempat untuk penyimpanan barang.	Penyimpanan data di rak-rak barang dan di laci.	Diperlukan selalu tempat untuk menyimpan data yang sudah terisi penuh ( <i>bin card, log book store</i> )
6.	Pelayanan / <i>Service</i>	Pelayanan kepada bagian <i>Accounting</i> masih kurang.	Pelaporan masih manual.	Waktu yang cukup lama dalam penyampaian laporan stok barang.

b. *RAD Design Workshop*

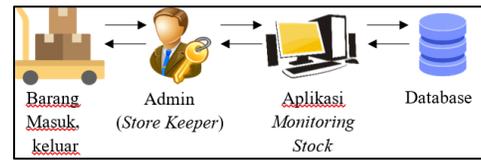
Fase ini merupakan fase untuk merancang dan memperbaiki, fase ini bisa digambarkan sebagai *workshop*. Analis dan *programmer* dapat bekerjasama dalam membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran

aplikasi yang akan dikembangkan. Selama *workshop* desain RAD berlangsung, pengguna akan memberikan respon terhadap prototipe yang sebenarnya dan analis akan memperbaiki modul-modul yang dirancang berdasarkan respon pengguna.

Pada fase ini, dilakukan diskusi lagi terkait aplikasi yang sudah dibuat dengan meminta tanggapan dari pembimbing lapangan di Hotel Santika Radial Palembang. Hal ini bertujuan agar aplikasi yang dirancang dapat sesuai dengan apa yang diminta oleh pengguna (diwakilkan oleh pembimbing lapangan).

• Perancangan Arsitektur Sistem

Arsitektur dari sistem yang akan dibuat secara keseluruhan pada sub-bagian *Store* di Hotel Santika Radial Palembang dapat dilihat pada **Gambar 3.2** di bawah ini.



**Gambar 2 Arsitektur Sistem yang Diusulkan**



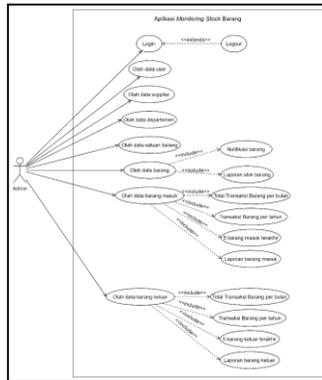
Gambar diatas menjelaskan bahwa barang masuk dan barang keluar akan di cek oleh bagian *staff store keeper* kemudian *staff store keeper* akan meng-input data barang masuk dan barang keluar, kemudian data barang masuk dan barang keluar ini akan disimpan ke dalam *database*. Setelah itu, *database* akan menampilkan *form* jumlah stok barang di Aplikasi *Monitoring Stock* barang di Hotel Santika Radial Palembang. Aplikasi *Monitoring Stock* akan menghitung jumlah barang dan stok barang di Hotel Santika Radial Palembang. Kemudian aplikasi akan menampilkan informasi data barang masuk dan barang keluar serta stok barang dari Aplikasi *Monitoring Stock* dan informasi ini dapat diunduh ke komputer.

• Pemodelan Proses Sistem

UML (*Unified Modeling Language*) merupakan Bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak [2].

a) *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use Case Diagram yang diusulkan dapat dilihat pada Gambar 3.



Gambar 3 Use Case Diagram yang Diusulkan

Berdasarkan Gambar 3.3 di atas, hanya ada 1 aktor dalam use case diagram yaitu admin/staff store keeper yang akan menggunakan aplikasi.

b) Activity Diagram

Activity Diagram menggambarkan workflow (alur kerja) atau aktivitas dari sebuah sistem atau proses bisnis.



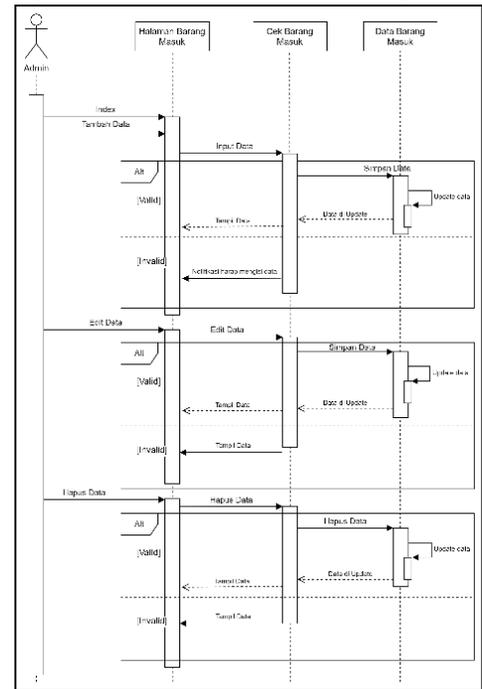
Gambar 4 Activity Diagram Barang Masuk

Gambar 5 Activity Diagram Barang Keluar

Berdasarkan Gambar 4 dan Gambar 5 diatas, dapat dilihat bahwa activity diagram untuk barang masuk dan barang keluar memiliki alur yang mirip. Dimulai dari memilih menu transaksi, kemudian memilih sub-menu barang masuk atau barang keluar. Kemudian aplikasi akan menampilkan data, lalu memilih aksi tambah data barang, mengisi form yang ditampilkan, dan data akan di-update.

c) Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.



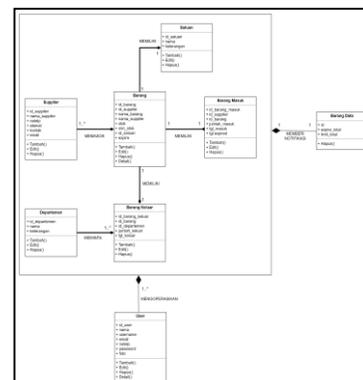
Gambar 6 Sequence Diagram Halaman Barang Masuk

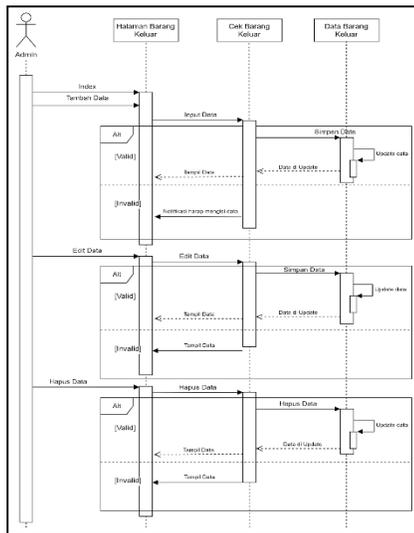
Gambar 7 Sequence Diagram Halaman Barang Keluar

Berdasarkan Gambar 6 dan 7 dapat dilihat bahwa sequence diagramnya mirip antara barang masuk dan barang keluar. Halaman yang akan ditampilkan pertama kali adalah index yang berisi data barang masuk atau keluar, admin dapat mengolah barang masuk keluar dengan melakukan tambah data barang, edit data barang, dan hapus data barang.

• Pemodelan Data Sistem

Pemodelan data sistem pada pembangunan aplikasi ini menggunakan UML Class Diagram. Class diagram merupakan hubungan antar kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Class diagram meliputi: kelas (class), relasi Associations, Generalization, dan





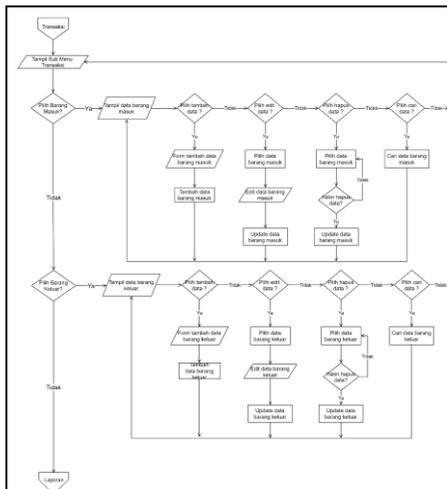
Aggregation, atribut (attributes), operasi (operation/method), dan visibility, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan multiplicity atau cardinality. Class diagram yang diusulkan dapat dilihat pada gambar di bawah ini.

**Gambar 8 Class Diagram yang Diusulkan**

Berdasarkan Gambar 8 class diagram aplikasi yang diusulkan memiliki delapan entitas yaitu supplier, departemen, barang, satuan, barang data, barang masuk, barang keluar, dan user. Class diagram memiliki satu level user yaitu admin/staff store keeper yang bertugas untuk mengoperasikan aplikasi yang diusulkan. Pada class diagram ini satu barang memiliki satu satuan. Satu supplier memasok banyak barang, dan satu barang memiliki satu barang masuk. Departemen meminta banyak barang keluar, satu barang memiliki satu barang keluar. Sedangkan pada barang data digunakan untuk menampung data barang yang akan expired dan barang dengan stok hampir habis.

- Perancangan Flowchart Program

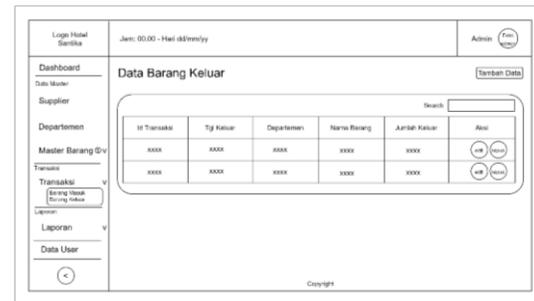
Bagan alir (flowchart) merupakan sebuah bagan yang menunjukkan alir (flow) di dalam program atau prosedur sistem secara logika[3]. Bagan alir digunakan untuk alat bantu komunikasi dan untuk dokumentasi.



Sedangkan flowchart program adalah bagan yang menjelaskan secara detail langkah-langkah dari proses program.

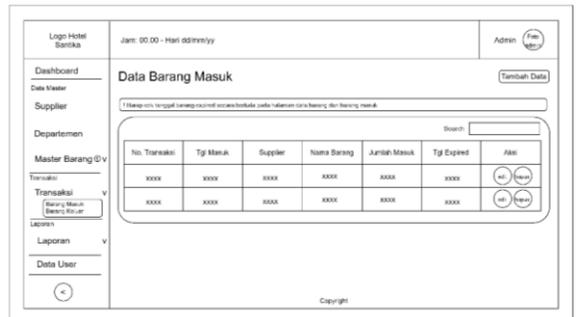
**Gambar 9 Flowchart Barang Masuk dan Barang Keluar**

Pada Gambar 9 dijelaskan bahwa menu transaksi terdiri dari beberapa sub-menu yaitu barang masuk dan barang keluar. Pada kedua sub-menu ini, admin dapat melakukan input data, edit data, hapus data dan cari data.

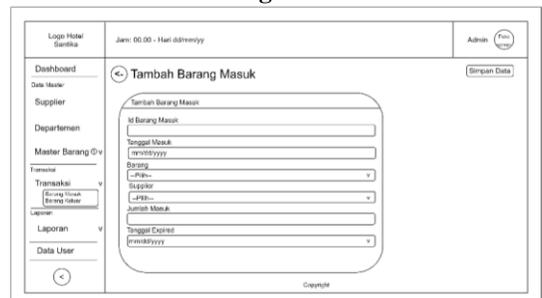


- Perancangan Antarmuka Sistem

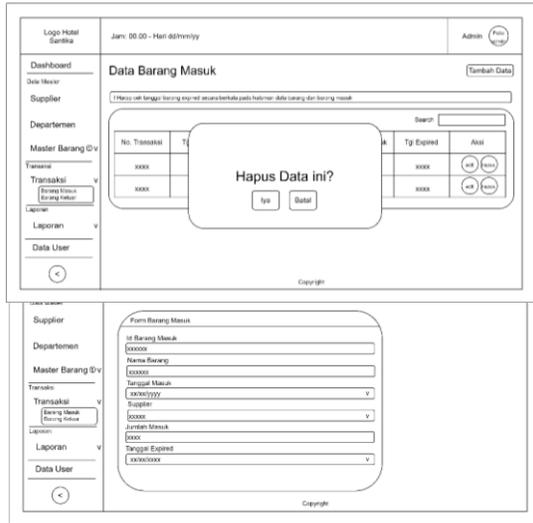
Perancangan antarmuka halaman barang masuk dan barang keluar digunakan oleh admin untuk melihat data, menambah data, mengedit data, dan menghapus data.



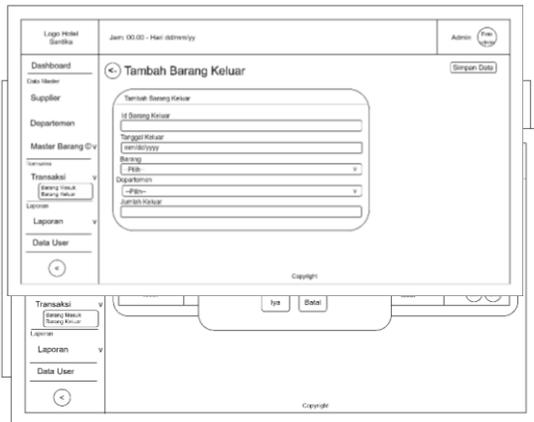
**Gambar 10 Perancangan Antarmuka Halaman Barang Masuk**



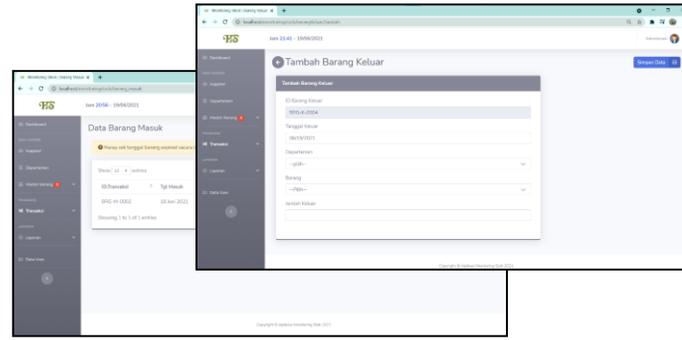
**Gambar 11 Perancangan Antarmuka Halaman  
Tambah Barang Masuk**  
**Gambar 12 Perancangan Antarmuka Halaman  
Edit Barang Masuk**



**Gambar 13 Perancangan Antarmuka Halaman  
Hapus Barang Masuk**  
**Gambar 14 Perancangan Antarmuka Halaman  
Barang Keluar**  
**Gambar 15 Perancangan Antarmuka Halaman  
Tambah Barang Keluar**  
**Gambar 16 Perancangan Antarmuka Halaman  
Edit Barang Keluar**  
**Gambar 17 Perancangan Antarmuka Halaman  
Hapus Barang Keluar**

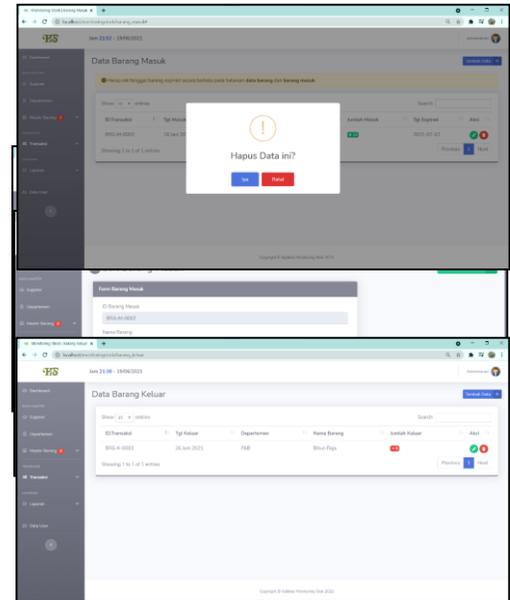


c. *Implementation*



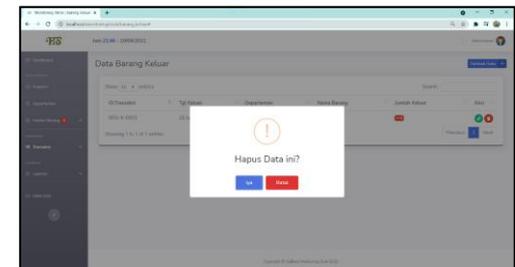
- Pada implementasi akan dijelaskan tampilan aplikasi pada menu barang masuk dan barang keluar.

**Gambar 18 Implementasi Halaman Barang Masuk**  
**Gambar 19 Implementasi Halaman Tambah Barang  
Masuk**  
**Gambar 20 Implementasi Halaman Edit Barang  
Masuk**



**Gambar 21 Implementasi Halaman Hapus Barang  
Masuk**  
**Gambar 22 Implementasi Halaman Barang Keluar**  
**Gambar 23 Implementasi Halaman Tambah Barang  
Keluar**  
**Gambar 24 Implementasi Halaman Edit Barang  
Keluar**  
**Gambar 25 Implementasi Halaman Hapus Barang  
Keluar**

- Pengujian *Blackbox*  
Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak, dapat diartikan bahwa



teknik pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program[7].

**Tabel 2 Hasil Uji Halaman Barang Masuk dan Barang Keluar**

No.	Item Pengujian	Hasil yang Diterapkan	Hasil Pengujian	Keterangan
<b>Sub-Menu Barang Masuk</b>				
1.	Button Tambah Barang Masuk dan mengisi data yang wajib diisi pada form tambah	Sistem menerima	Sukses dan data akan tersimpan di database serta di update pada tampilan halaman barang masuk	[✓] Diterima [ ] Ditolak
2.	Button Tambah Barang Masuk dan tidak mengisi data yang wajib diisi pada form tambah	Sistem menolak	Data tidak akan tersimpan di database dan akan muncul popup peringatan	[✓] Diterima [ ] Ditolak
3.	Button Edit Barang Masuk	Sistem menerima	Sukses dan data akan tersimpan di database serta di update pada tampilan halaman barang masuk	[✓] Diterima [ ] Ditolak
4.	Button Hapus Satuun Barang	Sistem menerima	Sukses dan data akan terhapus di database serta di update pada tampilan halaman barang masuk	[✓] Diterima [ ] Ditolak
<b>Sub-Menu Barang Keluar</b>				
5.	Button Tambah Barang Keluar dan mengisi data yang wajib diisi pada form tambah	Sistem menerima	Sukses dan data akan tersimpan di database serta di update pada tampilan halaman barang keluar	[✓] Diterima [ ] Ditolak
6.	Button Tambah Barang Keluar dan tidak mengisi data yang wajib diisi pada form tambah	Sistem menolak	Data tidak akan tersimpan di database dan akan muncul popup peringatan	[✓] Diterima [ ] Ditolak
6.	Button Edit Data Barang	Sistem menerima	Sukses dan data akan tersimpan di database serta di update pada tampilan halaman barang keluar	[✓] Diterima [ ] Ditolak
7.	Button Hapus Data Barang	Sistem menerima	Sukses dan data akan terhapus di database serta di update pada tampilan halaman barang keluar	[✓] Diterima [ ] Ditolak

• Pengujian *Whitexbox*

Pengujian *whitebox* disebut juga pengujian kotak kaca (*glass box testing*), merupakan sebuah filosofi perancangan *test case* yang menggunakan struktur kontrol yang dijelaskan sebagai bagian dari perancangan peringkat komponen untuk menghasilkan *test case* [7].

```

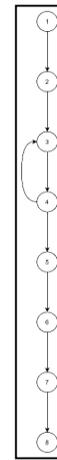
        $data = array(
            'id_barang_masuk' => $idMasuk,
            'id_barang' => $barang,
            'id_supplier' => $supplier,
            'jumlah_masuk' => $jumlahMasuk,
            'tgl_masuk' => $tglMasuk,
            'id_user' => $supplier,
            'tgl_expired' => $expired
        );
        $barang = array('id_barang' => $barang);
        $this->barangMasuk_model->save($data, $barang);

        public function tambah_data($data, $barang)
        {
            $this->db->insert($data, $barang);
            $sql = "UPDATE barang SET stok = stok + ($data['jumlah_masuk']) WHERE id_barang = ($data['id_barang'])";
            $this->db->query($sql);
        }

        public function stok_tampil()
        {
            $barang = $this->db->get($barang->nama());
            $limit = 0;
            foreach ($barang as $s) {
                if ($s->stok <= $s->stok_min) {
                    $limit++;
                }
                $data['limit_total'] = $limit;
                $this->db->update('barang_ditel', $data, ['id' => $s]);
            }

            <?php if ($limit_barang_ditel_total > 0) :>
                <div class="alert alert-warning" role="alert">
                    <i class="fas fa-exclamation-triangle"> </i> Ada <?php echo $limit_barang_ditel_total; ?> barang dengan stok hampir habis!
                </div>
            <?php endif; ?>
    
```

**Gambar 26 Source Code Barang Masuk**



**Gambar 27 Grafik Alir Barang Masuk**

```

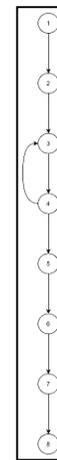
        $data = array(
            'id_barang_keluar' => $idKeluar,
            'id_barang' => $barang,
            'jumlah_keluar' => $jumlahKeluar,
            'tgl_keluar' => $tglKeluar,
            'id_supplier' => $supplier
        );
        $barang = array('id_barang' => $barang);
        $this->barangKeluar_model->save($data, $barang);

        public function tambah_data($data, $barang)
        {
            $this->db->insert($data, $barang);
            $sql = "UPDATE barang SET stok = stok - ($data['jumlah_keluar']) WHERE id_barang = ($data['id_barang'])";
            $this->db->query($sql);
        }

        public function stok_tampil()
        {
            $barang = $this->db->get($barang->nama());
            $limit = 0;
            foreach ($barang as $s) {
                if ($s->stok <= $s->stok_min) {
                    $limit++;
                }
                $data['limit_total'] = $limit;
                $this->db->update('barang_ditel', $data, ['id' => $s]);
            }

            <?php if ($limit_barang_ditel_total > 0) :>
                <div class="alert alert-warning" role="alert">
                    <i class="fas fa-exclamation-triangle"> </i> Ada <?php echo $limit_barang_ditel_total; ?> barang dengan stok hampir habis!
                </div>
            <?php endif; ?>
    
```

**Gambar 28 Source Code Barang Keluar**



**Gambar 29 Grafik Alir Barang Keluar**

Berdasarkan grafik alir (*flowgraph*) diatas, dapat di hitung *cyclomatic complexity flowgraph* barang masuk, berikut adalah perhitungan kompleksitas siklotmatiknya.

$$V(G) = E - N + 2$$

- Ket : - E : jumlah *edge* (anak panah) grafik alir
- N : jumlah *node* grafik alir

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 8 - 8 + 2 \\ &= 2 \end{aligned}$$

Setelah dihitung nilai *cyclomatic complexity*, didapatkan bahwa  $V(G) = 2$ , maka dapat disimpulkan bahwa aplikasi ini memiliki prosedur yang sederhana dan tingkat risiko yang rendah. *Path* yang dihasilkan yaitu :

**Tabel 3 Path Barang Masuk-Keluar yang Dihasilkan**

Path	Hasil	Tipe Prosedur dan Risiko
1	1-2-3-4-5-6-7-8	Prosedur sederhana, risiko rendah ( <i>low risk</i> )
2	1-2-3-4-3-4-5-6-7-8	Prosedur sederhana, risiko rendah ( <i>low risk</i> )

Berdasarkan pengujian *white box* yang telah dilakukan maka dapat disimpulkan Aplikasi *Monitoring Stock* Barang ini memiliki prosedur yang sederhana dan tingkat risiko yang rendah.

#### IV. KESIMPULAN DAN SARAN

##### A. Kesimpulan

Setelah melakukan kerja praktek di Hotel Santika Radial, dibuatlah Aplikasi *Monitoring Stock* Barang dengan Fitur Notifikasi Berbasis Web pada Hotel Santika Radial Palembang sebagai solusi atas permasalahan yang ada. Adapun hasil dari pembuatan aplikasi ini adalah sebagai berikut:

1. Aplikasi dapat menyimpan data *supplier*, departemen, satuan barang, barang, barang *expired*, barang masuk, barang keluar, dan data *user*.
2. Aplikasi dapat menampilkan dan mengunduh laporan barang masuk, barang keluar, dan stok barang.
3. Aplikasi dapat memberikan notifikasi sebagai pengingat (*reminder*) bagi *staff store keeper* terkait stok barang yang hampir habis dan barang yang sudah dan/atau akan *expired* pada bulan yang berjalan.
4. Kehilangan dan kerusakan data dapat diminimalisir dengan menyimpan data pada *database*.
5. Meminimalisir terjadinya kelalaian *staff store keeper* dalam memenuhi permintaan barang, terutama untuk barang yang memiliki masa pakai (masa *expired*).
6. Laporan ini dibuat sebagai referensi bagi Hotel Santika Radial Palembang yang belum memiliki aplikasi terkhusus untuk *me-monitor* barang dan stok barang.

##### B. Saran

Adapun saran yang dapat dilakukan untuk pengembangan sistem di tahap selanjutnya yaitu :

1. Perlunya pengembangan pada transaksi barang keluar, agar pada satu id transaksi dapat terdiri dari beberapa permintaan barang (misalnya 1 departemen dapat meminta 10 barang yang berbeda).
2. Perlunya pengembangan pada aplikasi terkait pengguna, untuk saat ini hanya satu level *user*, jika kedepannya dibutuhkan maka sebaiknya

dikembangkan agar aplikasi dapat menjadi *multi level user*.

3. Menambah fasilitas printer agar laporan dapat langsung dicetak.

#### DAFTAR PUSTAKA

- [1] Britton, Carol. 2001. *Object-Oriented Systems Development*. McGraw-Hill. hlm. 28–29, 269. ISBN 0-07-709544-8.
- [2] Gata, Windu, dan Grace Gata. 2013. *Sukses Membangun Aplikasi Penjualan Dengan Java*. Jakarta: Elex Media Komputindo.
- [3] Jogiyanto, H. M. 2005. *Analisis dan Desain (Sistem Informasi, Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis)*. Yogyakarta : Andi.
- [4] Kendall, Kenneth E. 2008. *System Analysis, System Design*. Prentice Hall : New Jersey.
- [5] Noor, Juliansyah. 2011. *Metodologi Penelitian*. Jakarta : Prenada Media Group.
- [6] Pressman, Roger, S. 2002. *Rekayasa Perangkat Lunak (pendekatan praktisi)* Yogyakarta : Andi
- [7] Pressman, Roger S. 2012. *Rekayasa Perangkat Lunak (Pendekatan Praktisi) Edisi 7 : Buku I*. Yogyakarta : Andi.
- [8] Santika Indonesia, 2019. About Us. <https://www.mysantika.com/generic/about>. Diakses pada tanggal 10 Mei 2021.



**Ivana Celesta.** Lahir di Palembang tanggal 17 September 2001. Saat ini merupakan mahasiswi Universitas Katolik Musi Charitas Palembang dengan program studi S1 Sistem Informasi Angkatan 2018. Merupakan penerima beasiswa Siswa Berprestasi Universitas Katolik Musi Charitas Palembang. Saya pernah tergabung dengan Himpunan Mahasiswa

Program Studi Sistem Informasi pada tahun 2018-2021, Badan Eksekutif Mahasiswa pada tahun 2019-2020.



Maria Bellanier Ismiati, S.Kom., M.Eng. lahir di Palembang pada tanggal 29 Desember 1989. Penulis mendapatkan gelar S.Kom. dari Universitas Kristen Duta Wacana Yogyakarta sedangkan untuk gelar S2 yaitu M.Eng. didapatkan dari Universitas Gadjah Mada Yogyakarta. Saat ini bekerja sebagai

dosen di Universitas Katolik Musi Charitas Palembang. Adapun bidang ilmu yang ditekuni saat ini adalah Artificial Intelligence, Statistika Inferensia