**IMPLEMENTASI METODE *TIMEOPTIMIZED* DAN *BUDGETOPTIMIZED* DALAM FITUR PENGAMBILAN KEPUTUSAN PADA APLIKASI LAPORAN *INVENTORY* PT.TOTAL PRINT**



**FIRDAUS ANGGA DEWANGGA**

**1112001013**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK DAN ILMU KOMPUTER**

**UNIVERSITAS BAKRIE**

**JAKARTA**

**2016**

# **HALAMAN PERNYATAAN ORISINALITAS**

Tugas Akhir ini adalah hasil karya saya sendiri,

dan semua sumber baik yang dikutip maupun yang dirujuk

telah saya nyatakan dengan benar

Nama : Firdaus Angga Dewangga

NIM : 1112001013

Tanda Tangan

Tanggal : 2 Juni 2016

# **HALAMAN PENGESAHAN**

Judul Penelitian : Implementasi Algoritma *Time Optimization* dan *Budget Optimization* Dalam Fitur Pengambilan Keputusan pada Aplikasi *Inventory* PT.TOTALPRINT

Peneliti Utama : Firdaus Angga Dewangga

Jenis Kelamin : Laki-Laki

Unit Kerja : Program Studi Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Bakrie

Alamat Kerja : Gelanggang Mahasiswa GOR Soemantri Brojonegoro Suite GF-22 Jl.HR.Rasuna Said Kav C-22, Jakarta Selatan

Alamat Email : [Dewangga.Firdaus@gmail.com](mailto:Dewangga.Firdaus@gmail.com)

Lama Penelitian : 9 (Sembilan) Bulan

Usulan Penelitian Tahun : 2016

Menyetujui Jakarta, 2 Juni 2016

Pembimbing Tugas Akhir Peneliti

Yusuf Lestanto, ST., M.Sc Firdaus Angga Dewangga

# **KATA PENGANTAR**

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT, yang telah memberikan rahmat, berkat, taufik dan hidayah-Nya, sehingga penulisan tugas akhir dengan judul ” Implementasi Algoritma *Time Optimization* dan *Budget Optimization* Dalam Fitur Pengambilan Keputusan pada Aplikasi *Inventory* PT.TOTAL PRINT” dapat diselesaikan dengan baik.

Penulisan tugas akhir ini dibuat sebagai syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika Universitas Bakrie. Penyusunan tugas akhir ini tidak lepas dari bantuan berbagai pihak yang telah mendorong dan membimbing penulis, baik dari ide, pemikiran dan semangat. Oleh karena itu dalam kesempatan ini penulis ingin mengucapkan terima kasih sebesar-besarnya kepada :

1. Orang tua penulis, Bapak Hardi Erwiyanto dan Ibu Darul Khayati S.E serta adik penulis, Wildan Firman Nurrahman, Ivan Muhammad Fadzilah dan Kayla Ardiya Khalisha yang tak henti-hentinya memberikan dukungan dan semangat sampai saat ini.

2. Bapak Yusuf Lestanto, ST., M.Sc selaku dosen pembimbing yang dengan sabar telah memberikan bimbingan, arahan, dukungan dan solusi dalam penelitian yang dilakukan hingga penulisan tugas akhir ini dapat terselesaikan.

Penulis berharap semoga semua yang ditulis dan dikerjakan di dalam tugas akhir ini dapat bermanfaat bagi pembaca dan semua pihak terkait. Tidak ada manusia yang sempurna sehingga penulis mohon maaf atas segala kekurangan yang ada. Kritik dan saran yang membangun sangat diharapkan untuk menyempurnakan tugas akhir ini.

Jakarta, 2 Juni 2016

Penulis

# **HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI**

Sebagai sivitas akademik Universitas Bakrie, saya yang bertanda tangan dibawah ini :

Nama : Firdaus Angga Dewangga

NIM : 1112001013

Program Studi : Informatika

Fakultas : Fakultas Teknik dan Ilmu Komputer

Judul Skirpsi : Implementasi Algoritma *Time Optimization* dan *Budget Optimization* Dalam Fitur Pengambilan Keputusan pada Aplikasi *Inventory* PT.TOTALPRINT

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Bakrie **Hak Bebas Royalti Noneksklusif (Non-exclusive-Free Right)** atas karya ilmiah saya yang berjudul :

**Implementasi Algoritma *Time Optimization* dan *Budget Optimization* Dalam Fitur Pengambilan Keputusan pada Aplikasi *Inventory* PT.TOTALPRINT**

Beserta perangkta yang ada (Jika Dibutuhkan). Dengan hak bebas royalti noneksklusif ini Universitas Bakrie berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat dan mempublikasikan tugas akhir ini selama tetap mencantumkan nama penulis sebagai pemilih Hak Cipta untuk kepentingan akademis.

Demikian pernyataan ini dibuat dengan sebenar-benarnya.

Dibuat di : Jakarta

Pada tanggal : 2 Juni 2016

Yang mengatakan

Firdaus Angga Dewangga

**IMPLEMENTASI ALGORITMA *TIME OPTIMIZATION* DAN *BUDGET OPTIMIZATION* DALAM FITUR PENGAMBILAN KEPUTUSAN PADA APLIKASI *INVENTORY* PT.TOTALPRINT**

Firdaus Angga Dewangga

# **ABSTRAK**

PT. TOTALPRINT pada saat ini belum mempunyai sistem basis data dalam mengolah informasi laporan yang dimilikinya. Hal ini menyebabkan antara pihak perusahaan pusat dan cabang harus melakukan pengetikan ulang jika ingin menyatukan semua laporan yang ada. Oleh karena itu dibutuhkan sebuah aplikasi dengan mengeluarkan hasil berupa pengelolaan data laporan yang baik serta akurat agar proses kerja dalam laporan data perusahaan PT.TOTALPRINT lebih efisien. Aplikasi tersebut juga ditambahkan dengan fitur pengambilan keputusan dalam pengelolaannya dengan menggunakan metode Algoritma *TimeOptimization* dan *BudgetOptimization*. Aplikasi tersebut menggunakan Bootstrap, HTML5 dan CSS sebagai tampilan aplikasi, PHP sebagai bahasa yang digunakan untuk mengaplikasikan metode *TimeOptimization* dan *BudgetOptimization*, dan PhpMyAdmin sebagai *database.* Metode Algoritma *TimeOptimization* dan *BudgetOptimization* dinilai cocok dikarenakankemampuannya dalam memaksimalkan hasil yang diinginkan dengan keterbatasan yang diberikan.

**Kata** **Kunci:** Pengambilan Keputusan, Aplikasi Pengelolaan Inventory, *Timeoptimized*, *Budgetoptimized.*

**DAFTAR ISI**

[**HALAMAN PERNYATAAN ORISINALITAS** ii](#_Toc452729841)

[**HALAMAN PENGESAHAN** iii](#_Toc452729842)

[**KATA PENGANTAR** iv](#_Toc452729843)

[**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI** v](#_Toc452729844)

[**ABSTRAK** vi](#_Toc452729845)

[**DAFTAR TABEL** ix](#_Toc452729846)

[**BAB I** 1](#_Toc452729847)

[**PENDAHULUAN** 1](#_Toc452729848)

[1.1 Latar Belakang Masalah 1](#_Toc452729849)

[1.2 Rumusan Masalah 3](#_Toc452729850)

[1.3 Batasan Masalah 4](#_Toc452729851)

[1.4 Tujuan Penelitian 4](#_Toc452729852)

[1.5 Manfaat penelitian 5](#_Toc452729853)

[**BAB II** 6](#_Toc452729854)

[**TINJAUAN PUSTAKA** 6](#_Toc452729855)

[2.1 Penelitian Terdahulu 6](#_Toc452729856)

[2.2 Landasan Teori 8](#_Toc452729857)

[2.2.1 Manajemen Aset 8](#_Toc452729858)

[2.2.2 Aplikasi Berbasis *Web* 8](#_Toc452729859)

[2.2.3 Algoritma *Timeoptimized* dan *Budgetoptimized* 9](#_Toc452729860)

[2.2.4 Model *Waterfall* 12](#_Toc452729861)

[**BAB III** 15](#_Toc452729862)

[**METODE PENELITIAN** 15](#_Toc452729863)

[3.1 Observasi Awal 15](#_Toc452729864)

[3.2 Identifikasi Masalah 15](#_Toc452729865)

[3.3 Tinjauan Pustaka 16](#_Toc452729866)

[3.4 Pengumpulan Data 16](#_Toc452729867)

[3.5 Analisis dan Desain 17](#_Toc452729868)

[3.5.1 *Use Case* Diagram 18](#_Toc452729869)

[3.5.2 *Flow Chart* System 19](#_Toc452729870)

[3.5.3 *Pseudocode* Sistem 19](#_Toc452729871)

[3.5.4 Activity Diagram 23](#_Toc452729872)

[3.5.5 Relasi Database 24](#_Toc452729873)

[3.5.6 Penerapan Algoritma *Timeoptimized* dan *Budgetoptimized* 24](#_Toc452729874)

[3.5.7 *Flowchart* Algoritma *Timeoptimized* 31](#_Toc452729875)

[3.5.8 *Flowchart* Algoritma *Budgetoptimized* 32](#_Toc452729876)

[3.5.9 *Pseudocode* Algoritma *Timeoptimized* 33](#_Toc452729877)

[3.5.10 *Pseudecode* Algoritma *Budgetoptimized* 34](#_Toc452729878)

[3.6 Implementasi 35](#_Toc452729879)

[3.7 Pengujian 35](#_Toc452729880)

[3.7.1 Pengujian *WhiteBox* 37](#_Toc452729881)

[3.7.2 Pengujian *BlackBox* 37](#_Toc452729882)

[3.8 Hasil dan Laporan 39](#_Toc452729883)

[**DAFTAR PUSTAKA** 40](#_Toc452729884)

**DAFTAR GAMBAR**

Gambar 2.1 *Web based System* (Pressman, 2001) …………………………… 9

Gambar 2.2 Waterfall Model (Bassil, 2012) ………………………………… 12

Gambar 3.1 Fase Penelitian (Sadanti, 2016) ………………………………… 15

Gambar 3.2 *Use Case* Sistem ……………………………………………….. 18

Gambar 3.3 Admin *Flowchart* …………………………………………….... 19

Gambar 3.4 *Activity Diagram* ……………………………………………… 23

Gambar 3.5 Relasi *Database* ………………………………………………. 24

Gambar 3.6 *FlowChart* Algoritma *Timeoptimized* ………………………… 31

Gambar 3.7 *FlowChart* Algoritma *Budgetoptimized* ………………………. 32

Gambar 3.8 Gambaran Umum Pengujian Aplikasi ……………………….. 36

Gambar 3.9 Gambaran Proses Pengujian *WhiteBox* ……………………….. 37

Gambar 3.10 Gambaran Umum Proses Pengujian *BlackBox* ………………… 38

# **DAFTAR TABEL**

Tabel 2.1 Penelitian Terdahulu ……………………………………………7

Tabel 2.2 Perbandingan Algoritma ……………………………………….. 11

Tabel 3.1 Waktu Perjalanan Setiap Perusahaan (Menit) …………………. 27

Tabel 3.2 Waktu Pelayanan Setiap Layanan (Menit) ……………………. 27

Tabel 3.3 Tabel Keputusan Hasil Fungsi *Timeoptimized*Tempat ……….. . . 28

Tabel 3.4 Tabel Hasil Keputusan Fungsi *OptimizePlacebyObject* .............. 28

Tabel 3.5 Tabel *Brand* Printer dan Harga ………………………………… 29

Tabel 3.6 Tabel *Brand* Printer dan Performa ……………………………... 29

Tabel 3.7 Tabel Hasil Fungsi *Budgetoptimized*Bahan ……………………. 29

Tabel 3.8 Tabel Hasil Fungsi *BudgetoptimizedByObject …………………….* 29

# **BAB I**

# **PENDAHULUAN**

## Latar Belakang Masalah

Kehidupan manusia saat ini merupakan kehidupan yang serba digital, menyelesaikan masalah secara tepat, akurat dan cepat dengan biaya yang relatif murah merupakan fungsi penggunaan komputer sebagai alat bantu dalam menyelesaikan masalah. Setiap orang peduli dengan perkembangan teknologi yang tidak memiliki batas ruang dan waktu, memanfaatkan dan mengembangkan teknologi yang telah berkembang tersebut dalam berbagai instansi. Seiring dengan semakin pesatnya perkembangan teknologi dalam era globalisasi saat ini, kecepatan dan ketepatan kerja merupakan tuntutan yang harus terpenuhi dalam setiap instansi (Sahupala, 2013).

Perkembangan teknologi juga berdampak pada meningkatnya persaingan dunia industri saat ini, perusahaan dituntut untuk memiliki keunggulan kompetitif agar dapat bertahan di tingkat nasional dan internasional. Merencanakan perencanaan produksi dengan tepat merupakan salah satu cara yang dapat dilakukan. Perencanaan produksi berhubungan dengan penentuan *volume* produksi, ketepatan waktu penyelesaian dan utilisasi sumber daya yang tersedia. Dengan perencanaan yang tepat, proses produksi dapat berjalan efisien dan efektif sehingga berdampak pada peningkatan laba perusahaan (Junida, 2009).

Selain perencanaan produksi, persediaan barang berperan penting bagi perusahaan. Menurut Martani dkk (2012), perusahaan ritel, manufaktur, jasa, maupun entitas lainnya penting memiliki aset persediaan. PSAK 14 (revisi 2008) mendefinisikan persediaan sebagai aset yang (i) tersedia untuk dijual dalam kegiatan usaha biasa; (ii) dalam proses produksi untuk penjualan tersebut; (iii) dalam bentuk bahan atau perlengkapan (*supplies*) untuk digunakan dalam proses produksi atau pemberian jasa.

Berdasarkan wawancara dengan Bapak Dede Muhariramsyah, PT.TOTAL PRINT juga memiliki permasalahan yang sama, yaitu permasalahan dalam pengelolaan laporan persediaan stok barang. Menurut beliau, dikarenakan belum adanya sistem basis data dalam pengelolaan laporan stok barang pada PT.TOTAL PRINT mengakibatkan seringnya terjadi keterlambatan dalam pengarsipan buku laporan perusahaan, terjadinya ketidak cocokan nilai data laporan dan stok barang di lapangan, serta kurangnya kelengkapan dalam laporan pemasukan barang masuk dan keluar. Selain itu metode yang digunakan dalam proses perencanaan yang dilakukan pada saat ini menurut beliau belum tentu yang terbaik, masih banyak cara lain yang dapat digunakan dalam proses perencanaan. Pemilihan sumber daya terbaik untuk kondisi dan masalah tertentu masih belum terselesaikan. Beberapa permasalahan tersebut jika tidak ditangani secara cepat, maka dapat berdampak buruk bagi perusahaan dimasa yang akan datang.

Penelitian ini dapat menyimpulkan terdapat dua sumber daya utama dalam pelaksanaan operasional yang terjadi pada PT.TOTAL PRINT yaitu sumber daya barang (*sparepart*, mesin dan bahan pakai) dan sumber daya manusia (teknisi). Berdasarkan permasalahan yang telah dijelaskan serta batasan sumber daya yang digunakan, penelitian ini mencoba membuat sebuah aplikasi *web* untuk pengelolaan laporan *inventory* dengan fitur pengambilan keputusan agar dapat mengatasi masalah kesulitan dalam pengambilan keputusan berdasarkan fakta dan data yang ada di lapangan. Terdapat 2 parameter yang dijadikan tolak ukur penelitian ini yaitu parameter biaya untuk mesin, parameter waktu untuk teknisi.

Dengan adanya permasalahan, tujuan, batasan dan parameter, dalam penelitian ini mencari beberapa jenis metode tepat guna dalam menangani permasalahan yang ada agar memberikan tujuan serta manfaat yang tepat terhadap aplikasi yang akan dibuat. Berdasarkan penelitian yang telah dilakukan, metode *Timeoptimized* dan *Budgetoptimized* yang dibuat oleh Sai Rahul Reddy (2006)cocok dengan permasalahan, batasan dan parameter yang akan diuji. Untuk penjelasan metode tepat guna akan dijelaskan lebih lanjut pada BAB II.

## 1.2 Rumusan Masalah

Berdasarkan penjelasan pada latar belakang masalah, maka dapat dirumuskan permasalahan dalam penelitian ini yaitu:

1. Bagaimana mengoptimalkan biaya sumber daya mesin dan waktu pemberian tugas pada teknisi di PT.TOTAL PRINT dengan aplikasi pengelolaan laporan *inventory* dengan fitur pengambilan keputusan?

## 1.3 Batasan Masalah

Agar permasalahan dan pemecahan masalah menjadi terarah, tidak menyimpang dari pokok masalah dan menghindari pembahasan yang terlalu luas maka perlu diberi batasan pada permasalahan yang ada, yakni:

* 1. Data stok barang yang digunakan untuk peramalan operasional adalah data stok barang dan teknisi 1 bulan terakhir.
  2. Fungsi kendala yang dibahas adalah jenis pekerjaan, performa mesin, batas biaya, batas waktu, waktu layanan, waktu perjalanan, dan biaya pembelianmesin.
  3. Fungsi pengambilan keputusan hanya dapat memberikan solusi untuk satu parameter yang diinginkan.
  4. Fungsi pengambilan keputusan hanya mempertimbangkan variabel yang dimasukkan ke dalam aplikasi, fungsi tidak memperhitungkan sesuatu diluar pertimbangan seperti variabel yang tidak dimasukkan ke dalam aplikasi, bencana alam atau sesuatu yang tidak terprediksi.

## 1.4 Tujuan Penelitian

Tujuan penelitian **“IMPLEMENTASI METODE *TIMEOPTIMIZED* DAN *BUDGETOPTIMIZED* DALAM FITUR PENGAMBILAN KEPUTUSAN PADA APLIKASI LAPORAN *INVENTORY* PT.TOTAL PRINT*”*** adalah sebagai berikut:

1. Penggunaan aplikasi *inventory* dan metode tepat guna dalam fitur pengambilan keputusan dapat mengoptimalkan biaya operasional mesin dan waktu layanan teknisi dengan batasan-batasan yang ada.

## 1.5 Manfaat Penelitian

Beberapa manfaat yang dapat diperoleh dari penelitian yang diteliti dalam penelitian ini adalah sebagai berikut:

1. Bagi penulis, penelitian ini dapat menjadi sarana pembelajaran ilmu pengetahuan yang telah diterima selama menjalani perkuliahan. Selain itu dapat melihat dan menerapkan suatu konsep ilmu di lapangan kerja nyata.
2. Bagi departemen, penelitian ini dapat menjadi literatur yang semakin memperkaya penerapan ilmu informatika di lapangan kerja nyata serta menjadi bahan literatur bagi penelitian oleh departemen maupun mahasiswa di kemudian hari.
3. Bagi perusahaan, penelitian ini berguna sebagai:
4. Masukan dalam perencanaan produksi untuk menentukan rencana operasional yang optimal agar dapat memaksimumkan kegiatan operasional perusahaan.
5. Masukan dalam memanfaatkan sumber daya yang tersedia seoptimal mungkin.

# **BAB II**

# **TINJAUAN PUSTAKA**

## 2.1 Rancang Bangun

Untuk rancang bangun aplikasi, penelitian ini menggunakan model *waterfall* .Model *Waterfall* merupakan sebuah pendekatan secara sekuensial dan sistematik dalam pengembangan perangkat lunak. Model linear yang berurutan dimulai pada tingkat sistem, lalu dilanjutkan dengan analisis, desain, implementasi, testing, dan *maintenance*. Model *Waterfall* sendiri bersifat berkelanjutan yaitu dilakukan secara bertahap sebelum lanjut ke tahap selanjutnya (Pressman, 2001).

Terdapat beberapa keunggulan model *waterfall* jika digunakan sebagai pendekatan pengembangan *software*. Beberapa keunggulan tersebut adalah pencerminan kepraktisan rekayasa yang membuat kualitas *software* tetap terjaga karena pengembanganya yang terstruktur dan terawasi. Selain itu model ini bersifat dokumen lengkap sehingga proses pemeliharaan dapat dilakukan dengan mudah. Model *Waterfall* melibatkan 6 tahapan pada pendekatannya dan setiap tahapan selalu dilakukan verifikasi atau *testing* (S.Sadanti, 2016). Tahapan model *Waterfall* meliputi:



**Gambar 2.2 *Waterfall* Model** (Bassil, 2012)

1. ***Software Requirement Analysis***

Proses pertama pada model *Waterfall* berfokus pada kebutuhan perangkat lunak. *Programmer* harus benar-benar mengerti informasi seperti *behavior*, performa, antar muka program dan fungsi-fungsi yang dibutuhkan. Serta perlu dilakukan pendokumentasian dalam pembahasan spesifikasi antara sistem dan perangkat lunak bersama *user*.

1. ***Design***

Tahapan kedua dalam model *Waterfall* berfokus pada desain perangkat lunak. Desain perangkat lunak merupakan sebuah proses yang berfokus pada 4 atribut dalam sebuah program yaitu struktur data, antarmuka, representasi dan prosedural (algoritmik) rinci, serta arsitektur perangkat lunak. Pada tahap ini, kebutuhan berubah menjadi representasi dalam bentuk *blueprint* sebelum memulai implemetasi.

1. ***Implementation***

Tahapan ketiga berfokus dalam proses penulisan kode sumber aplikasi. Pada tahapan ini *blueprint* yang telah dibuat dilanjutkan dalam bentuk kode agar dapat dimengerti oleh mesin. Tahapan ini merupakan implementasi dari tahapan desain.

1. ***Testing***

Tahapan keempat dalam model *Waterfall* berfokus pada uji coba terhadap fungsi-fungsi *software*, hal ini bertujuan agar *software* bebas dari masalah dan hasilnya harus sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

1. ***Maintenance***

Pengembangan merupakan salah satu pemeliharaan suatu software, hal ini terjadi karena software yang dibuat tidak selamanya hanya seperti itu. Dalam penggunaannya mungkin saja masih terdapat *error* kecil yang belum ditemukan sebelumnya, atau ada penambahan fitur-fitur yang diperlukan dan belum ada pada *software* tersebut. Pengembangan diperlukan ketika adanya perubahan dari ektsternal perusahaan seperti adanya penggantian sistem operasi atau perangkat lain.

Penelitian ini menggunakan model *Waterfall* karena beberapa alasan dan pertimbangan perancangan agar pembangunan aplikasi yang dilakukan lebih terperinci dan terstruktur pada perencanaan pada setiap tahapan. Lalu dengan model yang sederhana dan jelas untuk diterapkan, memudahkan aplikasi untuk digunakan dan dikembangkan karena memiliki alur dan dokumentasi yang pasti dan jelas.

## 2.2 Penelitian Terdahulu

Penelitian pertama berjudul ***The Challenge of Normative Decision Making in Industries***(Kindi, Lawati, & Rawahi, 2015)*.* Penelitian tersebut memperkenalkan konsep analisis dalam pengambilan keputusan untuk membantu perusahaan percetakan di negara Oman dalam memilih mesin cetak. Penelitian tersebut bertujuan untuk menghasilkan konsep analisis pengambilan keputusan dalam memilih mesin cetak dengan membandingkan 2 mesin cetak menggunakan pendekatan normatif dalam analisis pengambilan keputusan. Penelitian kedua digunakan dalam penelitian ini untuk menjadi penelitian pendukung dalam konsep pembangunan fitur pengambilan keputusan PT.TOTAL PRINT.

Penelitian kedua berjudul ***Market Economy Based Resource Allocation in Grids*** (Raddy, 2006)***.*** Pada penelitian ketiga ini berfokusterhadap beberapa peraturan pengalokasian sumber daya untuk pengguna *grids computing,* dan membantu mereka dalam memilih sumber daya yang cocok sesuai pilihan mereka. *Grid computing* didefinisikan sebagai koordinasi pembagian sumber daya dan pemecahan masalah dalam organisasi *virtual* multi-intitusional yang dinamis. Selain membantu dalam memilih sumber daya yang cocok, dalam peraturan tersebut juga mencoba untuk mengoptimalkan parameter seperti rata-rata waktu penyelesaian, rata-rata biaya per pekerjaan, dan jumlah pekerjaan yang dapat terselesaikan dalam tenggat waktu yang ditentukan berdasarkan pilihan pengguna. Penelitian ketiga digunakan dalam penelitian ini untuk menjadi panutan algoritma yang akan digunakan dalam aplikasi fitur dan *inventory* PT.TOTAL PRINT.

**Tabel 2.1 Penelitian Terdahulu**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Judul** | **Pengarang** | **Tahun** | **Permasalahan** | **Hasil** |
| 1 | Rancang Bangun Sistem Informasi Monitoring Penyewaan Gedung dan InfrastrukturTeknologi Informasi (Studi Kasus: PT INDOSAT, TBK DAN MITRA). | Alviani Wahyuni Suyodti. | 2014 | Dalam divisi Rental site milik PT.INDOSAT belum memiliki sistem monitor dalam proses penyewaan dan belum ada sistem yang dapat memberikan laporan secara sistematis. | Sebuah sistem informasi monitoring yang dapat digunakan dalam penyewaan gedung dan infrastruktur. |
| 2 | *The Challange of Normative Decision Making in Industries* | Mahmood Al Kindi; Mujtaba Al Lawati; Hani Al Rawahi. | 2015 | Dalam memilih mesin cetak pada perusahaan cetak di Oman, para pembuat keputusan belum dapat menggunakan sistem pembuat keputusan karena kerumitan sistem. | Dengan memberikan penjelasan terhadap fase keputusan, preferensi pembuat keputusan, alternatif dan menampilkan analisis sistem pembuat keputusan, perusahaan |
| 3 | *Market Economy based Resource Allocation in Grids* | Sai Rahul Reddy P. | 2006 | Keputusan dalam memilih sebuah sumber daya dalam melakukan sebuah pekerjaan belum teroptimisasi sesuai referensi pembuat keputusan. | Memilih dan mengoptimisasi sumber daya dalam melakukan sebuah pekerjaan berdasarkan referensi pembuat keputusan. |

## 2.3 Landasan Teori

Dalam penyusunan landasan teori, terdapat teori-teori yang berhubungan dengan aplikasi berdasarkan kriteria dan spesifikasi kebutuhan yang telah dijelaskan oleh pengguna. Pengguna dari aplikasi laporan *inventory* adalah Bapak Dede Muhariramsyah, seorang kepala bidang administrasi PT TOTAL PRINT yang merupakan tempat objek penelitian dilakukan dan diteliti. Beberapa kriteria dan spesifikasi yang akan dijelaskan adalah sebagai berikut:

### 2.3.1 Manajemen Aset

Menurut R.B.Faiz & A.Edirisinghe (2009), aset yang bervariasi dalam sebuah sistem manajemen dapat memiliki berbagai komponen dengan perbedaan yang bermacam-macam pada fungsinya. Setiap perusahaan tentu memiliki kebutuhan aset yang berbeda sehingga membutuhkan sistem manajemen yang berbeda pula. Untuk mengklarifikasi aset tersebut maka organisasi tersebut yang akan menentukannya. Dalam penelitian Y.Maryono dkk (2010), salah satu kunci dalam manajemen informasi aset adalah ketersediaan informasi pada saat yang tepat, dalam format yang tepat, untuk orang yang tepat, dengan *query* yang tepat, dan pada tingkatan yang tepat.

### 2.3.2 Algoritma *Timeoptimized* dan *Budgetoptimized*

Algoritma *Timeoptimized* merupakan sebuah algoritma yang mencoba untuk mengoptimalkan rata-rata waktu dengan menggunakan sumber daya yang diberikan dalam sebuah pekerjaan serta meningkatkan jumlah pekerjaan yang dapat selesai dalam tenggat waktu yang diberikan. Dalam sebuah algoritma *Timeoptimized*, seorang supplier akan selalu mencari sebuah target pekerjaan yang bisa diselesaikan dalam waktu paling cepat dengan batas waktu dan biaya yang ditetapkan (Raddy, 2006).

Penerapan algoritma *Timeoptimized* biasanya digunakan dalam masalah yang mempertimbangkan satu parameter dengan kondisi yang dapat ditentukan oleh penggunanya. Berikut adalah *pseudocode* algoritma *Timeoptimized* (Raddy, 2006):

S= { NULL };

//mencari sumber daya yang cocok dengan biaya dan deadline yang ditentukan

For (i=0; i<n; i++){

//waktu minimal untuk mengerjakan pekerjaan tersebut

Exec\_time = job\_length / Ri.speed;

//waktu penyelesaian sumber daya Ri

Completion\_time = Ri.resource\_usage\_start\_time + exec\_time;

If ((budget >=(Ri.price \* exec\_time)) AND (completion\_time<=deadline)) {S= S U Ri;

}

}

Sort S by completion time;

Select R1 form S;

*Budgetoptimized* juga merupakan algoritma yang mencoba untuk mengoptimalkan rata-rata biaya yang dibutuhkan dalam sebuah pekerjaan serta meningkatkan banyak pekerjaan yang bisa selesai dalam batas waktu yang ditetapkan. Berikut *pseudocode* algoritma *Budgetoptimized* (Raddy, 2006):

S= {NULL };

//mencari sumber daya yang cocok dengan biaya dan deadline yang ditentukan

For (i=0; i<n; i++)

{

//waktu minimal untuk mengerjakan pekerjaan tersebut

Exec\_time = job\_length / Ri.speed;

//waktu penyelesaian sumber daya Ri.

Completion\_time = Ri.resource\_usage\_start\_time + exec\_time;

If ((budget >=(Ri.price \* exec\_time)) AND (completion\_time<=deadline)) {S= S U Ri;

}

}

Sort S by completion time;

//R1 merupakan sumber daya dengan biaya terendah

Select R1 from S;

//max\_speed adalah kecepatan maksimum sumber daya yang dapat terbeli

Penalty = R1.speed / max\_speed

// jumlah minimum untuk menyelesaikan pekerjaan

Min\_amount\_needed = (job\_length/R1.speed)\*R1.price;

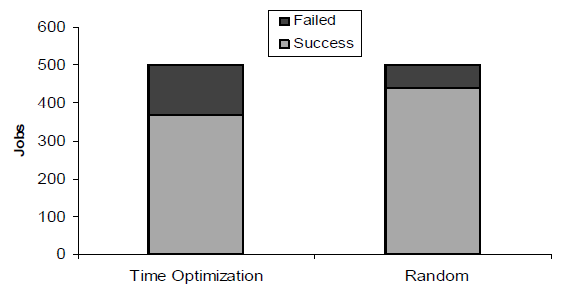
Bid\_amount = budget – budget – min\_amount\_needed) \* penalty;

Use bid\_amount to bid for R1;

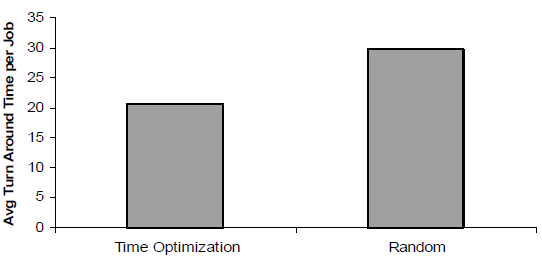
**2.2.3 Perbandingan Algoritma**

Terdapat beberapa metode dan algoritma tepat guna untuk melakukan penelitian yang berhubungan dengan optimisasi waktu dan biaya, beberapa metode tersebut telah diterangkan dan dijelaskan pada tesis yang dilakukan oleh Sai Rahul Raddy (2006). Dalam penelitiannya, Sai Rahul Raddy membandingkan 2 algoritma yaitu *Timeoptimized* dengan *Random*, *Budgetoptimized* dengan *Random*, dan memberikan sebuah algoritma baru yaitu *newBudgetoptimized* dan *newTimeoptimized*.

Dalam membandingkan penyelesaian rata-rata waktu dalam sebuah pekerjaan serta banyaknya pekerjaan yang dapat terselesaikan dalam batasan waktu tertentu, Sai Rahul Reddy membandingkan algoritma *Timeoptimized* dengan *Random* dan memberikan hasil sebagai berikut:

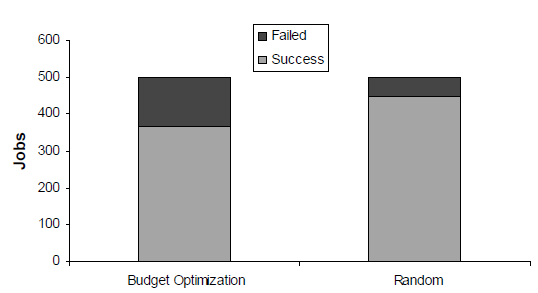


**Gambar 2.2 Skala Penyelesaian Pekerjaan *Timeoptimized* dan *Random*** (Raddy, 2006)**.**

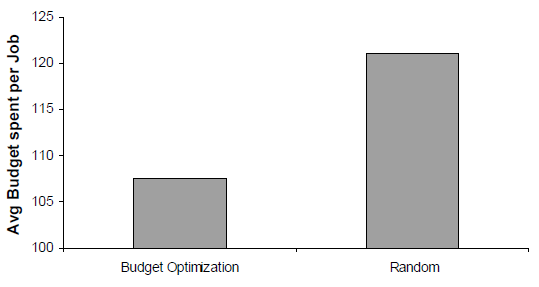


**Gambar 2.3 Rata-rata Waktu pada setiap Pekerjaan** (Raddy, 2006)**.**

Dalam penelitian yang dilakukan oleh Sai Rahul Reddy (2006) juga membandingkan rata-rata biaya dalam sebuah pekerjaan dan banyaknya pekerjaan yang dapat terselesaikan dalam biaya tertentu. Perbandingan tersebut menghasilkan hasil pada Gambar 2.4 dan Gambar 2.5:

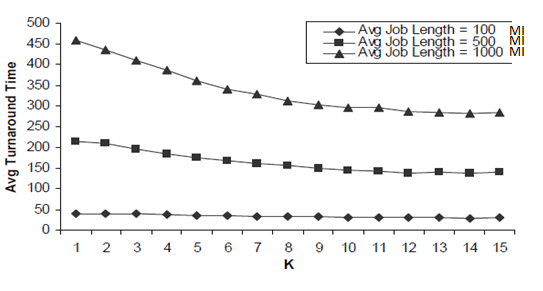


**Gambar 2.4 Skala Penyelesaian Pekerjaan Budget Optimization dan *Random*** (Raddy, 2006)**.**



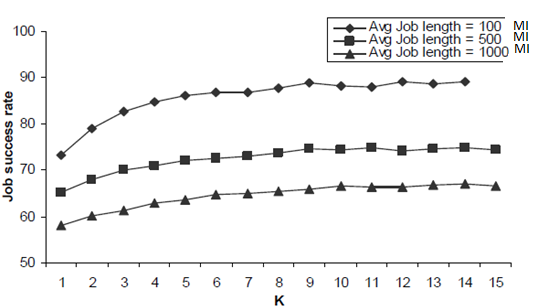
**Gambar 2.5 Rata-rata Biaya Pada Setiap Pekerjaan** (Raddy, 2006)**.**

Berdasarkan perbandingan yang telah dilakukan, Sai Rahul Reddy mengusulkan untuk memperbaharui *Timeoptimized* dan *Budgetoptimized* dengan menggabungkan kriteria algoritma *Random*. Pembaharuan tersebut menghasilkan sebuah algoritma *newTimeoptimized* dan *newBudgetoptimized* dan disimulasikan dengan parameter rata-rata waktu penyelesaian dalam sebuah pekerjaan, jumlah pekerjaan yang selesai atau sukses, dan rata-rata biaya dalam sebuah pekerjaan. Simulasi tersebut menghasilkan Gambar 2.6, Gambar 2.7, dan Gambar 2.8.



**Gambar 2.6 Rata-rata waktu penyelesaian dalam sebuah pekerjaan** (Raddy, 2006)**.**

Gambar 2.6 dan gambar 2.7 menampilkan hasil pengujian penggunaan algoritma newTimeOptimized dengan jumlah pekerjaan yang berbeda. Terlihat ketika kita memiliki banyak resource (k=n), tingkat kesuksesan meningkat dan waktu dalam melakukan pekerjaan berkurang.



**Gambar 2.7 Skala pekerjaan yang berhasil** (Raddy, 2006)**.**

**Tabel 2.2 Perbandingan Algoritma**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter | Algoritma | Pendekatan | Karakteristik | Kelebihan | Kekurangan |
| Rata-rata pekerjaan selesai *(Success Rate)* dan rata-rata penyelesaian waktu/biaya pekerjaan. *(Turnaround Time/ Turnaround Cost)* | *Timeoptimized & Budgetoptimized Policies(2006)* | Penerapan pada program yang ingin memilih *resource* yang dapat mengerjakan pekerjaan secepat mungkin dan dengan biaya sedikit mungkin. | Pemilihan resource selalu memilih sebuah *resource*. | Rata-rata penyelesaian waktu/biaya lebih kecil dibangdingkan dengan algoritma lain. | Penyelesaian pekerjaan yang gagal lebih tinggi dan rata-rata pekerjaan selesai lebih rendah. |
| Rata-rata pekerjaan selesai *(Success Rate)* dan rata-rata penyelesaian waktu/biaya pekerjaan *(Turnaround Time/ Turnaround Cost).* | *Random Policy(2006)* | Penerapan pada program yang ingin memilih *resource* secara *random* untuk melakukan sebuah pekerjaan. | Pemilihan *resource* selalu dilakukan dengan memilih secara acak. | Penyelesaian pekerjaan yang gagal lebih rendah dan rata-rata pekerjaan selesai lebih tinggi di bandingkan algoritma lain. | Rata-rata waktu/biaya dalam penyelesaian pekerjaan lebih tinggi. |
| Rata-rata pekerjaan selesai *(Success Rate)* dan rata-rata penyelesaian waktu/biaya pekerjaan. *(Turnaround Time/ Turnaround Cost).* | *newTimeoptimized* & new*Budgetoptimized* *Policies(2006)* | Penerapan pada program yang ingin menggunakan *policies* gabungan antara *TimeOptimization* & *BudgetOptimization* dengan Random *policy*. | Pemilihan *resource* untuk melakukan sebuah pekerjaan dapat berubah-ubah sesuai dengan keinginan *user*. | Mempunyai fleksibilitas dalam menentukan hasil. | Rata-rata pekerjaan selesai dan rata-rata waktu/biaya dalam menyelesaikan pekerjaan berubah-ubah. |

Tabel 2.2 menjelaskan tentang perbandingan antara algoritma *Timeoptimized* & *Budgetoptimized* dengan Algoritma *Random* dan *newTimeoptimized* & *newBudgetoptimized* sebagai sesama algoritma penemu *resource* terbaik untuk melakukan sebuah pekerjaan sesuai kondisi yang diinginkan dengan menggunakan satu parameter. Namun berdasarkan penjelasan sebelumnya dan berdasarkan wawancara yang telah dilakukan, algoritma *Timeoptimized* & *Budgetoptimized* menjadi pilihan sebagai algoritma yang digunakan karena r*esource* yang tersedia pada PT.TOTAL PRINT tidaklah acak. *Resource* ditentukan dan dipilih berdasarkan kebutuhan untuk menyelesaikan tugas tertentu. Algoritma *Timeoptimized* & *Budgetoptimized* diharapkan dapat menyelesaikan masalah dengan memberikan solusi keputusan terbaik dalam memilih suatu pekerjaan atau suatu mesin cetak.

# **BAB III**

# **METODE PENELITIAN**

Untuk melihat garis besar pada metode penelitian dapat menggunakan gambar dibawah ini:

1. Observasi

Awal

2. Identifikasi

Masalah

3. Tinjauan

Pustaka

4. Pengumpulan

Data

8. Hasil dan

Laporan

7. Pengujian

6. Implementasi

5. Analisis dan

Desain

**Gambar 3.1 Fase Penelitian**

## 3.1 Observasi Awal

Pada tahap pertama penelitian ini difokuskan pada pengamatan proses kegiatan yang terdapat pada sumber daya PT.TOTAL PRINT. Beberapa proses yang diamati dan diobservasi adalah laporan *inventory*. Berdasarkan pengamatan yang terjadi di lapangan, penelitian akan berfokus kepada proses bagian tersebut. Alasan ini terjadi karena hal tersebut memiliki dampak langsung terhadap pendapatan keuntungan yang terjadi pada perusahaan tersebut. Tahapan ini dilakukan berdasarkan wawancara dan observasi dengan seluruh pihak yang terkait dalam proses pengelolaan laporan *inventory.*

## 3.2 Identifikasi Masalah

Tahap kedua setelah dilakukannyaobservasi awal adalah identifikasi masalah. Pada tahapan ini, penelitian akan berfokus terhadap masalah yang terjadi dalam proses bagian yang diamati. Identifikasi masalah dapat membantu menganalisis hasil dari observasi awal*.* Permasalahan secara real dapat mendukung dan membantu penelitian, serta analisis dapat mengarahkan penelitian kepada tujuan dari penelitian ini.

## 3.3 Tinjauan Pustaka

Tahap ketiga dalam penelitian ini berfokus pada pengetahuan dan ilmu dasar dari pelaksanaan penelitian. Tahap ini berguna untuk membantu mendefinisikan, menentukan, dan menyingkap kesimpulan data untuk dapat dipresentasikan dan dimengerti oleh siapapun. Beberapa peran penting yang dapat diambil adalah sebagai pedoman untuk mencapai tujuan penelitian dan juga menyediakan beberapa teori pendukung yang menguatkan analisis dari pengamatan. Hasil dari pengamatan akan menjadi lebih valid karena telah diuji dalam pengamatan yang telah dilakukan dari beberapa teori dan terminology sebelumnya. Pada tahapan ini, penelitian berfokus kepada pencarian jurnal terdahulu untuk dijadikan acuan terhadap penelitian yang akan dilakukan. Tahap ini membahas metode pengembangan yang digunakan sebagai perancangan hingga implementasi aplikasi yang akan dibuat. Studi algoritma yang akan diterapkan mempunyai beberapa proses perbandingan untuk menentukan algoritma tepat guna dalam penggunaan penelitian ini dan membantu sebagai teori penguat. Dalam penelitian yang telah dilakukan, terdapat beberapa algoritma yang dapat ditemukan menggunakan kata kunci algoritma pembuat keputusan dalam optimisasi sumber daya. Algoritma *Timeoptimized* dan *Budgetoptimized* merupakan salah satu algoritma optimasi yang mempunyai banyak fungsi untuk menentukan hasil yang optimal sesuai dengan parameter yang diuji dalam aplikasi tersebut.

## 3.4 Pengumpulan Data

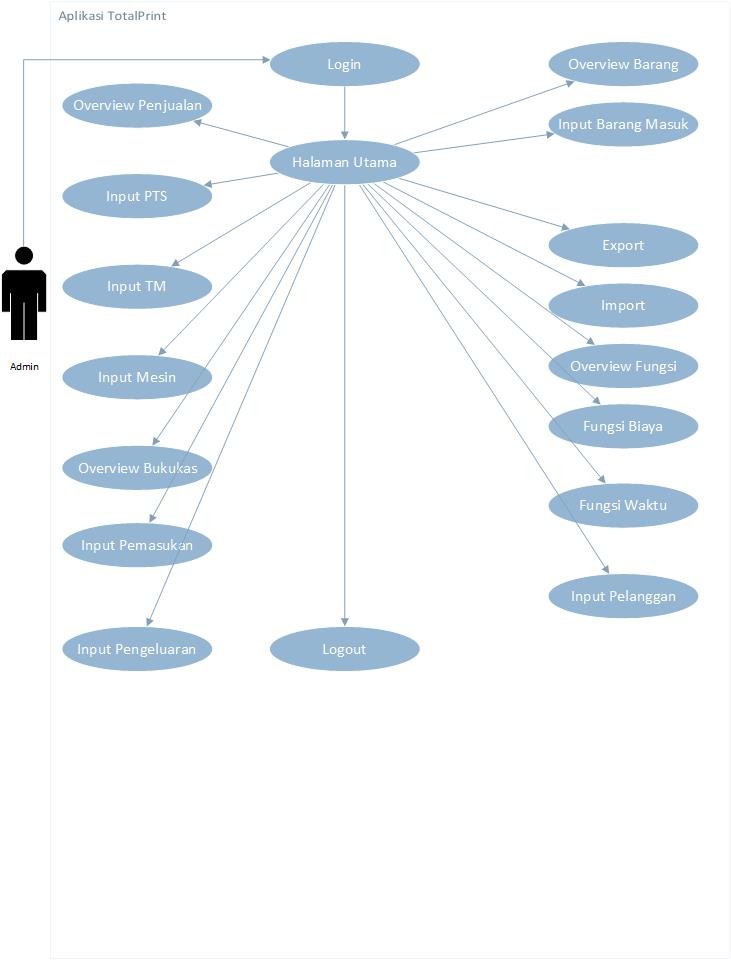
Pada tahapan penelitian ini berfokus terhadap pengumpulan data. Pengumpulan data dilakukan dengan melakukan wawancara dengan pengguna sistem yang akan memiliki hasil dari penelitian ini yaitu pihak administrasi PT.TOTAL PRINT, Bapak Dede Muhariramsyah. Wawancara ini dilakukan untuk mengumpulkan semua informasi tentang laporan *inventory* PT TOTAL PRINT, dan data untuk fitur pengambilan keputusan dalam memilih mesin cetak dan memaksimalkan layanan PT TOTAL PRINT.

## 3.5 Analisis dan Desain

Tahapan kelima dalam metode penelitian ini adalah analisis dan desain. Dalam tahapan penelitian ini berfokus kepada proses perancangan untuk menentukan hasil akhir dari pembangunan aplikasi sehingga perlu diperhatikan proses pembuatannya. Analisis yang tepat dan benar diperlukan agar hasil dapat digunakan dan diimplementasikan sesuai dengan kebutuhan sistem. Setelah semua data terkumpul dalam data *requirement*, perlu diadakannya analisis agar data tersebut dapat menggambarkan sistem berupa rancangan *interface, database* dan *flowchart* sistem aplikasi. Penerapan algoritma pada salah satu fungsi dalam aplikasi menjadi sebuah alur pengambilan keputusan yang akan menyelesaikan kasus permasalahan yang terdapat pada saat ini. Perancangan digambarkan dalam bentuk *flowchart* algoritma yang bekerja dalam sistem tersebut. Berikut ini adalah perancangan sistem dari aplikasi laporan *inventory* dengan fitur pengambilan keputusan dalam memilih mesin cetak dan layanan terbaik yang akan dibuat:

### 3.5.1 *Use Case* Diagram

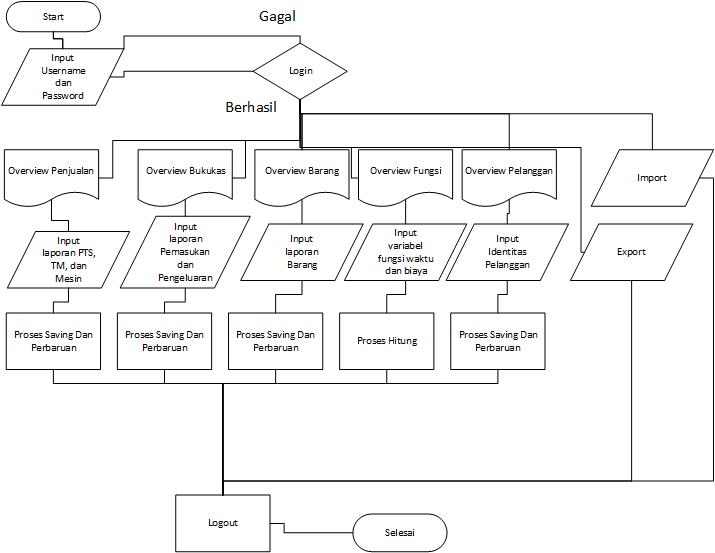
Berikut ini adalah perancangan dalam bentuk diagram *use case* yang akan digunakan oleh PT.TOTAL PRINT:



**Gambar 3.2 *Use Case* Sistem**

### 3.5.2 *Flow Chart* Sistem

Berikut ini adalah rancangan alur sistem dari tiap pengguna sistem yang dapat dilakukan oleh admin PT.TOTAL PRINT:

****

**Gambar 3.4 *Flowchart* Sistem**

### 3.5.3 *Pseudocode* Sistem

Berikut penjabaran alur sistem dengan *pseudocode* dari aplikasi laporan pengelolaan laporan *inventory* dengan fitur pengambilan keputusan:

1. Admin login

Declare Username, password

Input Username

Input Password

Fuction login()

Get username

Get password

If (username=username && password=password)

Then login-success

If (login-success)

Then switchpage()

End if

Else Login-failed

End if

1. Overview Barang

Declare m,d,y

Read m,d,y

Fuction bahan\_bakutotal()

GET bahan\_bakutotal

Print bahan\_bakutotal

1. Overview pelanggan

Declare m,d,y

Read m,d,y

Function pelanggan ()

Get pelanggan()

Print pelanggan

1. Overview bukukas

Declare m,d,y

Read m,d,y

Function Biaya\_inout()

Get bukukas

Print bukukas

1. Overview Penjualan

Declare m,d,y

Read m,d,y

Function Penjualan()

Get Penjualan()

Print Penjualan

1. Menggunakan pemutus keputusan waktu dan melihat hasilnya

Declare waktuperjalanan, tempattujuan1, tempattujuan2, tempattujuan3, tipe pekerjaan, waktudeadline

Input waktuperjalanan, tempattujuan1, tempattujuan2, tempattujuan3, tipe pekerjaan, waktudeadline

Function Hitung()

Get Hitung

Print Hitung

1. Menggunakan pemutus keputusan biaya dan melihat hasilnya

Declare batasbiaya, biayamesin1, biayamesin2, biayamesin3, performa mesin, pekerjaan

Input batasbiaya, biayamesin1, biayamesin2, biayamesin3, performa mesin, pekerjaan

Function Hitung()

Get Hitung

Print Hitung

1. Admin input barang masuk dan keluar

Declare id, nama, jenis, waktumasuk

Input id, nama, jenis, waktumasuk

Read timeslot

If (timeslot==1)

Then PRINT “Sudah Tersimpan”

Else Write barangmasuk[]

End if

Declare id, nama, jenis, waktukeluar

Input id, nama, jenis, waktukeluar

Read timeslot

If (timeslot==1)

Then PRINT “Sudah Tersimpan”

Else Write barangkeluar[]

End if

1. Admin input biayamasuk dan keluar

Declare m,d,y, id, nama, jenis, biayamasuk

Input m,d,y, id, nama, jenis, biayamasuk

Read timeslot

If (timeslot==1)

Then PRINT “Sudah Tersimpan”

Else Write inputbiayamasuk[]

End if

Declare m,d,y, id, uraian, biayakeluar

Input m,d,y, id, uraian,, biayakeluar

Read timeslot

If (timeslot==1)

Then PRINT “Sudah Tersimpan”

Else Write inputbiayakeluar[]

End if

1. Admin input penjualan layanan PTS

Declare m,d,y, invoicekluar, namapelanggan, Minimumbiayaprojek, includecopyprojek, biayakelanjutanprojek, tanggalmulaiprojek, awaljumlahproduk, tanggalselesaiprojek, akhirjumlahvolume, copyvolume, miscopy, materai,

Input m,d,y, invoicekluar, namapelanggan, Minimumbiayaprojek, includecopyprojek, biayakelanjutanprojek, tanggalmulaiprojek, awaljumlahproduk, tanggalselesaiprojek, akhirjumlahvolume, copyvolume, miscopy, materai,

Read timeslot

If (timeslot==1)

Then PRINT “Sudah Tersimpan”

Else Write layananPTS[]

End if

1. Admin input penjualan TM

Declare m,d,y, invoicekluar, namapelanggan, idbarang,SNbarang, Jumlahbarang, satuan, hargapersatuan,

Input m,d,y, invoicekluar, namapelanggan, idbarang,SNbarang, Jumlahbarang, satuan, hargapersatuan,

Read timeslot

If (timeslot==1)

Then PRINT “Sudah Tersimpan”

Else Write layananTM[]

End if

1. Admin input penjualan Mesin

Declare m,d,y, invoicekluar, namapelanggan, idbarang, SNbarang, Jumlahbarang, satuan, hargapersatuan, materai

Input m,d,y, invoicekluar, namapelanggan, idbarang,SNbarang, Jumlahbarang, satuan, hargapersatuan, materai

Read timeslot

If (timeslot==1)

Then PRINT “Sudah Tersimpan”

Else Write layananMesin[]

End if

1. Admin input daftar pelanggan

Declare nama, alamat, nomerhp

Input nama, alamat, nomerhp

Read timeslot

If (timeslot==1)

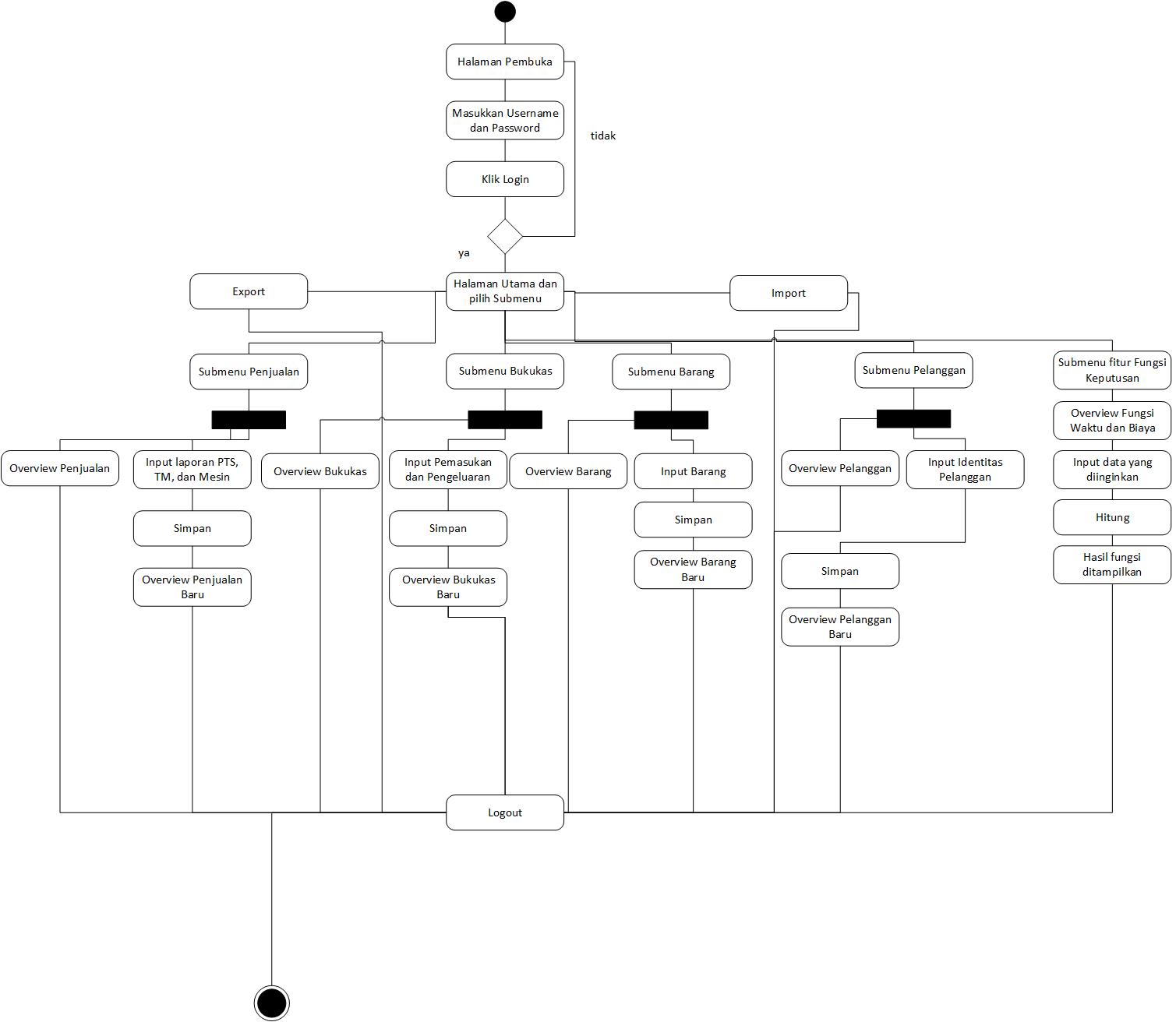
Then PRINT “Sudah Tersimpan”

Else Write daftarpelanggan[]

End if

### 3.5.4 *Activity* Diagram

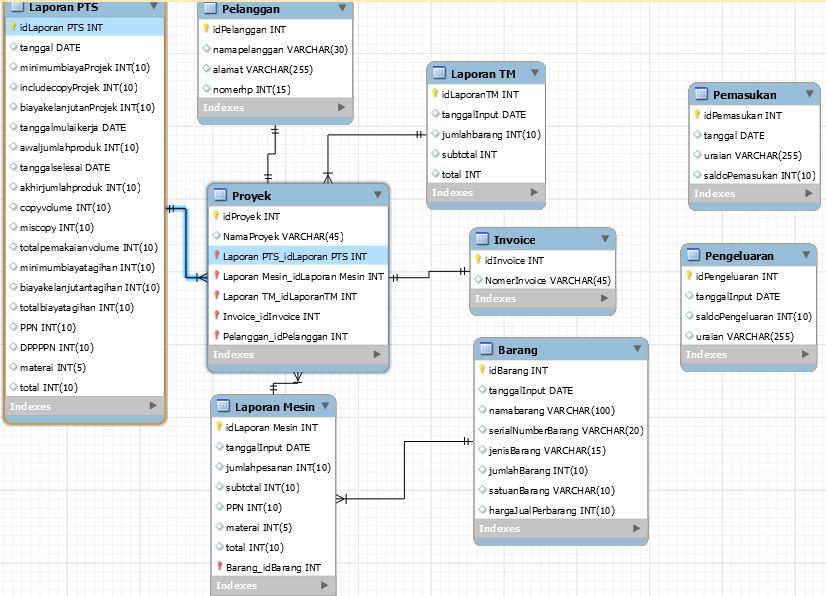
Diagram gambar di bawah ini merupakan penjelasan alur dari tiap pengguna yang terhubung, dikarenakan beberapa komponen dalam sistem saling menentukan proses *inventory* bekerja dengan benar.



**Gambar 3.5 *Activity Diagram***

### 3.5.5 Relasi Database

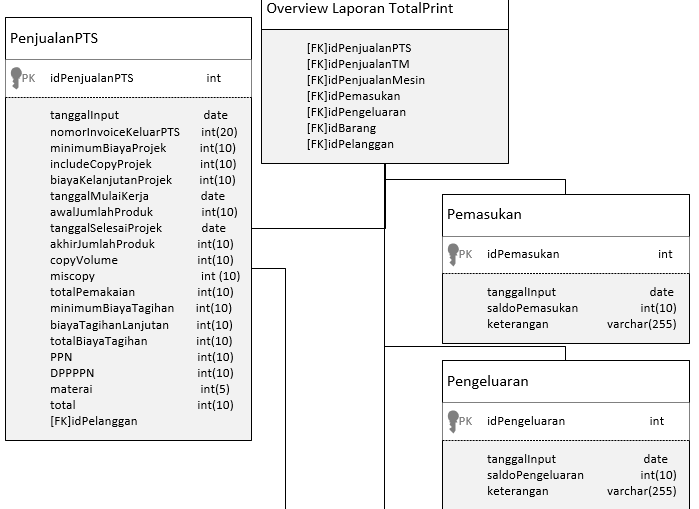
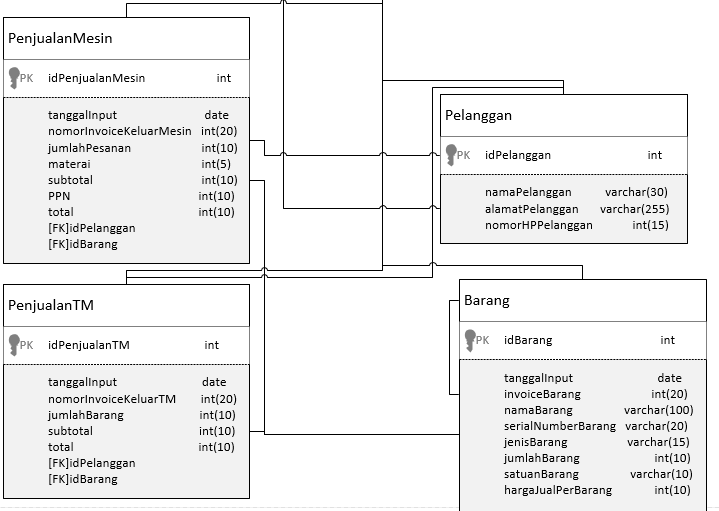
Berikut merupakan penjelasan alur relasi *database* dalam aplikasi laporan *inventory* dengan fitur pembuat keputusan.



**Gambar 3.6 Relasi *Diagram***

***3.5.6 Physical Database Design***

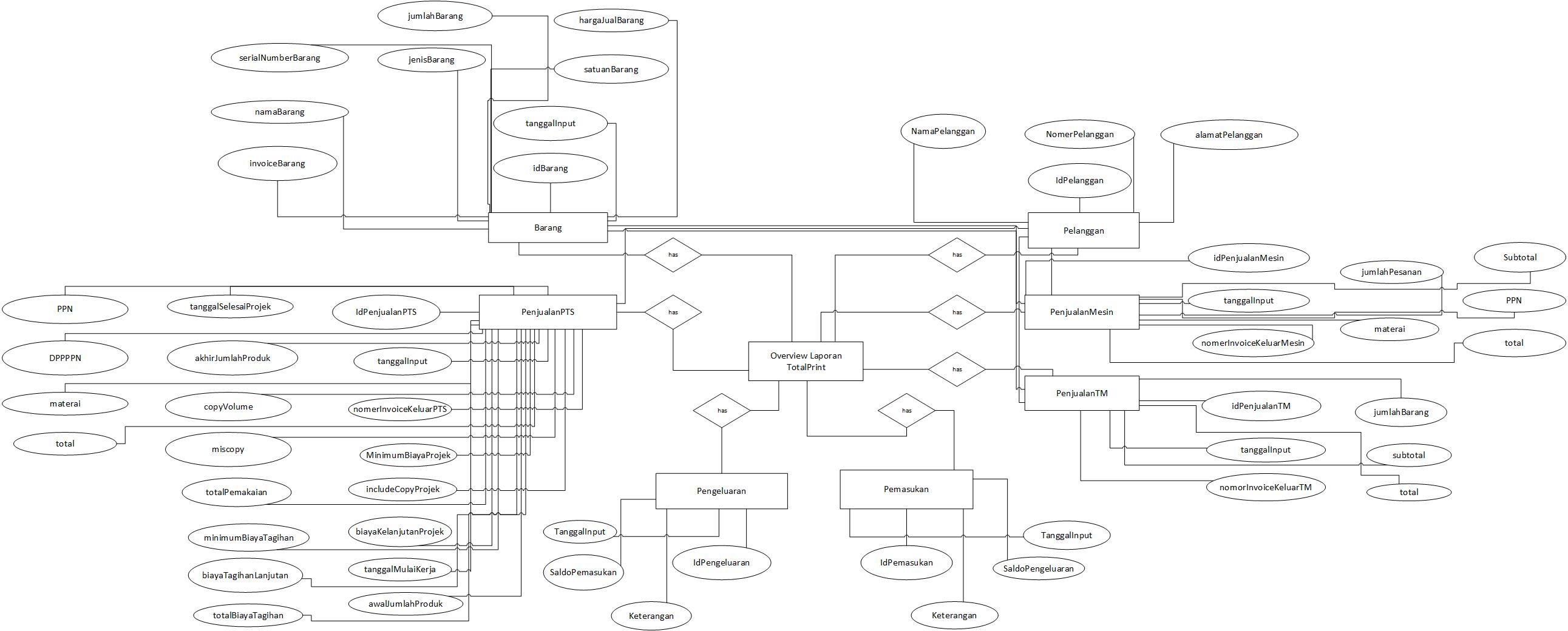
Berikut merupakan penjelasan alur UML diagram dalam aplikasi laporan *inventory* dengan fitur pembuat keputusan.



**Gambar 3.7 *Physical* Design**

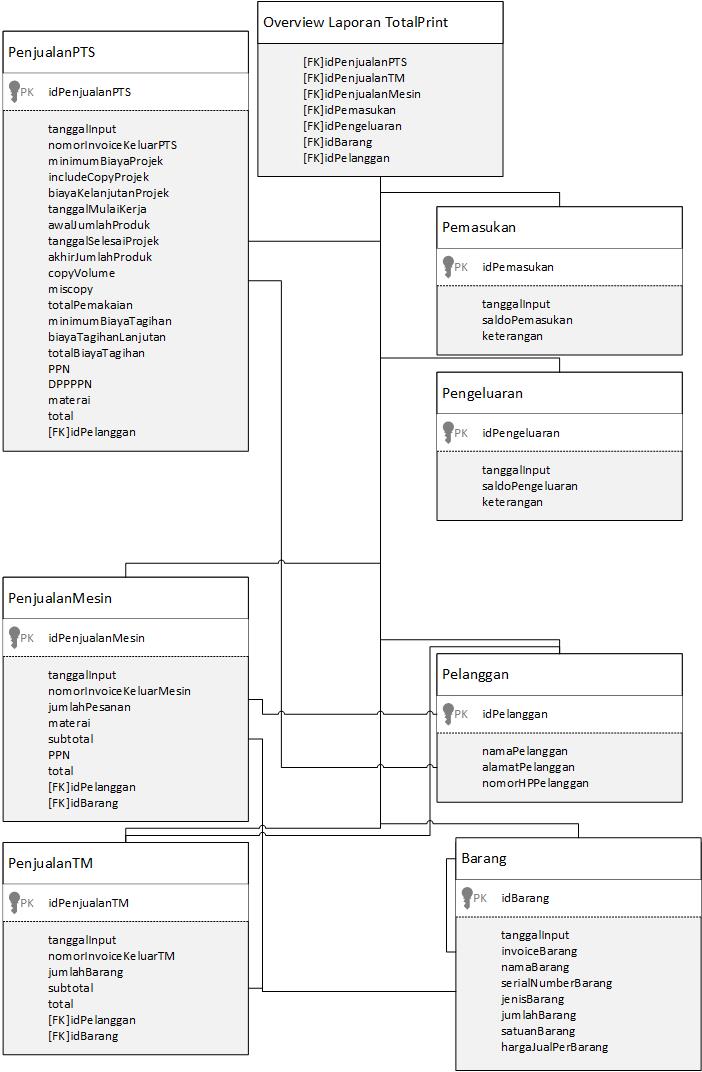
***3.5.7 Conceptual Database Design***

Berikut merupakan penjelasan alur *Conceptual Design* dalam aplikasi laporan *inventory* dengan fitur pembuat keputusan.



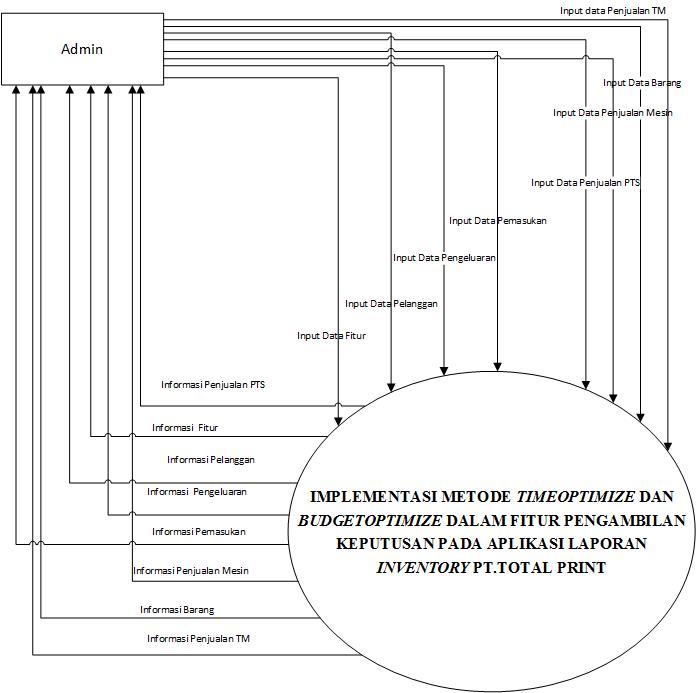
***3.5.8 Logical Design***

Berikut merupakan penjelasan alur *Logical Design* dalam aplikasi laporan *inventory* dengan fitur pembuat keputusan.

******

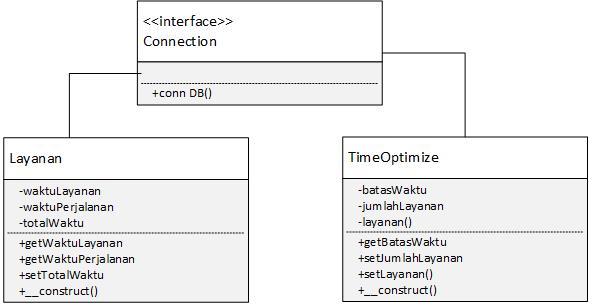
***3.5.9 DFD***

Berikut merupakan penjelasan alur *DFD* dalam aplikasi laporan *inventory* dengan fitur pembuat keputusan.



***3.5.10 Class Diagram***

Berikut merupakan penjelasan alur *class* diagram dalam aplikasi laporan *inventory* dengan fitur pembuat keputusan.



**Gambar 3.8 Class Diagram**

### 3.5.6 Penerapan Algoritma *Timeoptimized* dan *Budgetoptimized*

Aplikasi *inventory* dengan fitur pengambilan keputusan yang akan dibuat dan digunakan pada PT.TOTAL PRINT untuk pemilihan mesin cetak serta pengambilan keputusan dalam kegiatan operasional layanan menggunakan algoritma *Timeoptimized* dan *Budgetoptimized*. Algoritma tersebut bekerja pada saat fitur pengambilan keputusan diakses dan digunakan oleh admin untuk dapat menentukan mesin cetak yang tepat ataupun memilih keputusan dalam memberikan tugas terhadap teknisi ke beberapa tempat yang dapat terlaksana. Pihak administrasi sebagai admin adalah pemegang kewenangan sepenuhnya atas pengelolaan aplikasi dan fitur pengambilan keputusan. Dengan arti lain, aplikasi menyediakan laporan *inventory* yang jelas dan *user-friendly* sehingga dapat memberikan laporan yang rapi dan terdokumentasi, memberikan beberapa keputusan dalam pemilihan mesin cetak dengan memperhitungkan biaya sebagai pusatnya, dan memberikan sebuah keputusan berdasarkan fakta dan data agar layanan yang dapat dilakukan oleh teknisi lebih optimal. Maka digunakanlah algoritma *Timeoptimized* dan *Budgetoptimized* sebagai penyelesaian fitur pengambilan keputusan sebagai algoritma yang sesuai dengan kebutuhan masalah yang akan dipecahkan.

Algoritma *Timeoptimized* dan *Budgetoptimized* memberikan keputusan yang terbaik dengan memenuhi kondisi yang ditetapkan dengan memenuhi satu parameter yang dinginkan. Jika user ingin memutuskan sebuah keputusan yang berhubungan dengan parameter waktu, maka *Timeoptimized* akan memberikan beberapa hasil yang memenuhi kondisi user dengan mempertimbangkan parameter waktu sebagai permasalahan utama. Begitu juga dalam sebuah keputusan yang berhubungan dengan parameter biaya, maka *Budgetoptimized* akan memberikan beberapa hasil yang memenuhi kondisi pengguna dengan mempertimbangkan parameter biaya sebagai permasalahan utama.

Dalam aplikasi *inventory* PT.TOTAL PRINT dengan fitur pengambilan keputusan yang menggunakan algoritma *Timeoptimized* dan *Budgetoptimized*, tabel tempat pada satu hari dan barang pada satu pembelian di representasikan dalam tabel masing-masing. Representasi tempat dan biaya barang yang ditampilkan merupakan sebuah contoh simulasi kondisi tempat dan biaya bahan baku yang dibutuhkan dalam operasional PT.TOTAL PRINT. Gambaran waktu maksimal dan biaya maksimal digambarkan menggunakan kolom *max*.

Permintaan tempat yang dapat ditempuh dengan jenis pekerjaan yang dilayani oleh teknisi dikondisikan seperti berikut. Seorang kepala teknisi akan memberikan tugas terhadap teknisinya untuk memberikan layanan ke beberapa pelanggan. Waktu perjalanan bervariasi mulai dari 30 menit, 60 menit, 90 menit, dan 120 menit. Selain lama perjalanan, waktu pekerjaan dalam pelayanan juga bervariasi. Lama waktu pelayanan dikondisikan dengan beberapa jenis layanan yang akan diberikan seperti penggantian bahan baku 60 menit, pemasangan alat 30 menit, meneliti dan memperbaiki kerusakan 180 menit. Berikut representasi lama pekerjaan dan lama layanan dapat dilihat dalam tabel dibawah ini dengan hitungan menit.

**Tabel 3.1 Waktu Perjalanan Setiap Perusahaan (Menit)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tempat | A | B | C | D | E | F | G |
| MaxPerjalan | 30 Menit | 60 Menit | 120 Menit | 90 Menit | 60 Menit | 120 Menit | 60 Menit |

**Tabel 3.2 Waktu Pelayanan Setiap Layanan (Menit)**

|  |  |  |  |
| --- | --- | --- | --- |
| Layanan | Penggantian Bahan baku (PBA) | Pemasangan Alat (PA) | Penelitian dan Perbaikan Alat (PPA) |
| MaxLayanan | 60 Menit | 30 Menit | 120 Menit |

Setelah penjelasan tabel diatas, berikutnya dalam penelitian ini menjalankan fungsi seleksi yaitu fungsi *Timeoptimized*Tempat. Fungsi ini memberikan opsi dengan menyeleksi tempat berdasarkan waktu perjalanan dan layanan yang ditentukan. Pada kasus berikut, dalam penelitian ini mencontohkan kepala teknisi akan memberikan 3 jenis tugas kepada teknisinya. Penggantian bahan baku pada PT.A, Penelitian dan perbaikan alat pada PT.E dan pemasangan alat pada PT.D dengan sisa waktu kerja 180 menit.

**Tabel 3.3 Tabel Keputusan Hasil Fungsi *Timeoptimized*Tempat**

|  |  |
| --- | --- |
| Perusahaan dan Layanan | Waktu |
| A.PBA | 90 Menit |
| E.PA | 90 Menit |
| A.PBA – E.PA | 180 Menit |

Hasil tabel di atas merepresentasikan hasil dari fungsi *Timeoptimized*Tempat. Tabel tersebut menghasilkan sebuah himpunan tempat dengan waktu layanan dan waktu perjalanan yang sama atau kurang dari waktu yang ditetapkan. Setelah melalui tahapan tersebut, algoritma akan melakukan fungsi pemilihan ke fungsi selanjutnya yaitu fungsi *OptimizePlacebyObject*.

Dalam fungsi *OptimizePlacebyObject*, fungsi tersebut akan memberikan sebuah keputusan terbaik berdasarkan himpunan solusi yang telah dijalankan. Fungsi *OptimizePlacebyObject* akan memilih sebuah keputusan terbaik dengan mempertimbangkan waktu yang ditempuh juga banyaknya sebuah pekerjaan yang dapat dilakukan dalam sebuah tenggat waktu yang ditetapkan.

**Tabel 3.4 Tabel Hasil Keputusan Fungsi *OptimizePlacebyObject***

|  |  |
| --- | --- |
| Perusahaan dan Layanan | Waktu |
| A.PBA – E.PA | 180 Menit |

Selain itu juga terdapat permintaan biaya bahan baku yang dapat memenuhi kondisi yang ditetapkan oleh pengguna dengan contoh kasus seperti berikut. Seorang kepala keuangan akan membeli bahan baku yang cocok untuk memenuhi jenis layanan yang diberikan. Biaya alat yang berbeda-beda seperti printer Xerox 2,4 juta, printer HP 1,5 juta, dan printer Canon 3,1 juta. Selain biaya yang berbeda-beda, alat-alat tersebut juga mempunyai performa yang berbeda-beda. Misalnya printer Xerox dapat memprint 2000 *copy*, printer HP 1000 *copy* dan printer Canon 3000 *copy*. Untuk lebih jelasnya, dapat dilihat dlam tabel dibawah ini.

**Tabel 3.5 Tabel *Brand* Printer dan Harga**

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Printer | Xerox | HP | Canon |
| HargaPerjuta | 2,4 Juta | 1,5 Juta | 3,1 Juta |

**Tabel 3.6 Tabel *Brand* Printer dan Performa**

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Printer | Xerox | HP | Canon |
| Performa*Copy* | 2000 *Copy* | 1000 *Copy* | 3000 *Copy* |

Berdasarkan penjelasan tabel sebelumnya, penelitian ini memasukkan semua tabel tersebut dalam fungsi *Budgetoptimized*Bahan dengan kondisi operasional yang dibutuhkan. Contohnya, ketika seorang kepala operasional membutuhkan sebuah mesin *copy* dengan kebutuhan 12000 *copy* dengan alokasi biaya 15 juta.

**Tabel 3.7 Tabel Hasil Fungsi *Budgetoptimized*Bahan**

|  |  |
| --- | --- |
| Alat | Biaya |
| Xerox | 14,4 Juta |
| Canon | 12,4 Juta |

Lalu dengan adanya hasil dari fungsi *Budgetoptimized*Alat yang memberikan sebuah hasil himpunan biaya yang sama atau lebih kecil dari biaya yang dialokasikan untuk menjadi alat yang mampu mengatasi kondisi yang diberikan. Setelah melalui tahapan berikut, maka penelitian akan masuk ke fungsi selanjutnya yaitu fungsi *BudgetoptimizedByObject* yang akan memberikan sebuah keputusan terbaik dalam himpunan solusi yang diberikan seperti tabel di bawah ini.

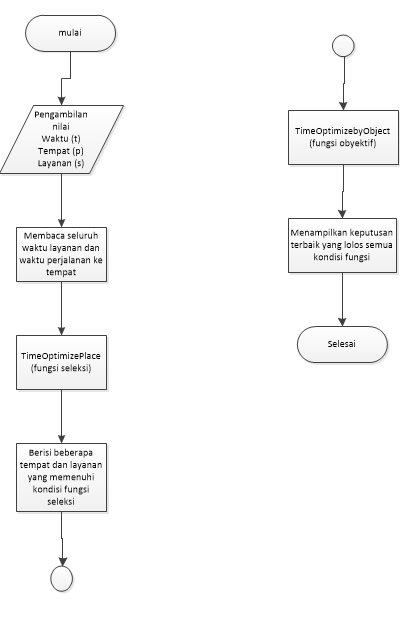
**Tabel 3.8 Tabel Hasil Fungsi *BudgetoptimizedByObject***

|  |  |
| --- | --- |
| Alat | Biaya |
| Canon | 12,4 Juta |

Pada tabel diatas menghasilkan sebuah keputusan terbaik yang memenuhi kondisi pengguna. Fungsi *BudgetoptimizedByObject* menghasilkan sebuah keputusan untuk membeli sebuah alat *copy* canon yang memiliki harga tertinggi namun juga performa tertinggi. Dengan tingginya performa maka penalty yang akan disebabkan alat tersebut juga akan berkurang dan mengurangi biaya berkelanjutan dan biaya keseluruhan.

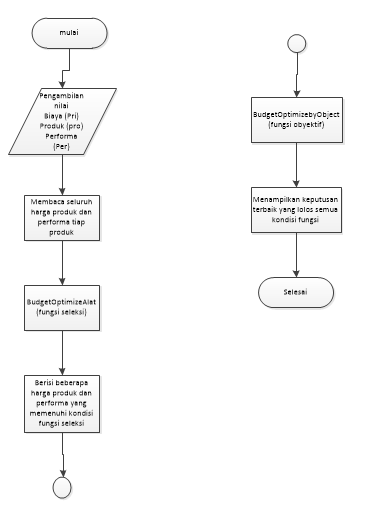
Sesuai wawancara dengan Bapak Dede Murariramsyah, selaku kepala bidang administrasi PT.TOTAL PRINT untuk memberikan sebuah sistem basis data laporan *inventory* yang lebih mendetil dan memiliki fitur untuk mengatasi masalah nyata dalam dunia kerja, diterapkan algoritma *Timeoptimized* untuk menyelesaikan permasalahan dalam masalah layanan teknisi untuk pengambilan keputusan yang berhubungan dengan waktu dan algoritma *Budgetoptimized* untuk menyelesaikan masalah biaya dalam pemilihan mesin cetak untuk meminimalisir biaya pengeluaran agar laba lebih optimal. Namun jika pada penerapan algoritma ini terdapat kondisi hal-hal yang tidak terprediksi namun mempunyai dampak dalam biaya serta waktu, maka dibutuhkan kebijakan dari admin PT.TOTAL PRINT untuk membuat sebuah keputusan sebagai solusi untuk mengatasi permasalahan yang dihadapi dengan beberapa pertimbangan kondisi yang terjadi pada saat tersebut.

### 3.5.7 *Flowchart* Algoritma *Timeoptimized*



**Gambar 3.9 *FlowChart* Algoritma *Timeoptimized***

### 3.5.8 *Flowchart* Algoritma *Budgetoptimized*



**Gambar 3.10 *FlowChart* Algoritma *Budgetoptimized***

### 3.5.9 *Pseudocode* Algoritma *Timeoptimized*

Berikut ini *pseudocode* dari algoritma *Timeoptimized* ketika berjalan pada fitur pengambilan keputusan didalam aplikasi *inventory:*

Function Timeoptimized()

Declare place, time, service

Get place, time, service

Read database

TimeoptimizedTempat()

S={NULL}

(i=0; i<n, i++) {

WaktuPerjalanan = place / 22

WaktuPenyelesaian = layanan + waktuPengerjaan

If WaktuPenyelesaian<=deadline

Then S = TO //TO adalah hasil fungsi

End if

If ∑TO << waktupenyelesaian

Function Combination TO(array[TO]<= ∑TO;TO++)

Combination set (range(0, ∑TO))

Assert Combination TO(x ,y)

[TO0, TO1,TO2]

[TO0,TO1,TO3]

[TO0,TO1,TO4]

[TO0,TO1,TOn]

Display CombinationResult ()

}

TimeoptimizedByObject()

If CombinationResult == 1

Sort CombinationResult by waktuPenyelesaian

Print CombinationResult

Else

Sort S by WaktuPenyelesaian

Select TO1 from s;

Print TO1;

End if

### 3.5.10 *Pseudecode* Algoritma *Budgetoptimized*

Function Budgetoptimized()

Declare Biaya, harga, performa, pekerjaan

Get Biaya, performa, pekerjaan

Read database

BudgetoptimizedBahan()

S={NULL}

(i=0; i<n, i++) {

Penalty = performa/pekerjaan

BiayaPenyelesaian = harga \* Penalty

If Penyelesaian<=deadline

Then S = BO //BO adalah hasil fungsi

End if

If ∑BO << Biayapenyelesaian

Function Combination BO(array[BO]<= ∑BO;BO++)

Combination set (range(0, ∑BO))

Assert Combination BO(x ,y)

[BO0, BO1,BO2]

[BO0,BO1,BO3]

[BO0,BO1,BO4]

[BO0,BO1,BOn]

Display CombinationResult ()

}

BudgetoptimizedByObject()

If CombinationResult == 1

Sort CombinationResult by BiayaPenyelesaian

Print CombinationResult

Else

Sort S by BiayaPenyelesaian

Select TO1 from S;

Print TO1;

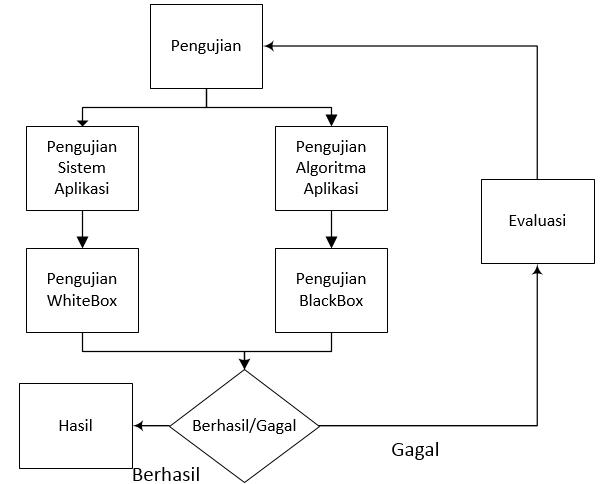
End if

## 3.6 Implementasi

Langkah berikutnya setelah melakukan analisis and desain adalah dengan melakukan implementasi. Implementasi merupakan sebuah proses desain yang telah dilakukan sebelumnya menjadi bahasa pemrograman yang secara teknis dilakukan oleh seorang programmer. Implementasi pada penelitian ini akan dilakukan dengan menggunakan bahasa pemrograman PHP dan pemrograman berorientasi objek OOP. Tahapan ini akan menghasilkan sebuah aplikasi *inventory* dengan fitur pengambilan keputusan dengan fungsi seluruh sistem yang sudah berjalan dengan baik.

## 3.7 Pengujian

Tahapan ini merupakan hal terpenting dalam sebuah penelitian. Hal tersebut disebabkan karena pada tahapan ini akan bertujuan untuk menemukan kesalahan dan kekurangan pada hasil aplikasi yang diuji. Tahapan ini bertujuan agar aplikasi yang dibuat dalam penelitian memenuhi kondisi dan kriteria yang sesuai dengan tujuan perancangan aplikasi. Pengetesan dilakukan agar menghasilkan sebuah aplikasi pengelolaan *inventory* dengan fitur pengambilan keputusan pada PT.TOTAL PRINT memiliki fungsi yang sesuai dengan perencanaan sebelumnya agar aplikasi ini dapat dioperasikan dengan baik dan benar.



**Gambar 3.11 Gambaran Umum Pengujian Aplikasi**

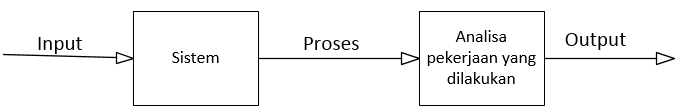
Penelitian ini melakukan pengujian dengan membagi menjadi 2 bentuk pengujian. Pengujian pertama dilakukan untuk menguji sistem aplikasi yang dibuat selama penelitian. Pengujian pertama menggunakan metode *WhiteBox* untuk menguji cara kerja aplikasi secara rinci sesuai dengan spesifikasinya (Rouf, 2011). Bentuk teknik pengujian yang akan digunakan dalam metode *WhiteBox* adalah teknik *Basis Path.* Sedangkan pengujian kedua dilakukan untuk menguji dan mengetes algoritma yang digunakan dalam aplikasi yang dibuat. Cara pengujian dan pengetesan yang dilakukan menggunakan metode pengetesan *BlackBox.* Pengujian metode *BlackBox* merupakan sebuah pengujian untuk mengetahui apakah semua fungsi perangkat lunak berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Rouf, 2011). Bentuk teknik pengujian metode *BlackBox* yang akan digunakan adalah tipe teknik *Equivalence Class Testing.* Untuk keterangan lebih lanjut tentang pengujian akan dijelaskan pada sub bab berikut ini.

### 3.7.1 Pengujian *WhiteBox*

Metode pengujian *WhiteBox* merupakan metode pengujian dengan menggunakan struktur kontrol program untuk memperoleh kasus uji (Rouf, 2011). Terdapat beberapa keuntungan yang akan didapatkan dengan menggunakan metode *WhiteBox* dalam kasus uji yang akan dilakukan. Beberapa keuntungan tersebut adalah:

1. Semua keputusan lojikal teruji.
2. Seluruh Loop yang sesuai dengan batasan dan kondisinya teruji.
3. Seluruh Struktur data internal yang menjamin validitas teruji.
4. Seluruh jalur independen didalam aplikasi tereksekusi sekurang-kurangnya sekali terjamin.

Selain mendapatkan beberapa keuntungan dalam pengujian metode *WhiteBox,* berikut ini merupakan gambarang umum bagaimana proses sistem kerja dari metode pengujian *WhiteBox.*



**Gambar 3.12 Gambaran Proses Pengujian *WhiteBox***

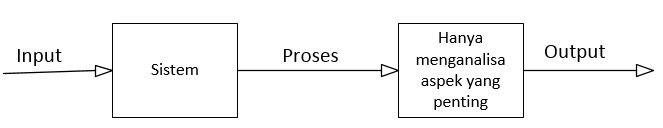
Dalam penggunaan metode pengujian *WhiteBox*, terdapat teknik yang akan digunakan sebagai panduan pengujian metode. Teknik yang akan digunakan adalah teknik *Basis Path*. Teknik *Basis Path* digunakan untuk mendapatkan kompleksitas lojik dari suatu prosedur dan menggunakan ukuran ini sebagai petunjuk dengan mendefinisikan himpunan jalur yang akan diuji.

### 3.7.2 Pengujian *BlackBox*

Metode pengujian *BlackBox* merupakan sebuah pengujian dengan memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program aplikasi (Rouf, 2011). Dalam pengujian yang dilakukan dalam penelitian ini, bebrapa keuntungan yang akan didapatkan jika menggunakan metode pengujian *BlackBox* adalah:

1. Menemukan fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan interface ditemukan.
3. Kesalahan dalam akses basis data eksternal atau struktur data teridentifikasi.
4. Kesalahan pada terminasi dan inisialisasi.
5. Fungsional tervalidasi.
6. Mengetahui kesensitifan sistem terhadap nilai input tertentu.
7. Mengetahui batasan dari suatu data.

Selain mendapatkan beberapa keuntungan dalam penggunaan metode *BlackBox*, berikut ini merupakan gambaran umum proses pengujian metode *BlackBox.*



**Gambar 3.13 Gambaran Umum Proses Pengujian *BlackBox***

Dalam pengujian metode *BlackBox*, terdapat beberapa teknik yang dimiliki oleh pengujian metode *BlackBox.* Dari beberapa teknik pengujian tersebut, penelitian ini akan menggunakan teknik pengujian *Equivalence Class Testing.* Penelitian ini menggunakan teknik tersebut karena tujuan serta batasan yang ada dalam teknik tersebut sejalan dengan tujuan dan batasan yang terdapat dalam aplikasi yang dibuat. Terdapat beberapa kondisi untuk melakukan pengujian teknik *Equivalence Class Testing.* Beberapa kondisi tersebut adalah:

1. Bagi domain input menjadi beberapa kelas unutk dijadikan sebagai kasus uji.
2. Kelas yang telah terbentuk disajikan sebagai kondisi input dalam kasus uji.
3. Terdapat nilai valid dan invalid dalam himpunan nilai-nilai kelas tersebut.

## 3.8 Hasil dan Laporan

Tahapan dari hasil percobaan berupa sebuah aplikasi laporan *inventory* yang telah teruji berdasarkan pengujian sistem dan pengujian algoritma. Setelah melalui tahapan *testing*, tahapan selanjutnya adalah tahapan dokumentasi dalam bentuk penulisan laporan tugas akhir dari hasil penelitian yang disusun dari awal hingga akhir penelitian. Penulisan yang dilakukan mengikuti aturan yang telah dipublikasikan oleh pihak Universitas Bakrie sebagai pedoman dalam penyusunan tugas akhir.

**BAB IV**

**PEMBAHASAN**

Perancangan dan pembangunan aplikasi laporan inventory dengan menggunakan metode Timeoptimized dan Budgetoptimized berbasis web untuk PT.TOTALPRINT akan dibahas dalam bab ini. Metode Waterfall digunakan sebagai metode pengembangan aplikasi dalam penelitian ini.

* 1. Software Requirement Analysis

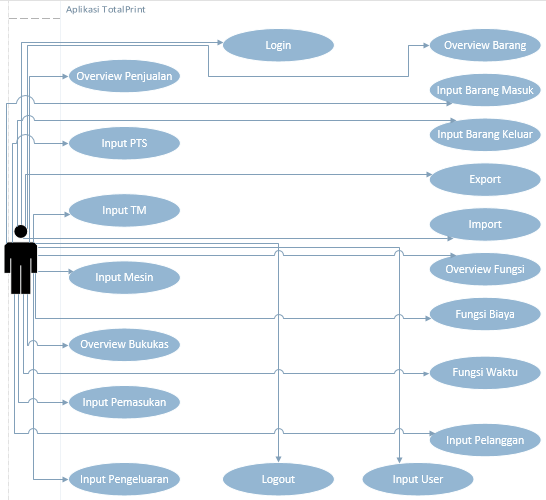
Pada penelitian ini memiliki aktivitas seluruh pelayanan dalam menjalankan perintah fungsi-fungsi dalam sistem berdasarkan hasil pengumpulan data yang telah dilakukan melalui observasi dan wawancara pada pengelola sistem dan pihak terkait. Wawancara dilakukan dengan Bapak Dede Muhariramsyah sebagai kepala administrasi PT TOTALPRINT dengan mendefinisikan rancangan sistem beserta fungsi-fungsi dan kriteria yang diinginkan oleh pihak pengguna.

4.2 Design

Dalam melakukan pengembangan aplikasi, perlu dilakukan sebuah perancangan aplikasi agar perancangan tersebut dapat menjadi acuan dalam tahap selanjutnya yaitu *implementation*. Tahap ini akan menjelaskan rancangan aplikasi dalam model UML seperti use case diagram, activity diagram, sequence diagram, class diagram, data model, dan pseudocode.

* + 1. Use Case Diagram

Use case diagram menggambarkan alur interaksi user dalam penggunaan aplikasi dan menampilkan hubungan beberapa use case. Diagram use case dalam aplikasi pengelolaan laporan *inventory* dapat kita lihat pada Gambar 4.1.



Gambar 4.1 Use Case Diagram Aplikasi Pengelolaan Inventory

Deskripsi use case diagram pada Gambar 4.1 akan dijelaskan lebih detail dalam table use case scenario berikut:

Login

|  |  |  |
| --- | --- | --- |
| Use case name | Login | |
| Use case ID | 1 | |
| Actor | Admin | |
| Description | Use case ini menggambarkan proses admin dalam melakukan login. | |
| Pre-condition | Admin mengisi username dan password | |
| Trigger | Use case ini dilakukan agar admin dapat masuk ke dalam aplikasi | |
| Typical of events | Actor Action | Sistem Response |
|  | 1. Membuka aplikasi |  |
|  | 1. mengisi user name dan password | 1. server mengecek username dan password yang telah disi, |
|  |  | 1. Menampilkan aplikasi dengan membuka halaman awal |
| Alternate Course | Jika gagal, akan kembali ke use case login dan tidak menampilkan halaman utama aplikasi | |
| Post-Condition | Aplikasi menampilkan halaman utama jika use case login berhasil | |

Overview Penjualan

|  |  |  |
| --- | --- | --- |
| Use case name | Overview Penjualan | |
| Use case ID | 2 | |
| Actor | Admin | |
| Description | Use case ini menggambarkan kegiatan melihat tabel laporan penjualan pada halaman utama | |
| Pre-condition | Admin sukses login | |
| Trigger | Use case ini dilakukan agar admin dapat melihat laporan penjualan yang telah diinput sebelumnya | |
| Typical of events | Actor Action | Sistem Response |
|  | 1. Membuka aplikasi | 1. Menampilkan halaman Login |
|  | 1. Login |  |
|  | 1. Mengisi username dan password | 1. server mengecek username dan password yang telah disi |
|  |  | 1. Menampilkan halaman overview penjualan |
| Alternate Course | Tidak ada scenario error input pada tahap ini | |
| Post-Condition | Aplikasi menampilkan halaman overview penjualan yang berisi tabel penjualan PTS, penjualan TM, dan penjualan mesin | |

Input PTS

|  |  |  |
| --- | --- | --- |
| Use case name | Input PTS | |
| Use case ID | 3 | |
| Actor | Admin | |
| Description | Use case ini menggambarkan kegiatan input laporan penjualan PTS | |
| Pre-condition | Admin sukses login pada aplikasi | |
| Trigger | Use case ini dilakukan agar admin dapat mengisi laporan penjualan PTS dan memperbaharui laporan PTS yang ada | |
| Typical of events | Actor Action | Sistem Response |
|  | 1. Membuka aplikasi | 2. Menampilkan halaman Login |
|  | 3. Login |  |
|  | 4. Mengisi username dan password | 5. server mengecek username dan password yang telah disi |
|  |  | 6. Menampilkan halaman overview penjualan |
|  | 7. Admin mengklik input laporan PTS pada navigasi penjualan | 8. Menampilkan halaman untuk menginput laporan PTS |
| Alternate Course | Tidak ada scenario error input pada tahap ini | |
| Post-Condition | Aplikasi menampilkan halaman input laporan PTS dan tabel laporan PTS yang telah ada sebelumnya | |

Input TM

|  |  |  |
| --- | --- | --- |
| Use case name | Input TM | |
| Use case ID | 3 | |
| Actor | Admin | |
| Description | Use case ini menggambarkan kegiatan input laporan penjualan TM | |
| Pre-condition | Admin sukses login pada aplikasi | |
| Trigger | Use case ini dilakukan agar admin dapat mengisi laporan penjualan TM dan memperbaharui laporan TM yang ada | |
| Typical of events | Actor Action | Sistem Response |
|  | 1. Membuka aplikasi | 2. Menampilkan halaman Login |
|  | 3. Login |  |
|  | 4. Mengisi username dan password | 5. server mengecek username dan password yang telah disi |
|  |  | 6. Menampilkan halaman overview penjualan |
|  | 7. Admin mengklik input laporan TM pada navigasi penjualan | 8. Menampilkan halaman untuk menginput laporan TM |
| Alternate Course | Tidak ada scenario error input pada tahap ini | |
| Post-Condition | Aplikasi menampilkan halaman input laporan TM dan tabel laporan TM yang telah ada sebelumnya | |

Input Mesin

|  |  |  |
| --- | --- | --- |
| Use case name | Input Mesin | |
| Use case ID | 4 | |
| Actor | Admin | |
| Description | Use case ini menggambarkan kegiatan input laporan penjualan Mesin | |
| Pre-condition | Admin sukses login pada aplikasi | |
| Trigger | Use case ini dilakukan agar admin dapat mengisi laporan penjualan Mesin dan memperbaharui laporan Mesin yang ada | |
| Typical of events | Actor Action | Sistem Response |
|  | 1. Membuka aplikasi | 2. Menampilkan halaman Login |
|  | 3. Login |  |
|  | 4. Mengisi username dan password | 5. server mengecek username dan password yang telah disi |
|  |  | 6. Menampilkan halaman overview penjualan |
|  | 7. Admin mengklik input laporan Mesin pada navigasi penjualan | 8. Menampilkan halaman untuk menginput laporan Mesin |
| Alternate Course | Tidak ada scenario error input pada tahap ini | |
| Post-Condition | Aplikasi menampilkan halaman input laporan Mesin dan tabel laporan Mesin yang telah ada sebelumnya | |

Overview Bukukas

|  |  |  |
| --- | --- | --- |
| Use case name | Overview Bukukas | |
| Use case ID | 5 | |
| Actor | Admin | |
| Description | Use case ini menggambarkan kegiatan melihat tabel laporan bukukas | |
| Pre-condition | Admin sukses login | |
| Trigger | Use case ini dilakukan agar admin dapat melihat laporan penjualan yang telah diinput sebelumnya | |
| Typical of events | Actor Action | Sistem Response |
|  | 1. Membuka aplikasi | 1. Menampilkan halaman Login |
|  | 1. Login |  |
|  | 1. Mengisi username dan password | 1. server mengecek username dan password yang telah disi |
|  |  | 1. Menampilkan halaman overview penjualan |
|  | 7. mengklik navbar samping bukukas | 8. Menampilkan halaman overview bukukas |
| Alternate Course | Tidak ada scenario error input pada tahap ini | |
| Post-Condition | Aplikasi menampilkan halaman overview bukukas yang berisi tabel pemasukan dan tabel pengeluaran | |

Input Pemasukan

|  |  |  |
| --- | --- | --- |
| Use case name | Input Pemasukan | |
| Use case ID | 6 | |
| Actor | Admin | |
| Description | Use case ini menggambarkan kegiatan melihat tabel laporan bukukas | |
| Pre-condition | Admin sukses login | |
| Trigger | Use case ini dilakukan agar admin dapat melihat laporan penjualan yang telah diinput sebelumnya | |
| Typical of events | Actor Action | Sistem Response |
|  | 1. Membuka aplikasi | 1. Menampilkan halaman Login |
|  | 1. Login |  |
|  | 1. Mengisi username dan password | 1. server mengecek username dan password yang telah disi |
|  |  | 1. Menampilkan halaman overview penjualan |
|  | 7. mengklik navbar samping bukukas | 8. Menampilkan halaman overview bukukas |
| Alternate Course | Tidak ada scenario error input pada tahap ini | |
| Post-Condition | Aplikasi menampilkan halaman overview bukukas yang berisi tabel pemasukan dan tabel pengeluaran | |

# **DAFTAR PUSTAKA**

Bassil, Y. (2012). A Simulation For The Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (iJET), ISSN: 2049-3444, Vol. 2, No. 5*.

Kindi, M. A., Lawati, A. M., & Rawahi, A. H. (2015). The Challenge of Normative Decision Making in Industries. *International Conference on Industrial Engineering and Operations Management*.

Martani, D., Veronica, S., Wardhani, R., Farahmita, A., & Tanujaya, E. (2012). *Akuntansi Keuangan Menengah Berbasis PSAK.* Jakarta: Salemba Empat.

Pearson. (2015). *Web Application Basics*. Retrieved March 21, 2016, from www.pearsonhighered.com[online]: www.pearsonhighered.com/samplechapter/0201730383. Pdf

Peter H. Carstensen, L. V. (2001). DESIGN OF WEB-BASED INFORMATION SYSTEM NEW CHALLENGES FOR SYSTEMS DEVELOPMENT. *Global Co-Operation in the New Millenium*, 536-544.

Pressman, R. (2001). Software Engineering A Practitioner’s Approach. *New York: McGraw-Hill*.

Raddy, S. R. (2006). *Market Economy Based Resource Allocation In Grids.* Indian Institute of Technology, Information Technology. India: School of Information Technology.

Rouf, A. (2011). *Pengujian Perangkat Lunak Dengan Menggunakan Metode Whitebox Dan Blackbox.* Sistem Informasi. Semarang: Skripsi STMIK HIMSYA.

Suyodti, A. W. (2014). *Rancang Bangun Sistem Informasi Monitoring Penyewaan Gedung dan Infrastruktur Teknologi Informasi (Studi Kasus: PT INDOSAT, TBK dan Mitra).* Fakultas Sains dan Teknologi, Program Studi Sistem Informasi. Jakarta: Skripsi Universitas Islam Negeri Syarif Hidayatullah.