

BAB III

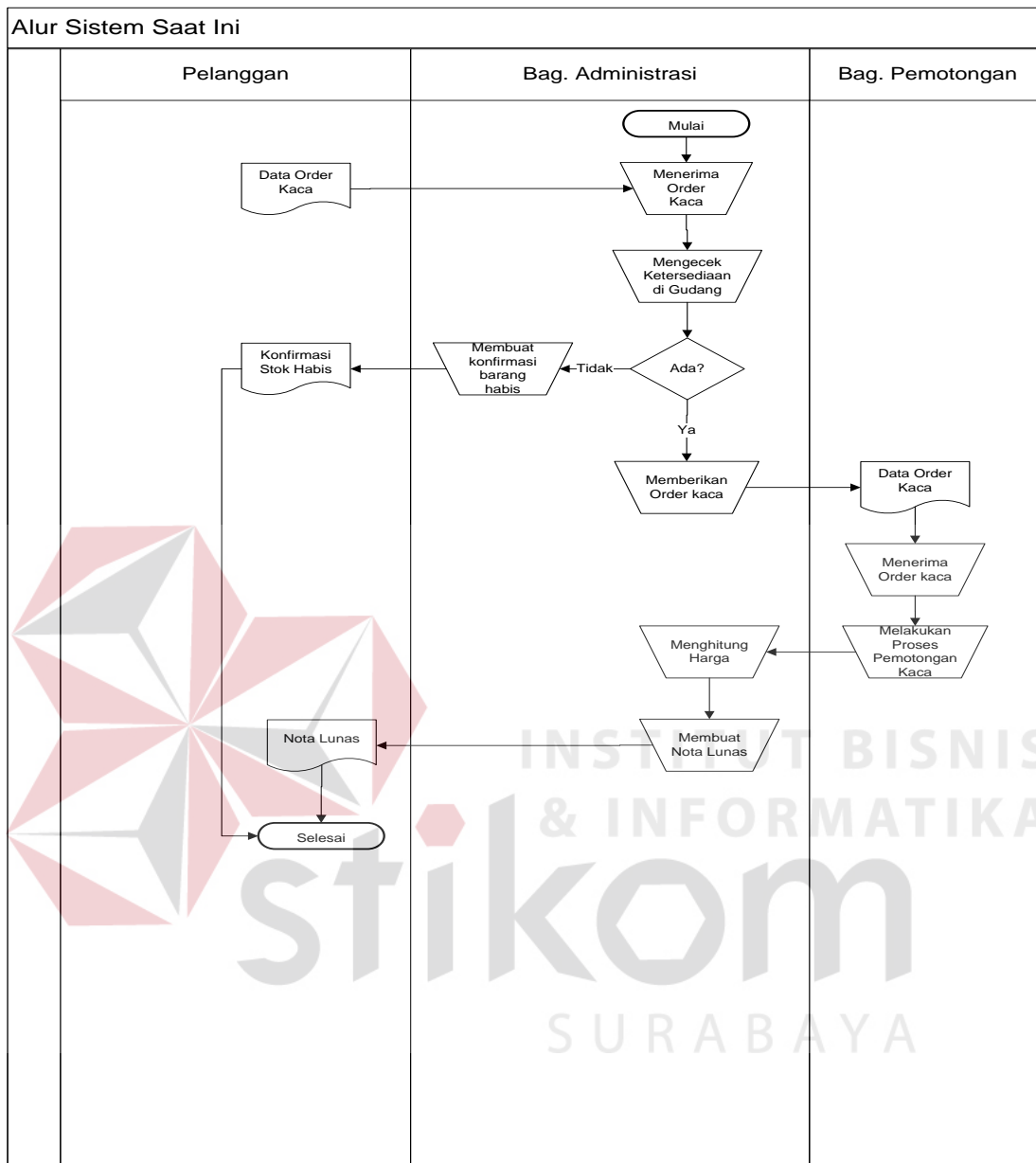
ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dibahas tentang analisis permasalahan, solusi permasalahan dan perancangan sistem dalam Rancang Bangun Aplikasi *Cutting Stock Optimization* Dengan Metode *Integer Linear Programming* Pada Toko Kaca Rejeki. Analisis dan hasil dari analisis tersebut juga akan dibahas lebih detil di bab ini.

3.1 Analisis Permasalahan

Analisis digunakan untuk mendefinisikan suatu permasalahan dan bagaimana cara mengatasi permasalahan tersebut. Pada langkah analisis dilakukan tahapan-tahapan untuk mengetahui permasalahan yang ada yaitu dengan observasi, sehingga dapat dilakukan suatu tindakan untuk menghasilkan solusi yang tepat untuk menyelesaikan permasalahan tersebut.

Dari hasil pengumpulan data yang dilakukan, diperoleh beberapa pengguna yang secara langsung berinteraksi dengan sistem yang sudah ada saat ini, yaitu Bagian Administrasi dan Bagian Pemotongan. Secara garis besar proses bisnis pada Toko Kaca Rejeki ini dimulai dari menerima order dan melakukan proses pemotongan. Adapun proses secara keseluruhan untuk kondisi saat ini dapat dilihat pada Gambar 3.1.



Gambar 3.1. Alir Sistem (*Current System*)

Gambar 3.1 merupakan alir sistem yang saat ini sudah ada. Adapun penjelasan Alir Sistem tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1. Penjelasan Alir Sistem (*Current System*)

Proses	Nama Proses	Kegiatan	Aktor
1	Menerima Order Kaca	Bagian administrasi menerima daftar order kaca dari pelanggan	Administrasi
2	Mengecek Ketersediaan Gudang	Mengecek ketersediaan stok kaca dengan kaca yang ada di order kaca pelanggan	Administrasi
3	Memberikan Order Kaca	Memberikan order kaca yang selesai dicek kepada bagian pemotongan	Administrasi
4	Menerima Order Kaca	Menerima order kaca dari bagian administrasi	Bagian Pemotongan
5	Melakukan Proses Pemotongan	Melakukan proses pemotongan kaca sesuai dengan order kaca pelanggan	Bagian Pemotongan
6	Menghitung Harga	Melakukan perhitungan harga dari order pelanggan	Bagian Administrasi
7	Membuat Nota Lunas	Membuat nota lunas pelanggan	Bagian Administrasi

Proses-proses yang telah dibahas sebelumnya merupakan proses yang dilakukan saat ini, dimana pada proses-proses tersebut akan dilakukan analisis untuk mengetahui kebutuhan dari setiap pengguna yang ada dan mengetahui proses-proses yang harus dielitisasi, proses yang diintegrasikan menjadi satu fungsi atau membangun fungsi baru, dengan maksud agar fungsi yang akan dibangun sesuai dengan kebutuhan pengguna.

3.2 Permasalahan

Setelah diketahui detil proses-proses yang dilakukan oleh tiap pengguna, penulis melakukan analisis kebutuhan yang sesuai dengan proses-proses tersebut. Analisis kebutuhan tersebut diperlukan untuk merancang perangkat lunak yang memiliki fungsi-fungsi yang sesuai dengan kebutuhan pengguna. Analisis kebutuhan dilakukan pada setiap pengguna yang secara langsung melakukan interaksi dengan sistem. Adapun analisis kebutuhan untuk pengguna-pengguna tersebut adalah:

1. Analisis pada Alir Sistem Bagian Administrasi

Bagian administrasi memiliki peran sebagai penerima order pelanggan dan mencatatnya. Setelah order dicatat akan dilakukan pengecekan ketersediaan kaca dengan order kaca. Jika kaca tidak mencukupi maka akan dilakukan konfirmasi langsung ke pelanggan. Jika kaca mencukupi maka order kaca diberikan ke bagian pemotongan. Proses-proses tersebut semuanya dilakukan secara manual. Dari proses tersebut dibutuhkan suatu proses yang dapat membantu mempercepat proses pengecekan ketersediaan stok kaca terhadap order kaca pelanggan.

2. Analisis pada Alir Sistem Bagian Pemotongan

Alir sistem bagian pemotongan dimulai saat order kaca pelanggan telah diterima. Setelah order diterima dilakukan pengecekan ulang untuk persiapan proses pemotongan kaca. Setelah dilakukan pengecekan maka dilakukan perhitungan pola dan proses pemotongan. Dari proses - proses tersebut terdapat proses yang dapat menimbulkan

permasalahan dalam hal penggunaan waktu yang diperlukan. Proses tersebut adalah proses perhitungan pola pemotongan.

Untuk mengatasi permasalahan penggunaan waktu yang tinggi, diharapkan untuk selanjutnya dapat dilakukan peningkatan penggunaan waktu proses sehingga menjadi lebih singkat dan proses pemotongan bisa dilakukan lebih cepat.

3.3 Solusi Permasalahan

Setelah dilakukan pengumpulan data, dan kemudian mengolah data-data tersebut untuk dilakukan analisis, maka didapatkanlah suatu permasalahan yang harus diselesaikan dengan memberikan solusi terbaik yang sesuai untuk permasalahan yang ada. Dalam hal ini, solusi untuk permasalahan tersebut adalah membangun aplikasi yang dapat membantu proses menghitung pola pemotongan kaca supaya proses pemotongan bisa dilakukan secara efektif dan efisien. Solusi tersebut dikerjakan sesuai dengan tahapan pengembangan perangkat lunak yang terdiri dari:

3.3.1 Kebutuhan Perangkat Lunak (*Software Requirement*)

Kebutuhan perangkat lunak merupakan langkah awal dalam membangun sebuah sistem atau aplikasi agar sesuai dengan kebutuhan pengguna. Proses identifikasi kebutuhan perangkat lunak dibagi menjadi beberapa tahapan yaitu:

3.3.1.1. Elisitasi Kebutuhan (*Requirement Elicitation*)

Proses elisitasi dilakukan yaitu dengan cara wawancara dan observasi awal, namun yang dilakukan wawancara hanya kepada stakeholder yang terkait saja. Sebelum kebutuhan dapat dianalisis, kebutuhan harus dikumpulkan melalui proses elisitasi. Pada tahapan ini dilakukan penyeleksian data yang diperoleh sehingga dapat diketahui data-data yang digunakan dan yang tidak digunakan terkait dengan pengembangan perangkat lunak.

Berikut ini data yang dikumpulkan melalui proses wawancara ataupun observasi pada perusahaan. Data tersebut meliputi :

a) Data Kaca

Data kaca berisi keterangan-keterangan mulai dari ukuran standar kaca, jenis-jenis kaca, harga kaca dan stok. Kaca sendiri terbagi menjadi 2 macam, yaitu kaca lembaran dan kaca sisa pemotongan.

Tabel 3.2 Data Kaca

Jenis Kaca	Ketebalan Kaca	Ukuran lebar dan panjang	Stok
Kaca Polos	5 mm	122 x 153	30
Kaca Riben	5 mm	122 x 153	35
Kaca Cermin	5 mm	122 x 153	20
Kaca Polos	8 mm	214 x 305	15
Kaca Riben	3 mm	122 183	20

b) Data order kaca

Data order kaca digunakan untuk menginformasikan ukuran, jenis, dan jumlah kaca yang dibeli oleh pelanggan.

Tabel 3.3 Data Order Kaca

Jenis kaca	Ketebalan Kaca	Ukuran lebar dan panjang	Jumlah pembelian
Kaca Polos	5 mm	50 x 30	100
Kaca Riben	5 mm	100 x 50	70
Kaca Cermin	5 mm	90 x 45	50
Kaca Polos	8 mm	50 x 50	80
Kaca Riben	3 mm	40 x 50	30

c) Data Pegawai

Data pegawai digunakan untuk membagi hak akses dalam penggunaan aplikasi perangkat lunak yang akan dibangun

Tabel 3.4 Data Pegawai

Kode Pegawai	Nama Pegawai	Jabatan
Rj1	Rika	Bagian Administrasi 1
Rj2	Ali	Bagian Pematangan

3.3.1.2 Analisis Kebutuhan Administrasi

Setelah dilakukan analisis pada tahap yang sebelumnya, maka bagian administrasi membutuhkan sebuah sistem yang dapat menunjukkan ketersediaan stok yang ada di gudang. Data yang dibutuhkan oleh bagian administrasi adalah data kaca.

3.3.1.3 Analisis Kebutuhan Pemotongan

Setelah dilakukan analisis pada tahap sebelumnya, maka bagian pemotongan dan administrasi membutuhkan suatu aplikasi yang dapat membantu meningkatkan keefektifan dan keefisienan dalam proses pemotongan. Adapun peningkatan tersebut maka dilakukan proses sebagai berikut :

- a. Bagian pemotongan dapat memotong kaca sesuai dengan pola yang telah dihitung menggunakan *cutting stock optimization*.
- b. Bagian pemotongan tidak perlu menghitung pola secara manual.

Dengan adanya perubahan tersebut, maka proses kedepannya akan mengalami peningkatan pemanfaatan informasi yang lebih cepat. Data – data yang dibutuhkan oleh bagian pemotongan adalah data order kaca dan data kaca.

3.3.1.4 Spesifikasi Kebutuhan (*Requirement Specification*)

Dalam proses membangun dan mengembangkan perangkat lunak, diperlukan perancangan spesifikasi perangkat lunak yang tepat dan detail, dengan tujuan agar perangkat lunak yang akan dikembangkan tersebut memiliki deskripsi fungsi-fungsi sesuai dengan

apa yang dibutuhkan oleh penggunanya. Adapun kebutuhan fungsi-fungsi tersebut dapat dikelompokkan menjadi kebutuhan fungsional dan kebutuhan non-fungsional.

a. Kebutuhan Fungsional

Kebutuhan fungsional merupakan dasar dari penyusunan fungsi-fungsi yang akan dibangun didalam perangkat lunak. Fungsi-fungsi perangkat lunak tersebut telah melewati proses identifikasi kebutuhan setiap pengguna. Adapun kebutuhan fungsional tersebut yaitu:

1. Bagian Administrasi

Kebutuhan fungsional yang diperlukan untuk mendukung proses bagian administrasi terdiri dari 2 fungsi yaitu mencatat order pelanggan dan membuat purchase order.

2. Bagian Pemotongan

Kebutuhan fungsional untuk mendukung proses yang dilakukan oleh bagian pemotongan terdiri dari 2 fungsi yaitu fungsi menghitung pola pemotongan dan fungsi membuat laporan hasil pemotongan. Untuk lebih jelasnya dapat dilihat pada tabel 3.5.

Tabel 3.5 Kebutuhan Fungsional Pengguna.

Pengguna	Kebutuhan Fungsional	
Bagian Administrasi	Mencatat Order Pelanggan	Mengacu pada Lampiran 2
	Membuat Purchase Order	Mengacu pada Lampiran 2
Bagian Pemotongan	Menghitung Pola Pemotongan	Mengacu pada Lampiran 2
	Membuat Laporan Hasil Pemotongan	Mengacu pada Lampiran 2

b. Kebutuhan Non-Fungsional

Dalam penerapan fungsi-fungsi tersebut dengan tujuan mendukung kinerja fungsi utama dari sistem, berikut ini adalah peran dari non-fungsional yang mendukung kinerja fungsi-fungsi utama yang ditunjukkan pada Tabel 3.6.

Tabel 3.6 Keterkaitan Fungsional dan Non-Fungsional Sistem

No	Aktor	Fungsional Sistem	Non-Fungsional Sistem
1.	Bagian Administrasi	a) Mencatat Order Pelanggan b) Membuat <i>Purchase Order</i>	a) <i>Security</i> b) <i>Operability</i> c) <i>Time Behaviour</i> d) <i>Accuracy</i> e) <i>Maintain Ability</i>
2.	Bagian Pemotongan	a) Menghitung Pola Pemotongan b) Membuat Laporan Hasil Pemotongan	a) <i>Security</i> b) <i>Operability</i> c) <i>Time Behaviour</i> d) <i>Accuracy</i>

3.3.2 Desain Sistem (*Software Design*)

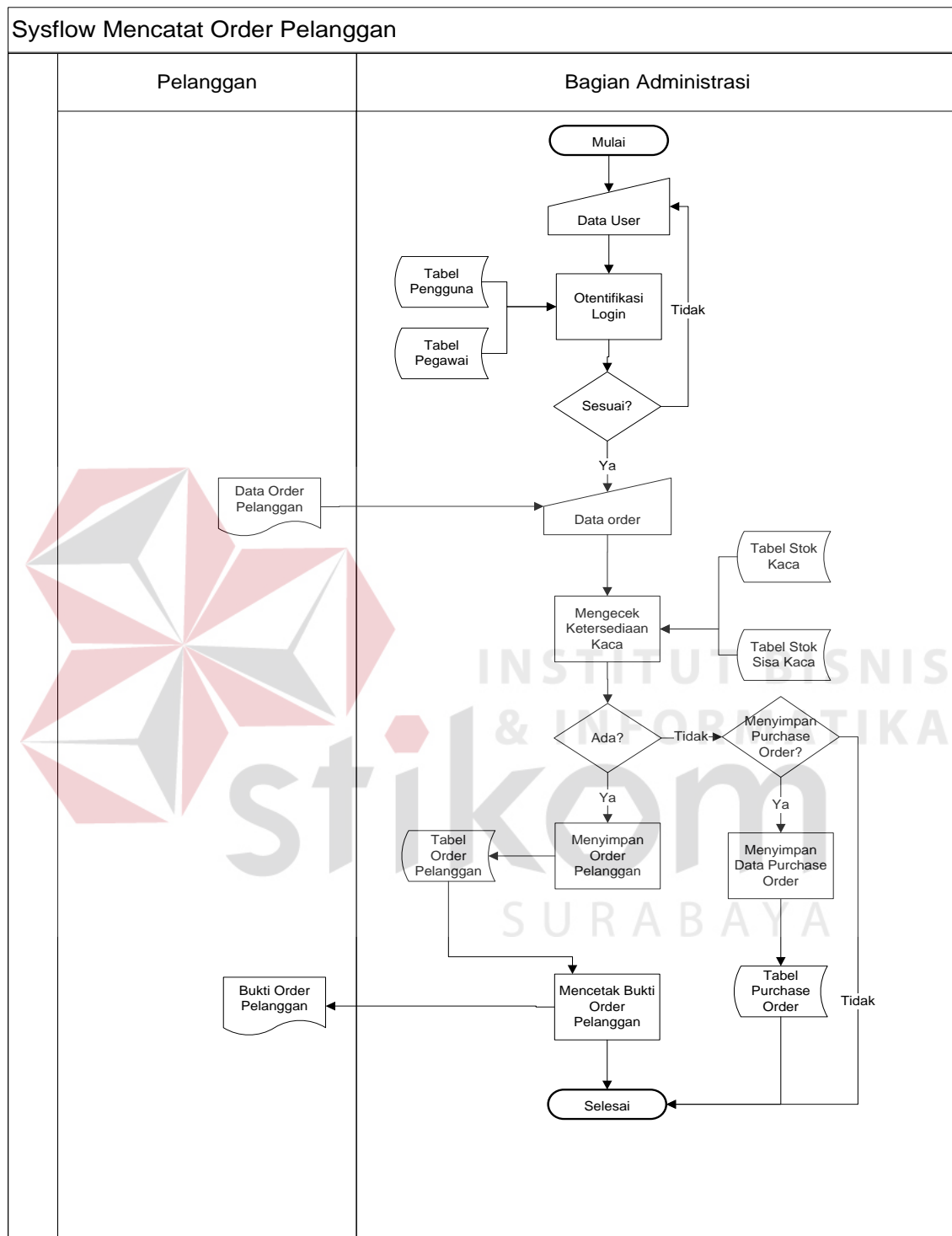
Rancangan perangkat lunak merupakan suatu kegiatan dalam merancang atau mendesain perangkat lunak yang akan dibangun sesuai dengan kebutuhan pengguna. Dikatakan sesuai dengan kebutuhan pengguna karena proses desain tersebut dilakukan berdasarkan hasil analisis kebutuhan yang telah dilakukan sebelumnya. Adapun rancangan perangkat lunak tersebut berupa rancangan alur sistem (*System Flow*), *Context Diagram*, *Data Flow Diagram*, dan rancangan tampilan antar muka (*Design Interface*).

3.3.2.1 Alur Sistem (*System Flow*)

Sesuai dengan hasil analisis kebutuhan, telah didefinisikan bahwa pengguna yang menggunakan sistem secara langsung berjumlah 2 (dua) pengguna yaitu Bagian Administrasi dan Bagian Pemotongan. Berikut merupakan alur sistem masing-masing pengguna yang baru.

a. Alur Sistem Bagian Administrasi

Berikut ini adalah rancangan alur sistem untuk Bagian Administrasi yang baru. Bagian Administrasi memiliki 2 fungsional sistem, yaitu Mencatat Order Pelanggan dan Membuat Purchase Order, yang ditunjukkan pada Gambar 3.2 dan Gambar 3.3.

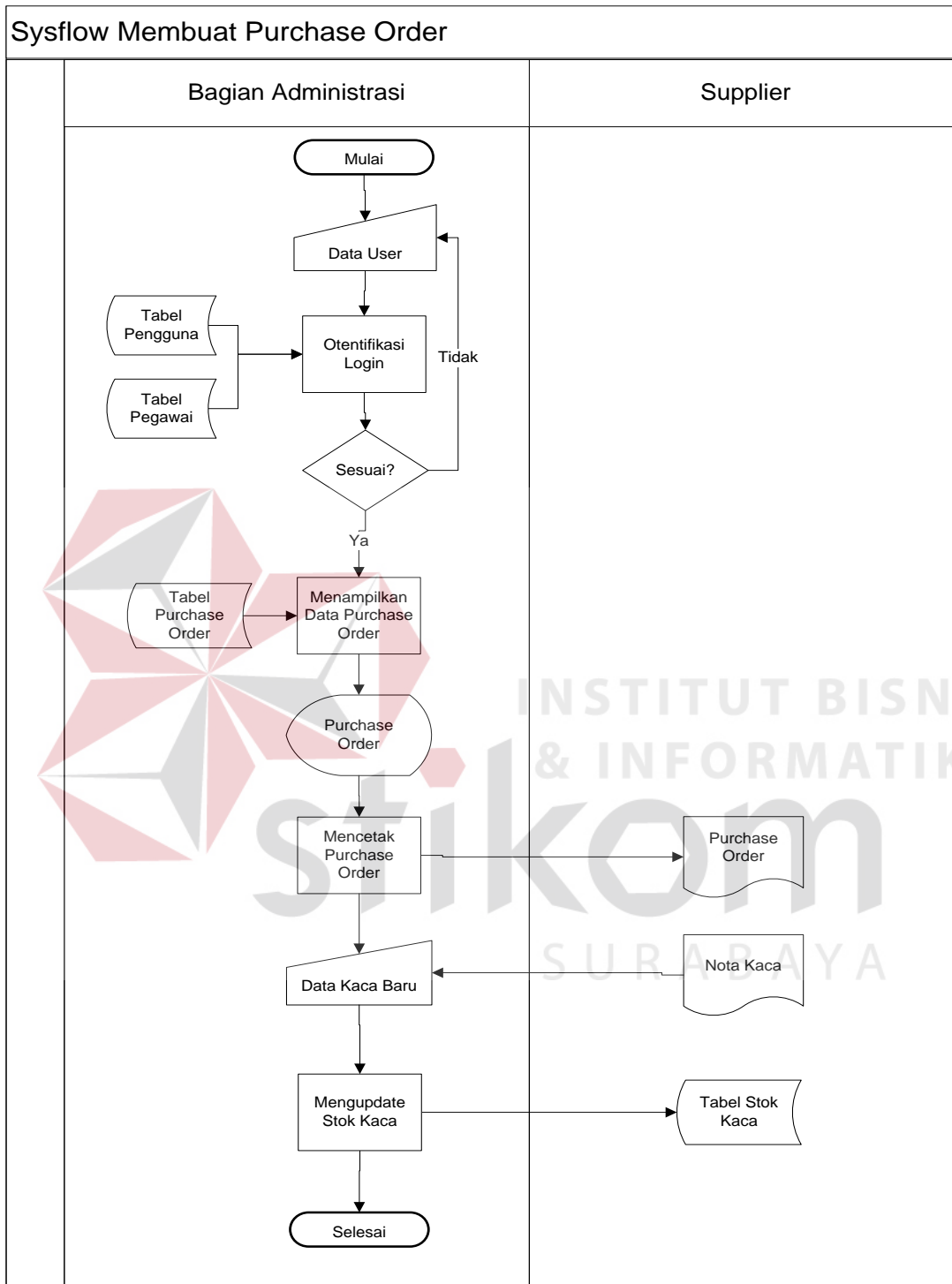


Gambar 3.2 Sysflow Mencatat Order Pemoangan

Penjelasan Alur Sistem Mencatat Order Pelanggan sesuai Gambar 3.2 dapat dilihat pada Tabel 3.7.

Tabel 3.7 Penjelasan Alur Sistem Mencatat Order Pematangan

Phase	No. Proses	Nama Proses	Input	Uraian Proses	Output
1	1	Input Data User	Data User	Menginputkan data user berupa Id_Pegawai dan Password	Data User
	2	Otentifikasi <i>Login</i>	Data User, Data Pegawai (Diambil dari Tabel)	Bagian Administrasi melakukan proses <i>login</i> sesuai dengan jabatan	Halaman Menu Utama
	3	<i>Input Data Order</i>	Data Order Pelanggan (Diambil dari dokumen)	Bagian Administrasi menerima data order dari pelanggan dan menginputkan ke sistem.	Inputan data order pelanggan
	4	Mengecek Ketersediaan Kaca	Inputan data order pelanggan	Sistem mengecek ketersediaan jumlah order kaca dengan stok kaca	Data stok kaca
	5	Menyimpan Data <i>Purchase Order</i>	Data order pelanggan	Menyimpan Data <i>Purchase Order</i>	Data <i>Purchase Order</i>
	6	Menyimpan Order Pelanggan	Data order pelanggan	Sistem menyimpan data order pelanggan ke tabel order pelanggan	Data order pelanggan tersimpan
	7	Mencetak Bukti Order pelanggan	Data order pelanggan	Sistem mencetak bukti order pelanggan dan diberikan kembali ke pelanggan sebagai bukti	Dokumen bukti order pelanggan



Gambar 3.3 Sysflow Membuat *Purchase Order*

Penjelasan Alur Sistem Membuat Purchase Order sesuai Gambar 3.3 dapat dilihat pada Tabel 3.8.

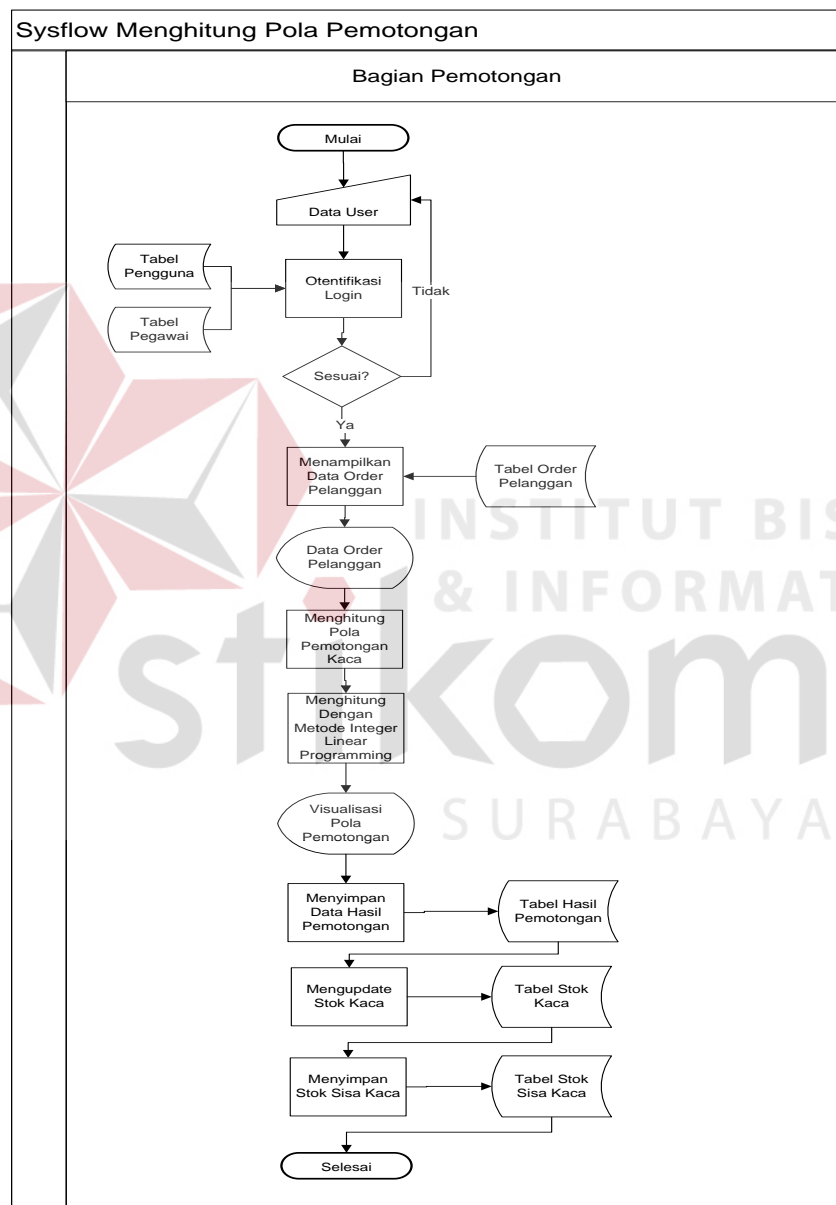
Tabel 3.8 Penjelasan Alur Sistem Membuat Purchase Order

Phase	No. Proses	Nama Proses	Input	Uraian Proses	Output
1	1	Input Data User	Data User	Menginputkan data user berupa Id_Pegawai dan Password	Data User
	2	Otentifikasi <i>Login</i>	Data User, Data Pegawai (Diambil dari Tabel)	Bagian Administrasi melakukan proses <i>login</i> sesuai dengan jabatan	Halaman Menu Utama
	3	Menampilkan Data <i>Purchase Order</i>	Data Stok Kaca, Data <i>Purchase Order</i> (Diambil dari tabel)	Sistem mengambil data dari tabel stok kaca dan tabel <i>purchase order</i> , kemudian menampilkannya ke layar	<i>Display</i> data <i>Purchase Order</i>
	4	Mencetak <i>Purchase Order</i>	Data <i>Purchase Order</i> Update	Sistem mencetak <i>purchase order</i>	<i>Purchase Order</i>
	5	Menginputkan Data Kaca Baru	Nota Kaca	Menginputkan Data Kaca yang telah selesai disorder	Data Kaca Diorder
	6	Mengupdate Stok Kaca	Stok Kaca Diorder	Mengupdate data stok kaca sesuai dengan jumlah yang telah disorder	Data Update Stok Kaca

b. Alur Sistem Pemotongan

Berikut ini adalah rancangan alur sistem untuk Pimpinan yang ditunjukkan pada

Gambar 3.4.



Gambar 3.4 Sysflow Menghitung Pola Pemotongan

Penjelasan alur sistem sesuai dengan Gambar 3.4 dapat dilihat pada Tabel 3.9.

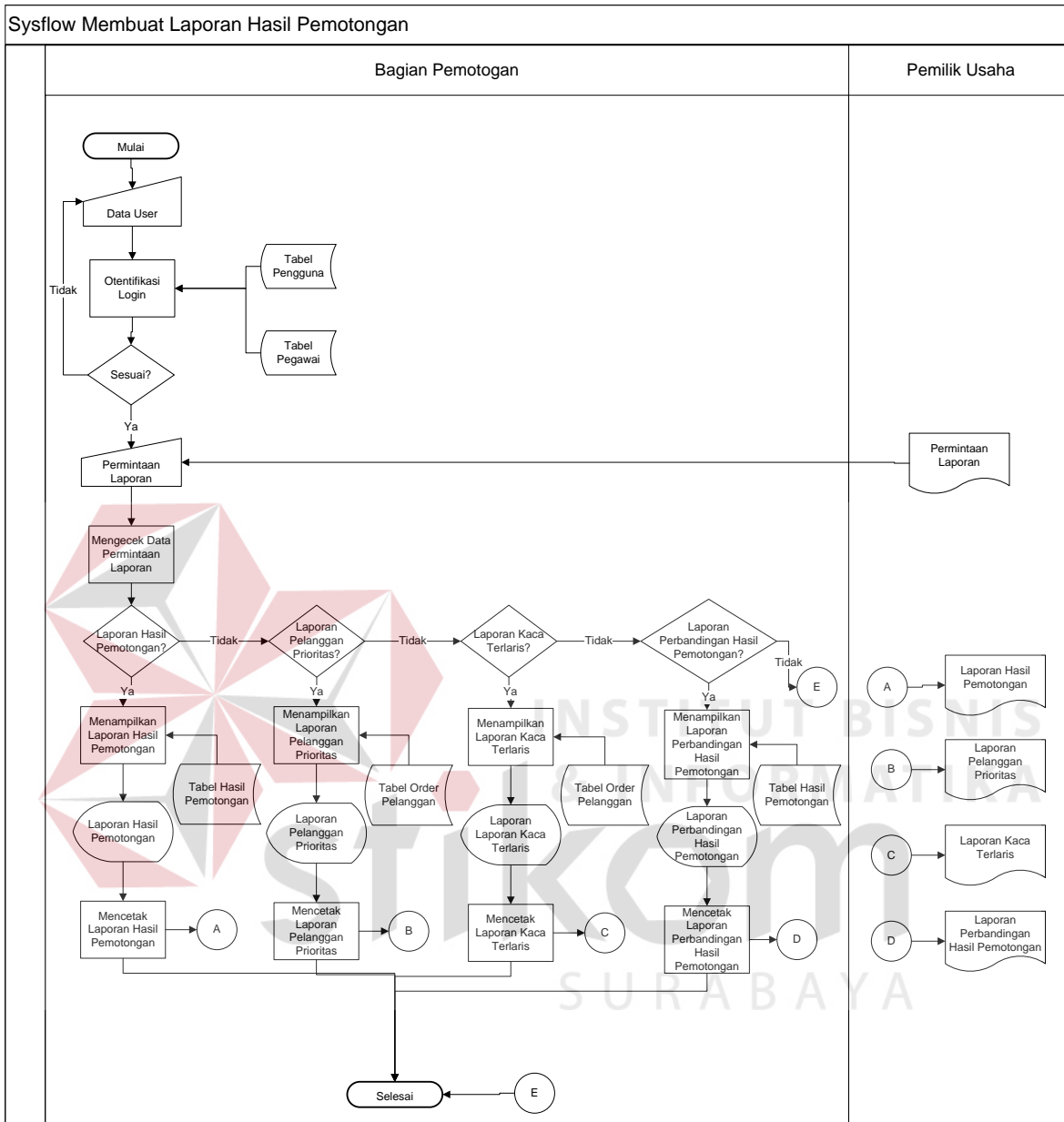
Tabel 3.9 Penjelasan Alur Sistem Menghitung Pola Pemotongan

Phase	No. Proses	Nama Proses	Input	Uraian Proses	Output
1	1	Input Data User	Data User	Menginputkan data user berupa Id_Pegawai dan Password	Data User
	2	Otentifikasi <i>Login</i>	Data User, Data Pegawai (Diambil dari Tabel)	Bagian Administrasi melakukan proses <i>login</i> sesuai dengan jabatan	Halaman Menu Utama
	3	Menampilkan Data Order Pelanggan	Data order pelanggan (Diambil dari tabel)	Sistem menampilkan data order pelanggan dari tabel order pelanggan dan menampilkannya ke layar	<i>Display</i> data order pelanggan
	4	Menghitung Pola Pemotongan Kaca	<i>Display</i> data order pelanggan	Sistem menghitung kebutuhan pola pemotongan	Data Pemotongan
	5	Menghitung Dengan Metode <i>Integer Linear Programming</i>	Data Pemotongan	Sistem menghitung pola pemotongan dengan metode <i>integer linear programming</i>	Visualisasi Pola Pemotongan
	6	Menyimpan Data Hasil Pemotongan	Data Pola Pemotongan	Sistem menyimpan hasil pemotongan ke dalam tabel hasil pemotongan	Data Hasil Pemotongan
	7	Mengupdate Stok Kaca	Data Stok Hasil Pemotongan	Sistem mengupdate jumlah stok kaca sesuai yang dipakai	Data Stok Kaca

				dalam proses pemotongan	
	8	Menyimpan Stok Sisa Kaca	Data Stok Hasil Pemotongan	Sistem menyimpan jumlah stok sisa kaca dalam proses pemotongan	Data Stok Sisa Kaca

Berikut ini adalah rancangan alur sistem Membuat Laporan Hasil Pemotongan yang ditunjukkan oleh Gambar 3.5.





Gambar 3.5 Sysflow Membuat Laporan Hasil Pemotongan

Penjelasan untuk Gambar 3.5 dapat dilihat pada Tabel 3.10.

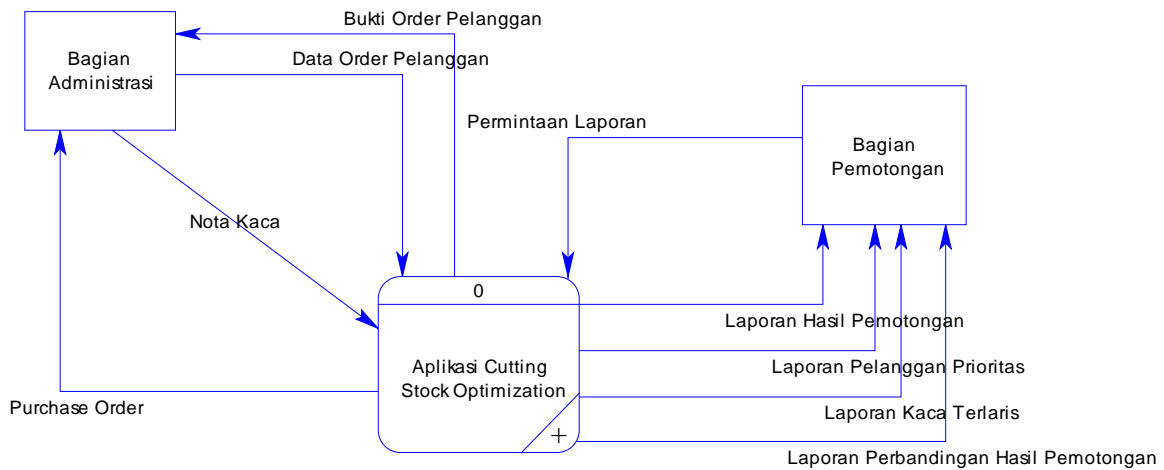
Tabel 3.10 Penjelasan Alur Sistem Membuat Laporan Hasil Pemotongan.

Phase	No. Proses	Nama Proses	Input	Uraian Proses	Output
1	1	Input Data User	Data User	Menginputkan data user berupa Id_Pegawai dan Password	Data User
	2	Otentifikasi <i>Login</i>	Data User, Data Pegawai (Diambil dari Tabel)	Bagian Administrasi melakukan proses <i>login</i> sesuai dengan jabatan	Halaman Menu Utama
	3	Menginputkan Permintaan Laporan	Dokumen permintaan laporan	Permintaan laporan diinputkan kedalam sistem	Data Permintaan Laporan
	4	Mengecek Data Permintaan Laporan	Data Permintaan Laporan	Sistem mengecek laporan apa yang akan ditampilkan	Data Permintaan Laporan
	5	Menampilkan Laporan Hasil Pemotongan	Data Permintaan Laporan, Data Hasil Pemotongan	Sistem mengambil data pemotongan dari tabel hasil pemotongan dan menampilkannya ke layar	Display Laporan Hasil Pemotongan
	6	Mencetak Laporan Hasil Pemotongan	Display Laporan Hasil pemotongan	Sistem mencetak laporan hasil pemotongan untuk dicetak	Laporan Hasil Pemotongan
	7	Menampilkan Laporan Pelanggan Prioritas	Data Permintaan Laporan, Data Pelanggan	Sistem mengambil data pelanggan yang paling banyak melakukan order dari tabel order pelanggan dan menampilkannya ke	Display Laporan Pelanggan Prioritas

				layar	
	8	Mencetak Laporan Pelanggan Prioritas	Display Laporan Pelanggan Prioritas	Sistem mencetak laporan pelanggan prioritas untuk dicetak	Laporan Pelanggan Prioritas
	9	Menampilkan Laporan Kaca Terlaris	Data Permintaan Laporan, Data Jenis Kaca	Sistem mengambil data jenis kaca yang paling banyak diorder dari tabel order pelanggan dan menampilkannya ke layar	Display Laporan Kaca Terlaris
	10	Mencetak Laporan Kaca Terlaris	Display Laporan Kaca Terlaris	Sistem mencetak laporan kaca terlaris untuk dicetak	Laporan Kaca Terlaris
	11	Menampilkan Laporan Perbandingan Hasil Pemotongan	Data Permintaan Laporan, Data Perbandingan Hasil Pemotongan	Sistem mengambil data perbandingan hasil pemotongan dari tabel hasil pemotongan dan menampilkannya ke layar	Display Laporan Perbandingan Hasil Pemotongan
	12	Mencetak Laporan Perbandingan Hasil Pemotongan	Display Laporan Perbandingan Hasil Pemotongan	Sistem mencetak laporan perbandingan hasil pemotongan untuk dicetak	Laporan Perbandingan Hasil Pemotongan

3.3.2.2 Context Diagram

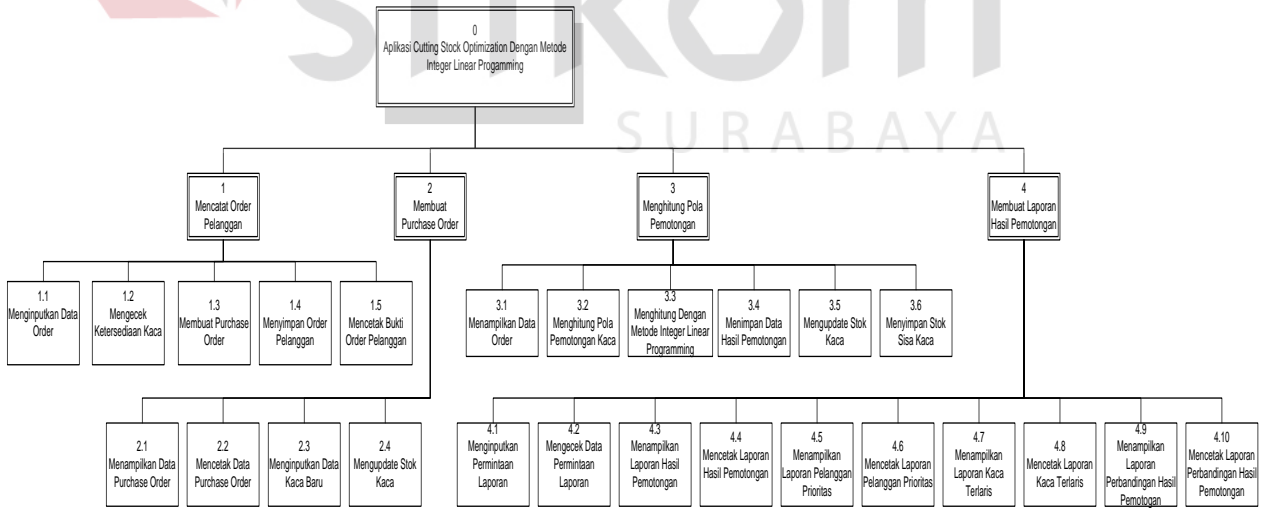
Berikut ini adalah desain *Context Diagram* untuk perangkat lunak yang akan dikembangkan yang dapat dilihat pada Gambar 3.6.



Gambar 3.6 Context Diagram

3.3.2.3 Diagram Berjenjang

Diagram Berjenjang merupakan diagram yang menggambarkan pembagian fungsi-fungsi dari sistem menjadi sub sistem yang lebih kecil. Diagram Berjenjang untuk aplikasi yang dikembangkan dapat dilihat pada Gambar 3.7.



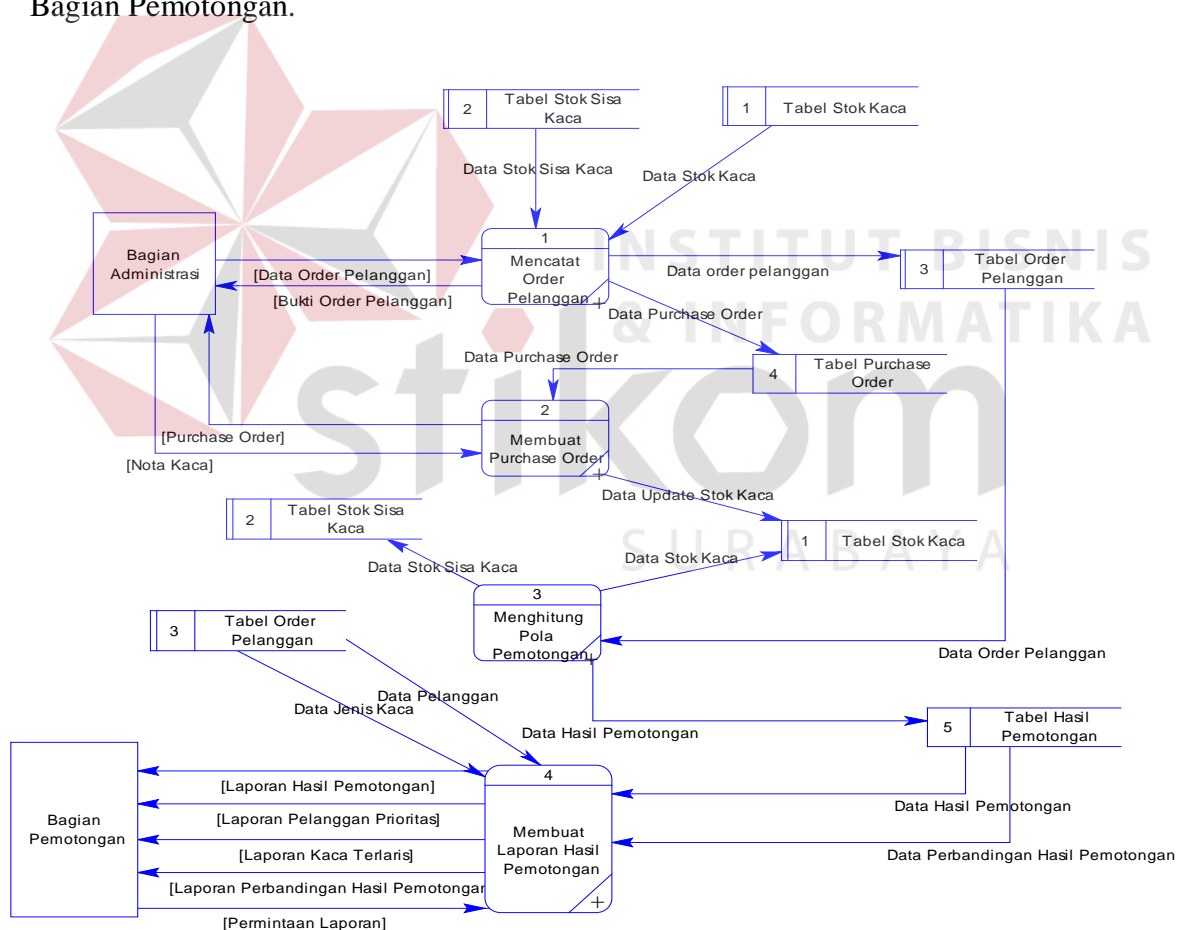
Gambar 3.7 Diagram Jenjang

3.3.2.4 Data Flow Diagram

Data Flow Diagram (DFD) untuk aplikasi yang sedang dikembangkan telah didefinisikan menjadi sub sistem Level 0 yang terdiri dari:

a) Data Flow Diagram Level 0

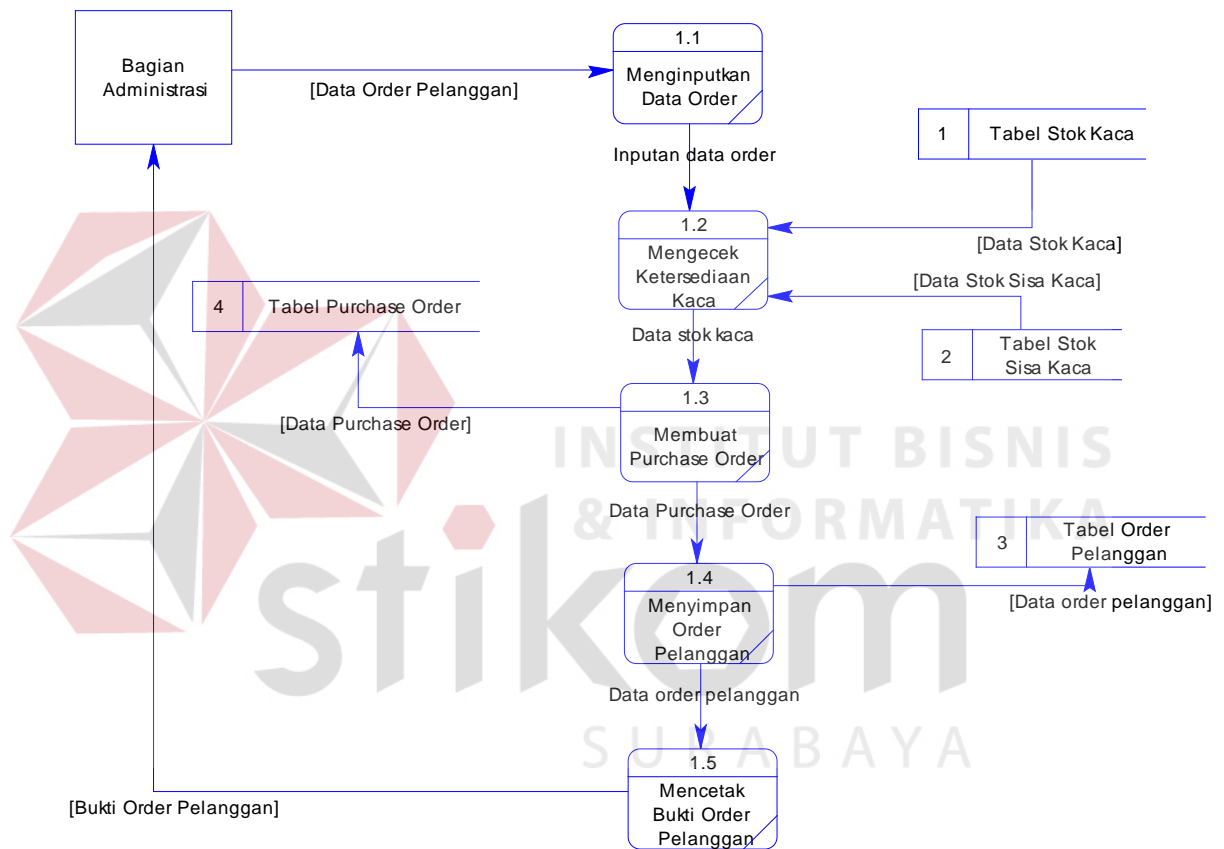
Pada Level 0 terlihat pada Gambar 3.8, aliran data yang masuk ke sistem, proses-proses yang dilakukan, dan tabel yang dibutuhkan berasal dari Bagian Administrasi dan Bagian Pemotongan.



Gambar 3.8 DFD Level 0

b) *Data Flow Diagram Level 1* Mencatat Order Pelanggan.

Untuk *Level 1* Mencatat Order Pelanggan, sistem menerima masukan data dari Bagian Administrasi berupa data order kaca yang akan disimpan didalam database. DFD *Level 1* Mencatat Order Pelanggan dapat dilihat pada Gambar 3.9.

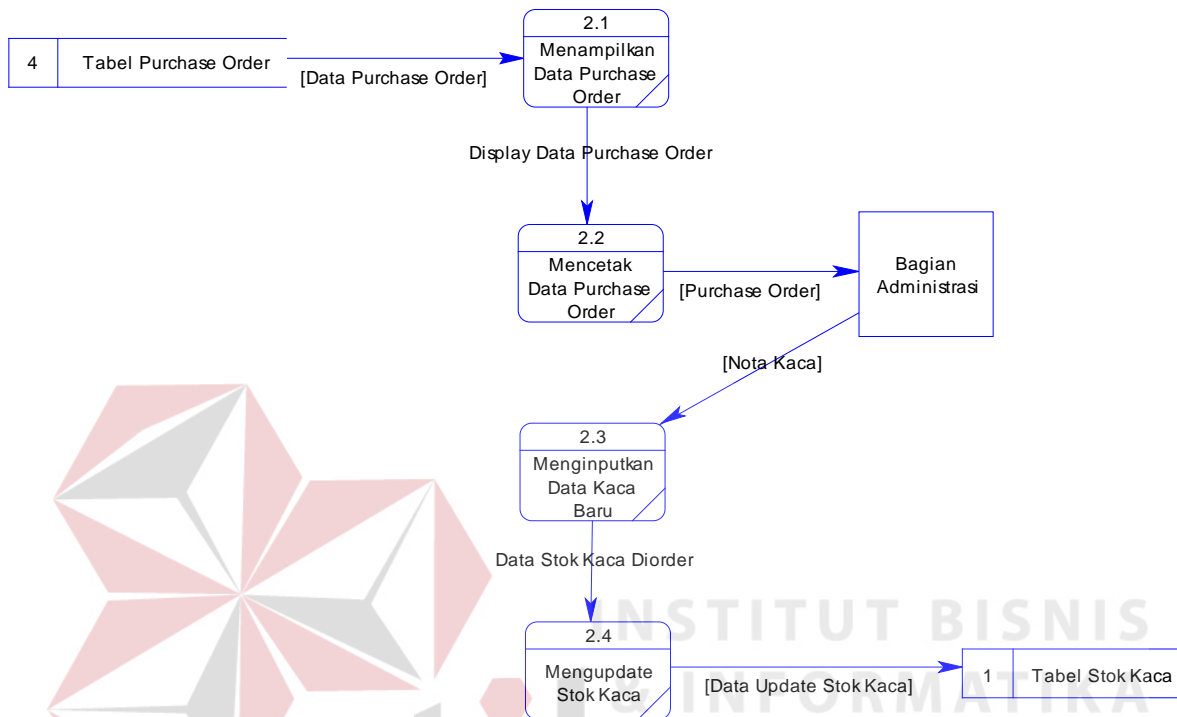


Gambar 3.9 DFD Level 1 Mencatat Order Pelanggan

c) *Data Flow Diagram Level 1* Membuat Purchase Order.

Sedangkan untuk *Level 1* Membuat *Purchase Order*, sistem akan menampilkan data purchase order yang diambil dari tabel stok kaca dan *purchase order*. Setelah data

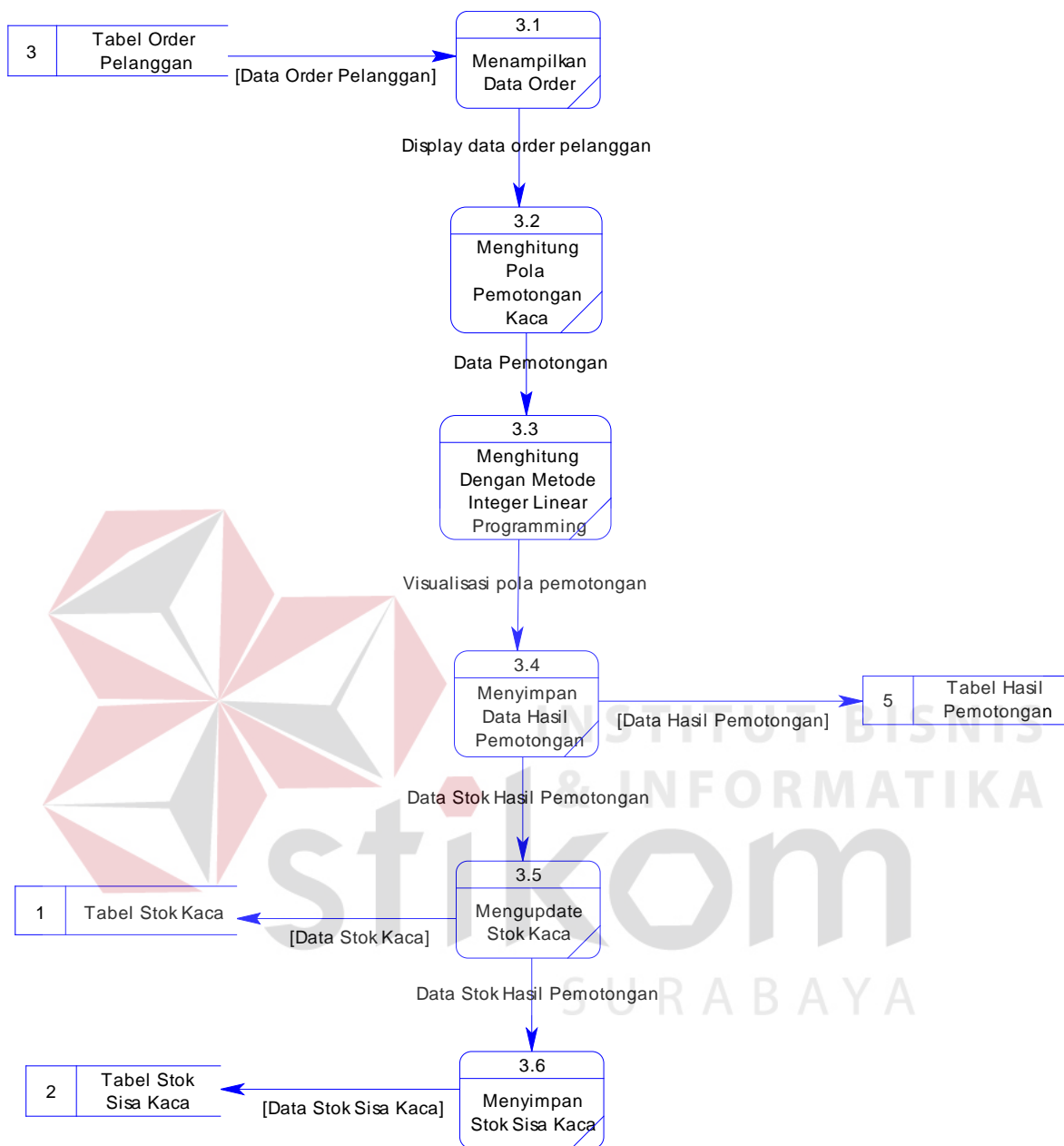
ditampilkan kemudian dapat dicetak menjadi *purchase order*. DFD *Level 1* Membuat *Purchase Order* dapat dilihat pada Gambar 3.10.



Gambar 3.10 DFD Level 1 Membuat *Purchase Order*

d) *Data Flow Diagram Level 1* Menghitung Pola Pemotongan.

Untuk DFD *Level 1* Menghitung Pola Pemotongan, sistem menampilkan data order kaca yang akan digunakan dalam proses pemotongan. Setelah data ditampilkan, kemudian diolah untuk mencari pola pemotongan terbaik. Terakhir hasil pemotongan akan disimpan ke dalam database. DFD *Level 1* Menghitung Pola Pemotongan dapat dilihat pada Gambar 3.11.

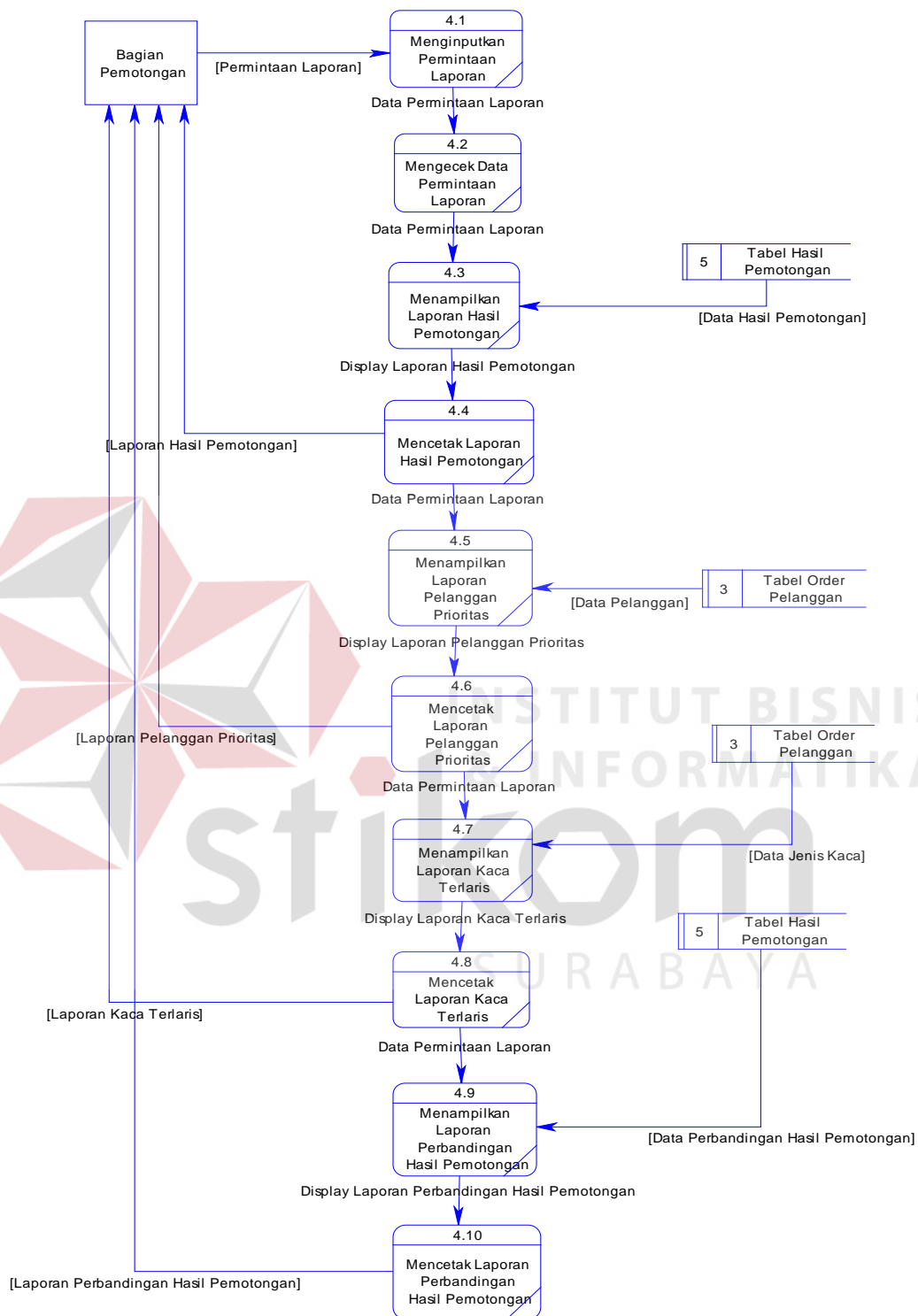


Gambar 3.11 DFD Level 1 Menghitung Pola Pemotongan

e) *Data Flow Diagram Level 1* Membuat Laporan Hasil Pemotongan.

Dalam DFD *Level 1* Membuat Laporan Hasil Pemotongan, sistem menampilkan laporan hasil pemotongan kaca, berdasarkan periode yang diinputkan. DFD *Level 1* Membuat Laporan Hasil Pemotongan dapat dilihat pada Gambar 3.12.





Gambar 3.12 DFD Level 1 Membuat Laporan Hasil Pemotongan

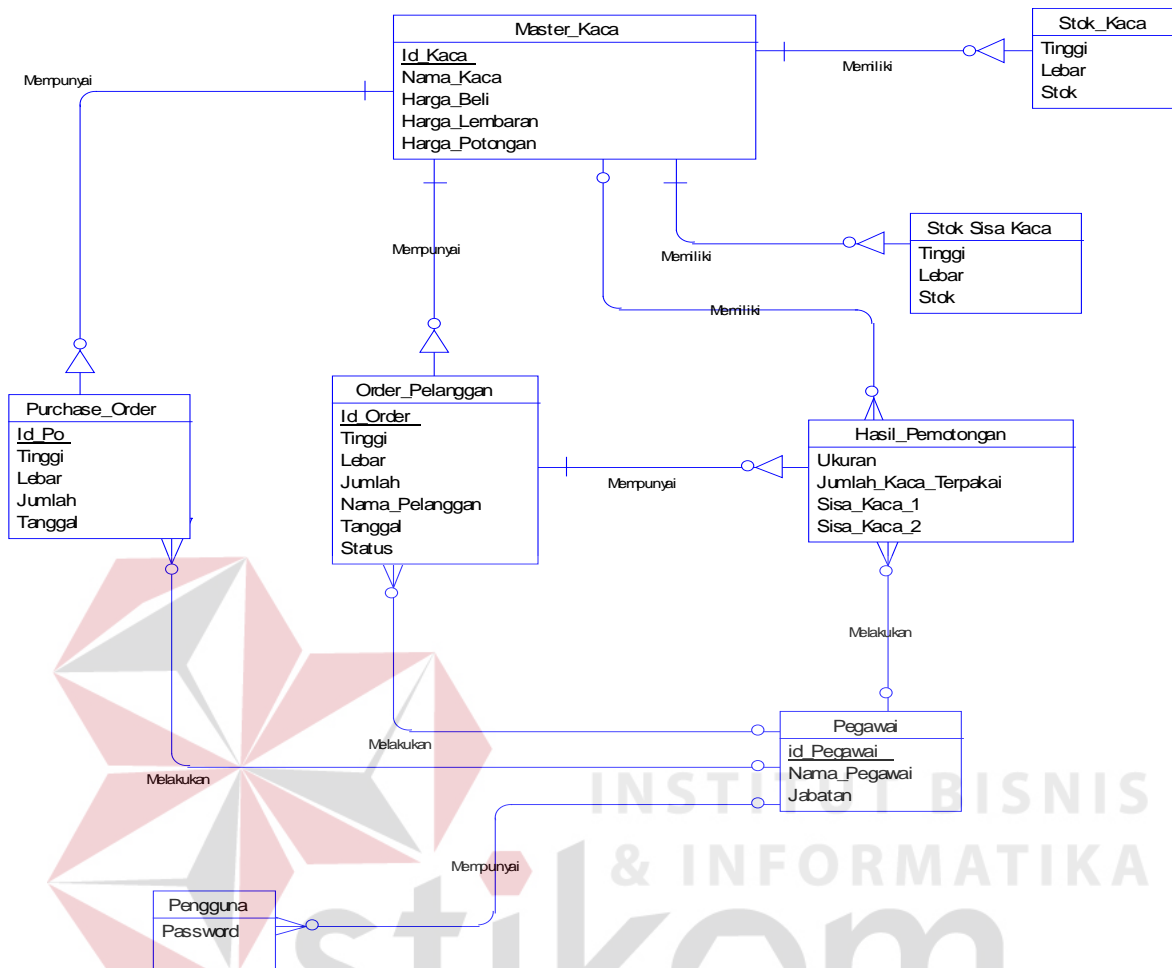
3.3.2.5 Entity Relationship Diagram

Entity Relationship Diagram (ERD) adalah suatu desain sistem yang digunakan untuk merepresentasikan, menentukan, dan mendokumentasikan kebutuhan sistem kedalam suatu bentuk dengan tujuan untuk menunjukkan struktur keseluruhan dari data pemakai. Dalam perancangan aplikasi ini, telah terbentuk ERD yang merupakan lanjutan dari pembuatan desain dengan menggunakan DFD, yang disimbolkan dengan menggunakan simbol *entity*. Adapun entity tersebut adalah:

- a) Bagian Administrasi
- b) Bagian Pemotongan

3.3.2.6 Conceptual Data Model

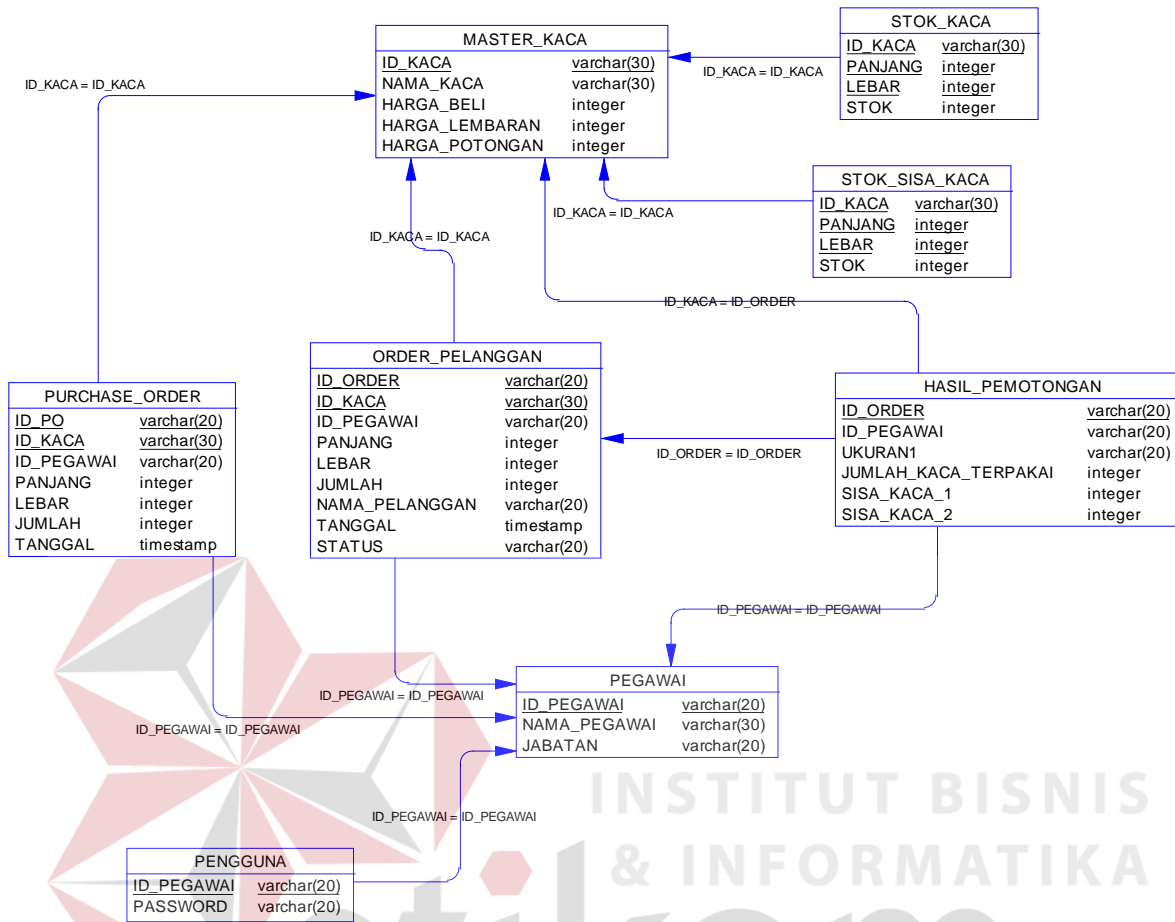
Conceptual Data Model merupakan gambaran secara keseluruhan tentang konsep struktur basis data yang dirancang untuk program atau aplikasi. Pada CDM belum tergambar jelas bentuk tabel-tabel penyusun basis data beserta *field* yang terdapat pada setiap tabel. Adapun CDM yang dirancang untuk Rancang Bangun Aplikasi *Cutting Stock Optimization* seperti tampak pada Gambar 3.13.



Gambar 3.13 Conceptual Data Model

3.3.2.7 Physical Data Model

Physical Data Model (PDM) menggambarkan secara detail konsep rancangan struktur basis data untuk suatu program atau aplikasi. PDM terbentuk dari *Conceptual Data Model* (CDM) yang menggambarkan tabel-tabel penyusun basis data beserta *field-field* yang terdapat pada setiap tabel. Adapun PDM untuk aplikasi seperti terlihat pada Gambar 3.14.



Gambar 3.14 Physical Data Model

3.3.2.8 Struktur Basis Data

Sesuai dengan PDM yang telah terbentuk, dapat dibentuk suatu struktur basis data yang akan digunakan untuk penyimpanan data yaitu:

1. Master Kaca

Primary Key : Id_Kaca

Foreign Key : -

Fungsi : Untuk menyimpan data master dari kaca

Tabel 3.11 Master Kaca.

No.	Nama Field	Constraint	Tipe	Size	Deskripsi
1	Id_Kaca	PK	Varchar	30	Kode Kaca
2	Nama_Kaca		Varchar	30	Nama Kaca
3	Harga_Beli		Integer		Harga Beli Kaca
4	Harga_Lembaran		Integer		Harga Lembaran
5	Harga_Potongan		Integer		Harga_Potongan

2. Tabel Stok Kaca

Primary Key : Id_Kaca, Panjang, Lebar

Foreign Key : Id_Kaca

Fungsi : Untuk menyimpan stok kaca

Tabel 3.12 Stok Kaca.

No.	Nama Field	Constraint	Tipe	Size	Deskripsi
1	Id_Kaca	PK, FK	Varchar	30	Kode Kaca
2	Panjang	PK	Integer		Panjang Kaca
3	Lebar	PK	Integer		Lebar Kaca
4	Stok		Integer		Jumlah Stok

3. Tabel Stok Sisa Kaca

Primary Key : Id_Kaca, Panjang, Lebar

Foreign Key : Id_Kaca

Fungsi : Untuk menyimpan stok kaca sisa

Tabel 3.13 Stok Sisa Kaca.

No.	Nama Field	Constraint	Tipe	Size	Deskripsi
1	Id_Kaca	PK, FK	Varchar	30	Kode Kaca
2	Panjang	PK	Integer		Panjang Kaca
3	Lebar	PK	Integer		Lebar Kaca
4	Stok		Integer		Jumlah Stok

4. Tabel Order Pelanggan

Primary Key : Id_Order

Foreign Key : Id_Kaca

Fungsi : Untuk menyimpan order dari pelanggan

Tabel 3.14 Order Pelanggan.

No.	Nama Field	Constraint	Tipe	Size	Deskripsi
1	Id_Order	PK	Varchar	20	Kode Purchase Order

2	Id_Kaca	FK	Varchar	30	Kode Kaca
3	Panjang		Integer		Panjang Kaca
4	Lebar		Integer		Lebar Kaca
5	Tanggal		Datetime		Tanggal
6	Jumlah		Integer		Jumlah Kaca
7	Status		Varchar	20	Status Order

6. Tabel *Purchase Order*

Primary Key : Id_Penerimaan_K

Foreign Key : Id_Kaca

Fungsi : Untuk menyimpan transaksi penerimaan kaca

Tabel 3.15 *Purchase Order*.

No.	Nama Field	Constraint	Tipe	Size	Deskripsi
1	Id_Po	PK	Varchar	20	Kode Purchase Order
2	Id_Kaca	FK	Varchar	30	Kode Kaca
3	Panjang		Integer		Panjang Kaca
4	Lebar		Integer		Lebar Kaca
5	Tanggal		Datetime		Tanggal
6	Jumlah		Integer		Jumlah Kaca

7. Tabel Hasil Pemotongan

Primary Key : Id_Pemotongan

Foreign Key : Id_Kaca

Fungsi : Untuk menyimpan transaksi penerimaan kaca

Tabel 3.16 Hasil Pemotongan.

No.	Nama Field	Constraint	Tipe	Size	Deskripsi
1	Id_Pemotongan	PK	Varchar	20	Kode Pemotongan
2	Id_Kaca	FK	Varchar	30	Kode Kaca
3	Ukuran		Varchar	20	Ukuran kaca yang dipotong
4	Jumlah		Integer		Lebar Kaca
5	Jumlah_Kaca_Terpakai		Integer		Jumlah kaca yang dipakai dalam proses pemotongan
6	Jumlah_Sisa_Kaca		Integer		Jumlah sisa kaca hasil pemotongan

8. Tabel Pegawai

Primary Key : Id_Pegawai

Foreign Key : -

Fungsi : Untuk menyimpan data pegawai

Tabel 3.17 Pegawai.

No.	Nama Field	Constraint	Tipe	Size	Deskripsi
1	Id_Pegawai	PK	Varchar	20	Kode Pegawai
2	Nama_Pegawai		Varchar	30	Nama Pegawai
3	Jabatan		Varchar	20	Jabatan Pegawai

3.3.2.9 Perancangan Prosedur dan Program Unit

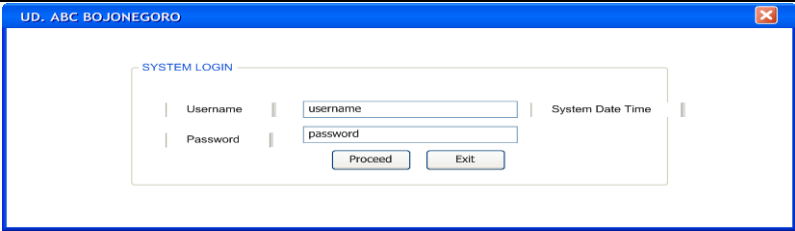
Detil sistem merupakan penjabaran pemrograman aplikasi dengan menggunakan *pseudocode* sehingga konstruksi awal pemrograman aplikasi yang akan dibangun dapat terlihat serta memberikan deskripsi dari setiap fungsi yang akan dibangun disertai dengan desain tampilan antarmuka aplikasi. Pada Tugas Akhir ini, penjabaran detil dari sistem akan dibagi berdasarkan pengguna aplikasi tersebut sesuai dengan yang telah dijelaskan sebelumnya.

a) *Form Login*

Menampilkan menu untuk masuk ke aplikasi, seperti terlihat pada Tabel 3.21.

Tabel 3.18 Detil *Form Login*

<i>Descriptions</i>	Form <i>Login</i> merupakan <i>form</i> yang akan tampil pertama kali saat pengguna akan menggunakan aplikasi. <i>Form Login</i> ini akan selalu menjadi tampilan awal di setiap pengguna.
<i>Table Input</i>	Tabel <i>login</i>
<i>Table Output</i>	-

<i>Interface</i>	 <p style="text-align: center;">Gambar 3.15 Desain <i>Form Login</i></p>	
<i>Functional</i>	-	
<i>Non-Functional</i>	<i>Security</i>	
<i>Query</i>	<i>Select</i>	<i>Select username, pass_word from login</i>
<i>Pseudocode</i>	<i>Begin</i> <i>Declare</i> <i>Koneksi() Login()</i> <i>End</i>	

Tabel 3.19 Detil *Form* Mencatat Order Pelanggan

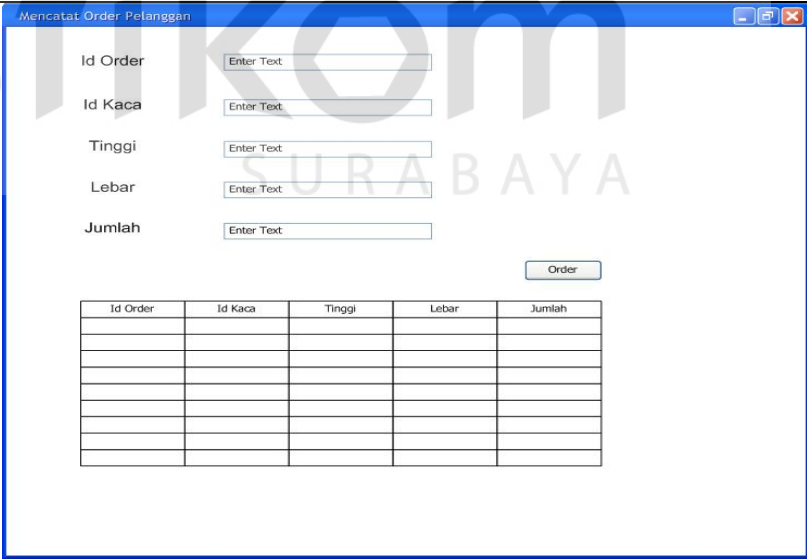
NamaFungsi	Mencatat Order Pelanggan
Stakeholder	Bagian Administrasi
Deskripsi	Mencatat order pemotongan dari pelanggan
Desain Interface	 <p style="text-align: center;">Gambar 3.16 Desain <i>Form</i> Mencatat Order Pelanggan</p>

Table Input	Tabel Pengguna, Tabel Pegawai
Table Output	Tabel Order Pelanggan
Query	Insert into tabel_order_pelanggan ()
Pseudocode	Begin Declare Otentikasi() Menginputkan_Order_Pelanggan() Menyimpan_Order_Pelanggan() Mencetak_Bukti_Order_Pelanggan() End
Kebutuhan Non-Fungsional	<i>Security</i> <i>Correctness</i> <i>Interface</i> <i>Performance</i> <i>Operability</i>

Tabel 3.20 Detil Form Membuat *Purchase Order*

NamaFungsi	Membuat <i>Purchase Order</i>
Stakeholder	Bagian Administrasi
Deskripsi	Membuat <i>purchase order</i> dari stok kaca yang jumlahnya hampir habis untuk dilakukan order pembelian

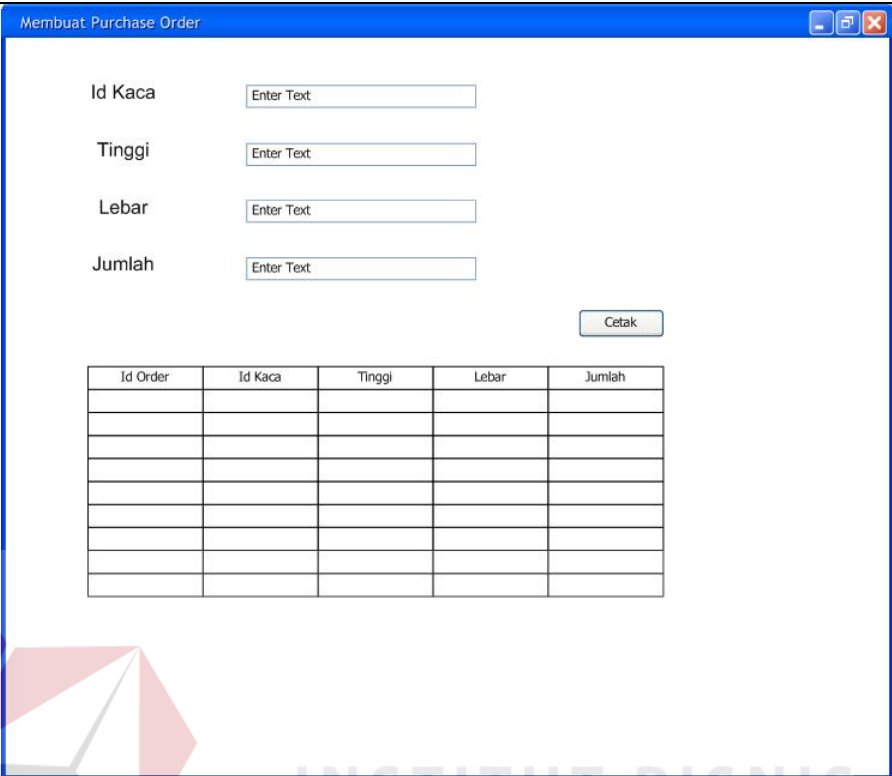
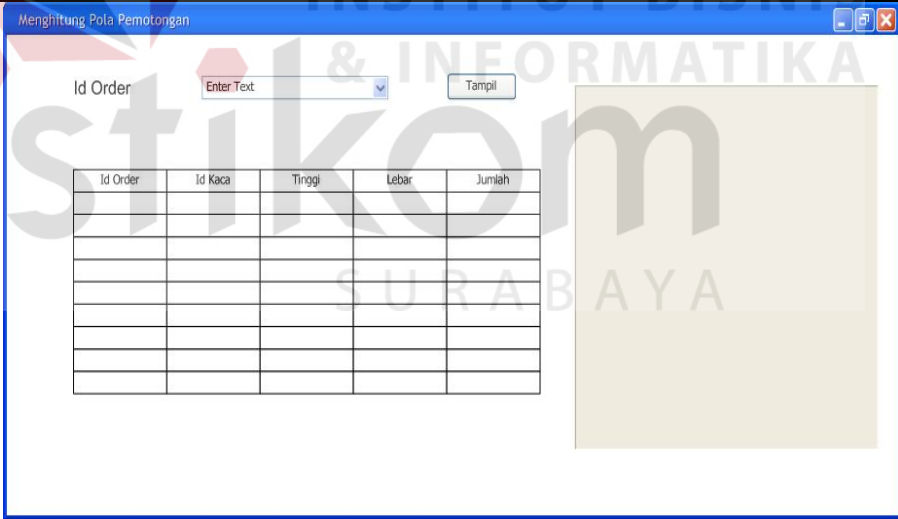
Desain Interface	
	<p style="text-align: center;">Gambar 3.17 Desain <i>Form Membuat Purchase Order</i></p>

Table Input	Tabel Pengguna, Tabel Pegawai, Tabel Purchase Order, Tabel Stok kaca
Table Output	-
Query	<pre>Select * from stok_kaca where stok<=5 Insert into tabel_order_pelanggan ()</pre>
Pseudocode	<pre>Begin Declare Otentifikasi() Menampilkan_Data_Purchase_Order() Mengupdate_Jumlah_Purchase_Order() Mencetak_Purchase_Order()</pre>

	End
Kebutuhan Non-Fungsional	<i>Security</i> <i>Correctness</i> <i>Interface</i> <i>Performance</i> <i>Operability</i>

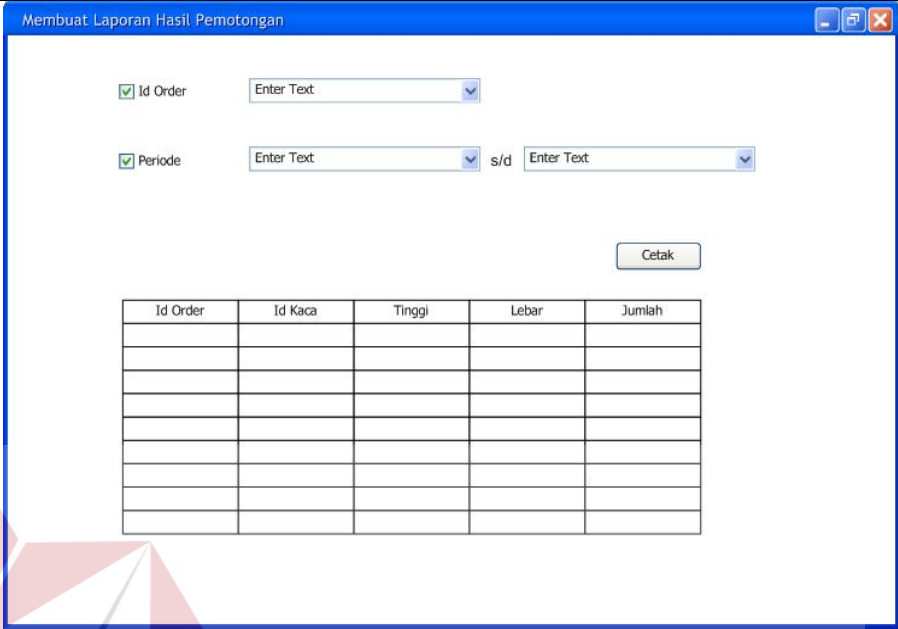
Tabel 3.21 Detil *Form* Menghitung Pola Pemotongan

NamaFungsi	Menghitung Pola Pemotongan
Stakeholder	Bagian Pemotongan
Deskripsi	Menghitung pola pemotongan terbaik berdasarkan ukuran dari order kaca pelanggan
Desain Interface	 <p style="text-align: center;">Gambar 3.18 Desain <i>Form</i> Menghitung Pola Pemotongan</p>
Table Input	Tabel Pegawai, Tabel Pengguna, Tabel Order Pelanggan, Tabel Stok Kaca
Table Output	Tabel Hasil Pemotongan, Tabel <i>Purchase Order</i>

Query	Select * from order_pelanggan Insert into hasil_pemotongan ()
Pseudocode	Begin Declare Otentikasi() Menampilkan_Data_Order_Pelanggan() Menghitung_Pola_Pemotongan_Kaca() Mengecek_Ketersediaan_Kaca() Menyimpan_Jumlah_Kekurangan_Kaca() Menghitung_Dengan_Metode_Integer_Linear_Programming() Menyimpan_Data_Hasil_Pemotongan() Mengupdate_Stok_Kaca() Menyimpan_Stok_Sisa_Kaca() End
Kebutuhan Non-Fungsional	<i>Security</i> <i>Correctness</i> <i>Interface</i> <i>Performance</i> <i>Operability</i>

Tabel 3.22 Detil *Form* Membuat Laporan Hasil Pemotongan

NamaFungsi	Membuat Laporan Hasil Pemotongan
Stakeholder	Bagian Pemotongan

Deskripsi	Membuat laporan dari hasil pemotongan yang telah dilakukan dalam periode yang ditentukan
Desain Interface	
Gambar 3.19 Desain <i>Form</i> Membuat Laporan Hasil Pemotongan	
Table Input	Tabel Hasil Pemotongan
Table Output	-
Query	Select * from hasil_pemotongan
Pseudocode	<pre> Begin Declare Otentikasi() Menginputkan_Periode_Laporan() Menampilkan_Laporan_Hasil_Pemotongan() Mencetak_Laporan_Hasil_Pemotongan() End </pre>
Kebutuhan Non-	<i>Security</i>

Fungsional	<p><i>Correctness</i></p> <p><i>Interface</i></p> <p><i>Performance</i></p> <p><i>Operability</i></p>
-------------------	---

3.3.2.10 Studi Kasus Optimasi Pemotongan Kaca

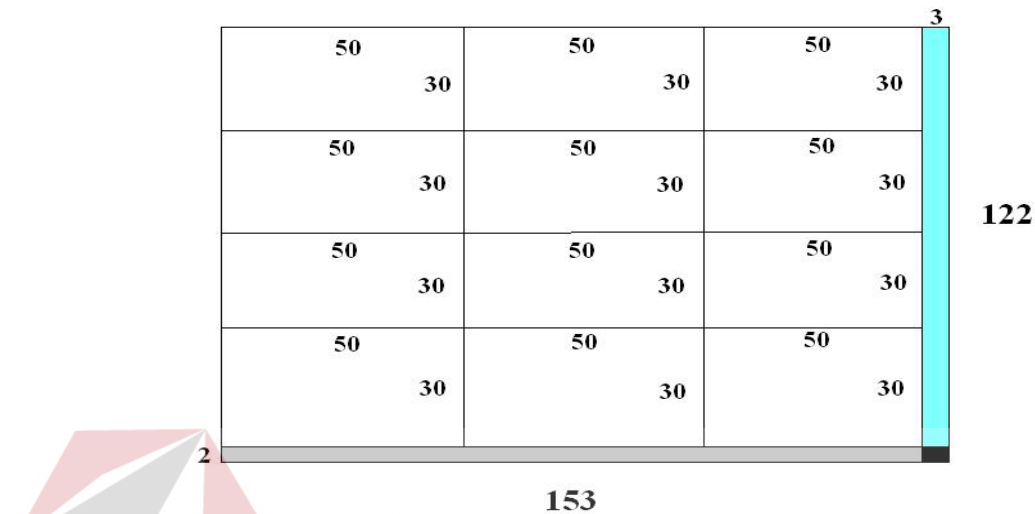
Berikut merupakan contoh dari studi kasus didalam pemotongan kaca yang digunakan untuk pembuatan Aquarium. Dari jenis kaca lembaran yang ada digunakan sebagai acuan untuk menghasilkan potongan yang sesuai. Beberapa kombinasi potongan dilakukan untuk menciptakan kondisi optimal dalam pemotongan kaca. Kombinasi tersebut menghasilkan pola potong dengan bentuk vertikal dan horizontal. Diketahui pesanan untuk kebutuhan aquarium dengan spesifikasi sebagai berikut :

Ukuran Pesanan : 30cm x 50cm

Jumlah Pesanan : 1000 Lembar

Tujuan : Menghasilkan pola pemotongan untuk 1000 lembar kaca dengan sisa potongan terkecil.

a. Kombinasi Horizontal



Gambar 3.20 Pola Pemotongan Horizontal

Perhitungan Jumlah Potongan : $(122/30) \times (153/50)$

Perkalian ini diambil pembulatan kebawah

Jadi menghasilkan : $4 \times 3 = 12$

Kombinasi ini menghasilkan 12 lembar kaca potongan dengan :

Sisa vertikal (biru muda Gambar 3.20) = $(153 - (50 \text{ cm} \times 3)) \times 122 \text{ cm}$

$$= 3 \text{ cm} \times 122 \text{ cm}$$

Sisa Horizontal (abu-abu Gambar 3.20) = $(122 \text{ cm} - (30 \text{ cm} \times 4)) \times 153 \text{ cm}$

$$= 2 \text{ cm} \times 153 \text{ cm}$$

Perpotongan antara sisa vertikal dan horizontal (hitam Gambar 3.20) = 3×2 cm

Setelah ditemukan sisa potongan, maka akan dilakukan pengecekan kembali antara ukuran order kaca dengan ukuran sisa potongan tersebut. Apabila ukuran order kaca masih lebih kecil daripada ukuran sisa potongan kaca, maka akan dilakukan optimasi kembali dan dilakukan proses perhitungan yang sama dengan cara diatas.

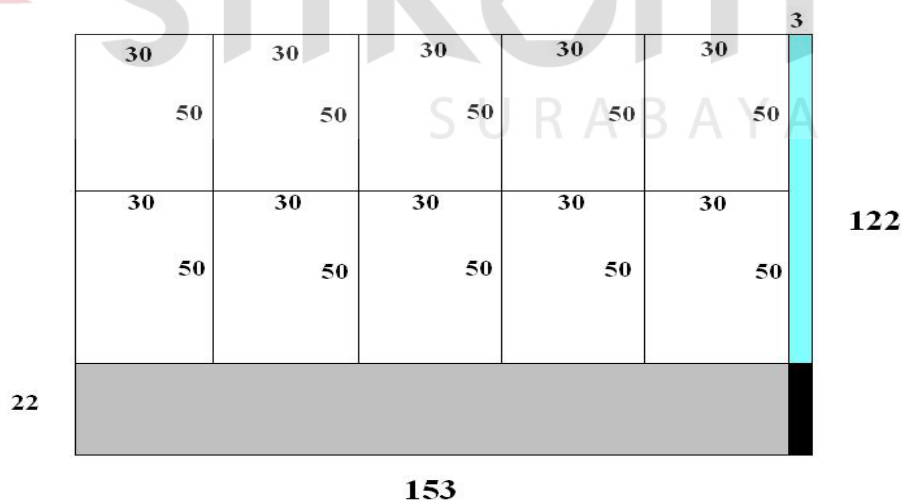
Karena sisa kaca dalam hitungan ini lebih kecil daripada ukuran pesanan maka :

$$\text{Total sisa} = (2 \text{ cm} \times 153 \text{ cm}) + (3 \text{ cm} \times 122 \text{ cm}) - (2 \text{ cm} \times 3 \text{ cm})$$

$$= 306 \text{ cm}^2 + 366 \text{ cm}^2 - 6 \text{ cm}^2$$

$$= 666 \text{ cm}^2$$

b. Kombinasi Vertikal



Gambar 3.21 Pola Pemotongan Vertikal.

$$\text{Jumlah Potongan Kaca} = (122/50) \times (153/30)$$

Perkalian ini diambil pembulatan kebawah

$$\text{Jadi menghasilkan } 2 \times 5 = 10$$

Kombinasi ini menghasilkan 10 lembar kaca dengan :

$$\text{Sisa vertical (biru muda Gambar 3.21)} = 153 - (30 \text{ cm} \times 5) \times 122 \text{ cm}$$

$$= 3 \text{ cm} \times 122 \text{ cm}$$

$$\text{Sisa Horizontal (abu-abu Gambar 3.21)} = 122 \text{ cm} - (50 \text{ cm} \times 2) \times 153 \text{ cm}$$

$$= 22 \text{ cm} \times 153 \text{ cm}$$

$$\text{Perpotongan antara sisa vertical dan horizontal (hitam Gambar 3.21)} = 3 \text{ cm} \times 22 \text{ cm}$$

Setelah ditemukan sisa potongan, maka akan dilakukan pengecekan kembali antara ukuran order kaca dengan ukuran sisa potongan tersebut. Apabila ukuran order kaca masih lebih kecil daripada ukuran sisa potongan kaca, maka akan dilakukan optimasi kembali dan dilakukan proses perhitungan yang sama dengan cara diatas.

Karena sisa kaca dalam hitungan ini lebih kecil daripada ukuran pesanan maka

$$\text{Total sisa} = (3 \text{ cm} \times 122 \text{ cm}) + (22 \text{ cm} \times 153 \text{ cm}) - (3 \text{ cm} \times 22 \text{ cm})$$

$$= 366 \text{ cm}^2 + 3366 \text{ cm}^2 - 66 \text{ cm}^2$$

$$= 3666 \text{ cm}^2$$

Dari hasil kedua kombinasi ini diambil yang paling optimum menghasilkan jumlah potongan dengan sisa terkecil untuk dibandingkan. Kombinasi horizontal mempunyai hasil perkalian dengan jumlah 12 lembar kaca dan sisa 666 cm². Jika dibandingkan dengan kombinasi vertikal yang hanya menghasilkan 10 lembar potongan dan sisa 3666 cm². Maka kombinasi horizontal adalah yang paling optimal.

Kombinasi potongan kaca dapat dilihat pada tabel 3.23 :

Tabel 3.23 Kombinasi Potongan Kaca

Ukuran	122 cm x 153 cm										
	122					153					
30	120	90	60	30	0	150	120	90	60	30	0
50	0	0	50	50	100	0	0	50	50	100	150
Sisa	2	32	12	42	22	3	33	13	43	23	3

Model Matematika untuk optimasi Horizontal pada kaca lembaran :

Z : Luas bidang potong optimal terhadap kaca lembaran.

X_{1,1} : Jumlah pola potong terhadap lebar kaca lembaran

X_{2,1} : Jumlah pola potog terhadap panjang kaca lembaran

Bentuk pemodelan pemrograman linear kombinasi horizontal, dapat dilihat pada tabel 3.24:

Tabel 3.24 Kombinasi Horizontal

Ukuran Pesanan	122	153	Pemecahan
30	120	0	270
50	0	150	
Sisa Potong	2	3	

Maksimalkan $Z = 30 X_{1,1} + 50X_{2,1}$

Batasan :

$$30 X_{1,1} \leq 122$$

$$50 X_{2,1} \leq 153$$

$$30 X_{1,1} + 50 X_{2,1} \leq 270$$

$$X_{1,1}, X_{2,1} \geq 0$$

Penyelesaian :

$$X_{1,1} \leq 122 / 30$$

$$30 (4,06) + 50 X_{2,1} \leq 270$$

$$Z = 30 (4,06) + 50 (2,96)$$

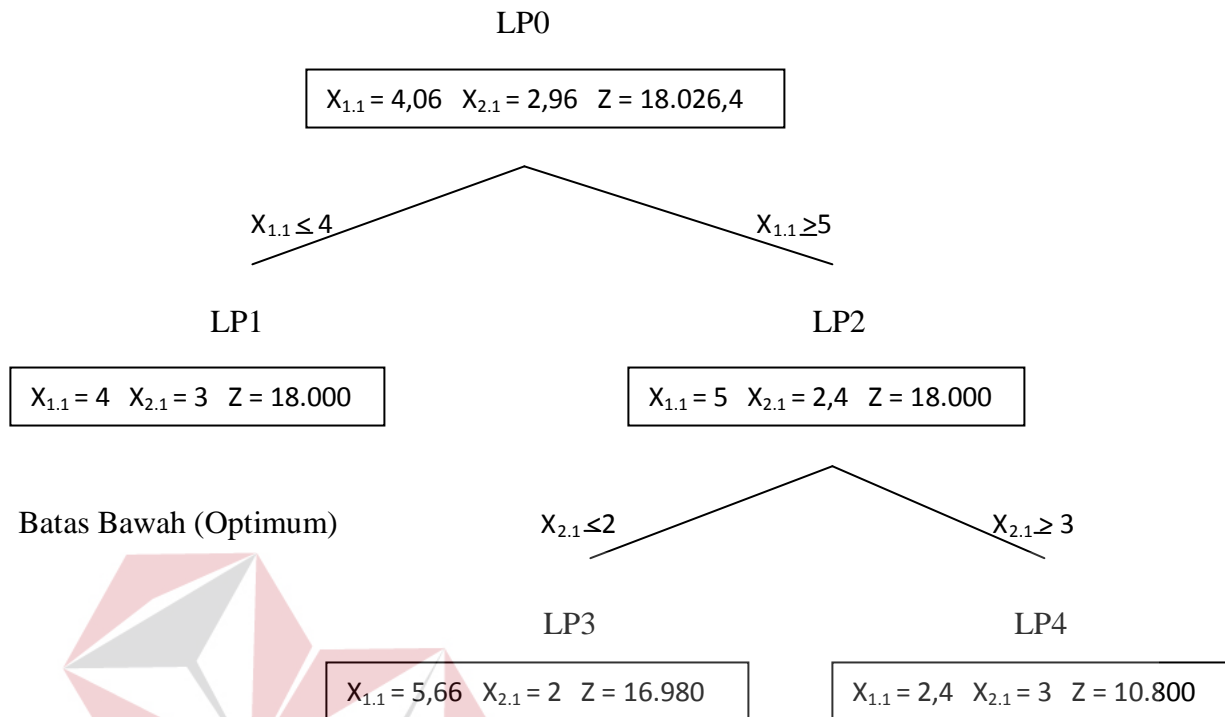
$$X_{1,1} \leq 4,06$$

$$50 X_{2,1} \leq 270 - 121,8$$

$$Z = 121,8 + 148$$

$$X_{2,1} \leq 2,96$$

$$Z = 18026,4$$



Gambar 3.22 Penyelesaian *Branch and Bound* Pola Horizontal

Setelah menggunakan metode *branch and bound* seperti pada Gambar 3.22 diketahui bahwa pemecahan optimum terdapat pada nilai LP1 sebesar $X_{1,1} = 4 \quad X_{2,1} = 3 \quad Z = 18.000$.

Dengan luas optimum pemotongan (Z) sebesar 18.000, maka didapatkan luas sisa potong sebesar:

$$\text{Luas Bidang Kaca Lembaran: } 122 \text{ cm} \times 153 \text{ cm} = 18.666 \text{ cm}^2$$

$$\text{Luas Bidang Optimum Pemotongan (Z)} = 18.000 \text{ cm}^2$$

$$\text{Luas Sisa Bidang Potong} = 18.666 - 18.000 = 666 \text{ cm}^2$$

Model Matematika untuk optimasi vertikal pada kaca lembaran :

Z : Luas bidang potong optimal terhadap kaca lembaran.

$X_{1,1}$: Jumlah pola potong terhadap lebar kaca lembaran

$X_{2,1}$: Jumlah pola potong terhadap panjang kaca lembaran

Bentuk pemodelan pemrograman linear kombinasi vertikal, dapat dilihat pada tabel 3.25:

Tabel 3.25 Kombinasi Vertikal

Ukuran Pesanan	122	153	Pemecahan
30	0	150	250
50	100	0	
Sisa Potong	22	3	

Maksimalkan $Z = 50 X_{1,1} + 30X_{2,1}$

Batasan :

$$50 X_{1,1} \leq 122$$

$$30 X_{2,1} \leq 153$$

$$50 X_{1,1} + 30 X_{2,1} \leq 250$$

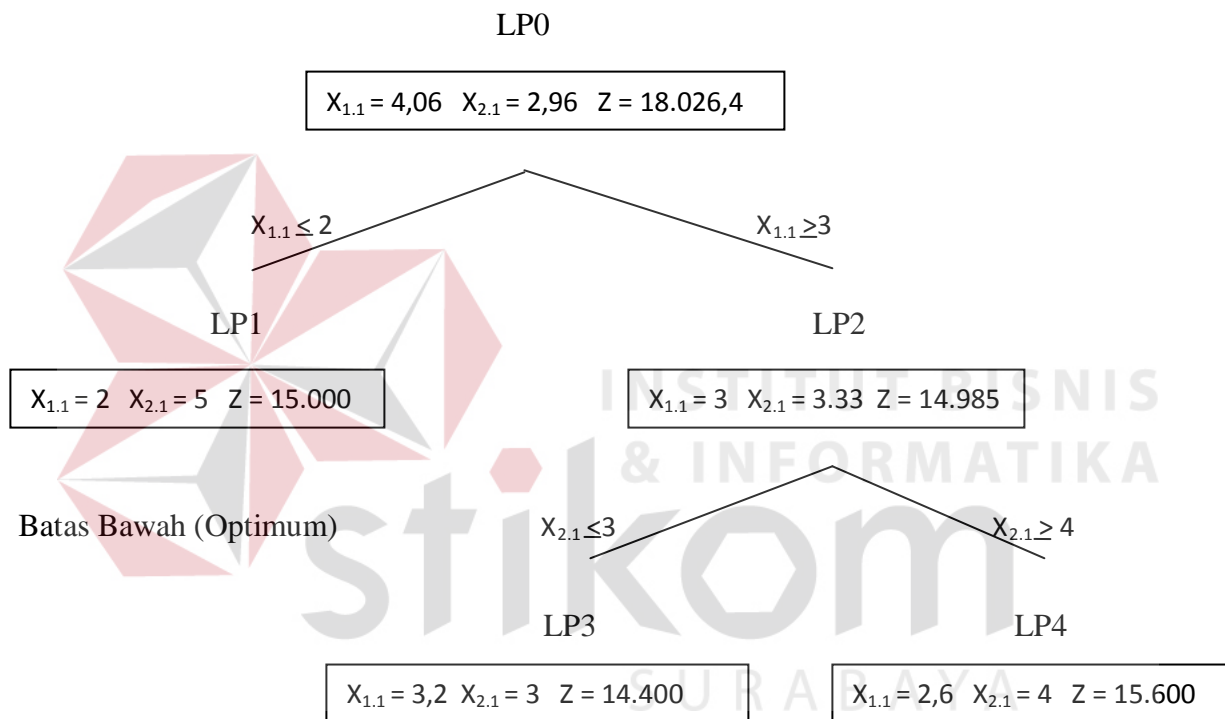
$$X_{1,1}, X_{2,1} \geq 0$$

Penyelesaian :

$$X_{1,1} \leq 122 / 50 \quad 50 (2,44) + 30 X_{2,1} \leq 250 \quad Z = 50 (2,44) + 30 (4,26)$$

$$X_{1,1} \leq 2,44 \quad 30 X_{2,1} \leq 250 - 122 \quad Z = 122 + 127,8$$

$$X_{2,1} \leq 4,26 \quad Z = 15.591,6$$



Gambar 3.23 Penyelesaian *Branch and Bound* Pola Vertikal

Setelah menggunakan metode *branch and bound* seperti pada Gambar 1.4 diketahui bahwa pemecahan optimum terdapat pada nilai LP1 sebesar $X_{1,1} = 2$ $X_{2,1} = 5$ $Z = 15.000$.

Dengan luas optimum pemotongan (Z) sebesar 15.000, maka didapatkan luas sisa potong sebesar:

$$\text{Luas Bidang Kaca Lembaran: } 122 \text{ cm} \times 153 \text{ cm} = 18.666 \text{ cm}^2$$

$$\text{Luas Bidang Optimum Pemotongan (Z)} = 15.000 \text{ cm}^2$$

$$\text{Luas Sisa Bidang Potong} = 18.666 - 15.000 = 3.666 \text{ cm}^2$$

Dari perhitungan diatas, dapat disimpulkan bahwa nilai optimum kombinasi Horizontal layak dipakai sebagai acuan pola pemotongan kaca karena menghasilkan nilai optimal dengan 12 lembar kaca potongan dengan luas bidang potong sebesar 18.000 cm² dan sisa bidang potong 666 cm². Dari keseluruhan total 1000 lembar kaca potongan yang harus dipotong, dari 1 lembar ukuran lembaran menghasilkan 12 lembar kaca potongan, maka yang akan dipakai sebesar $1000/12 = 84$ kaca lembaran.

3.3.2.11 Unit Program

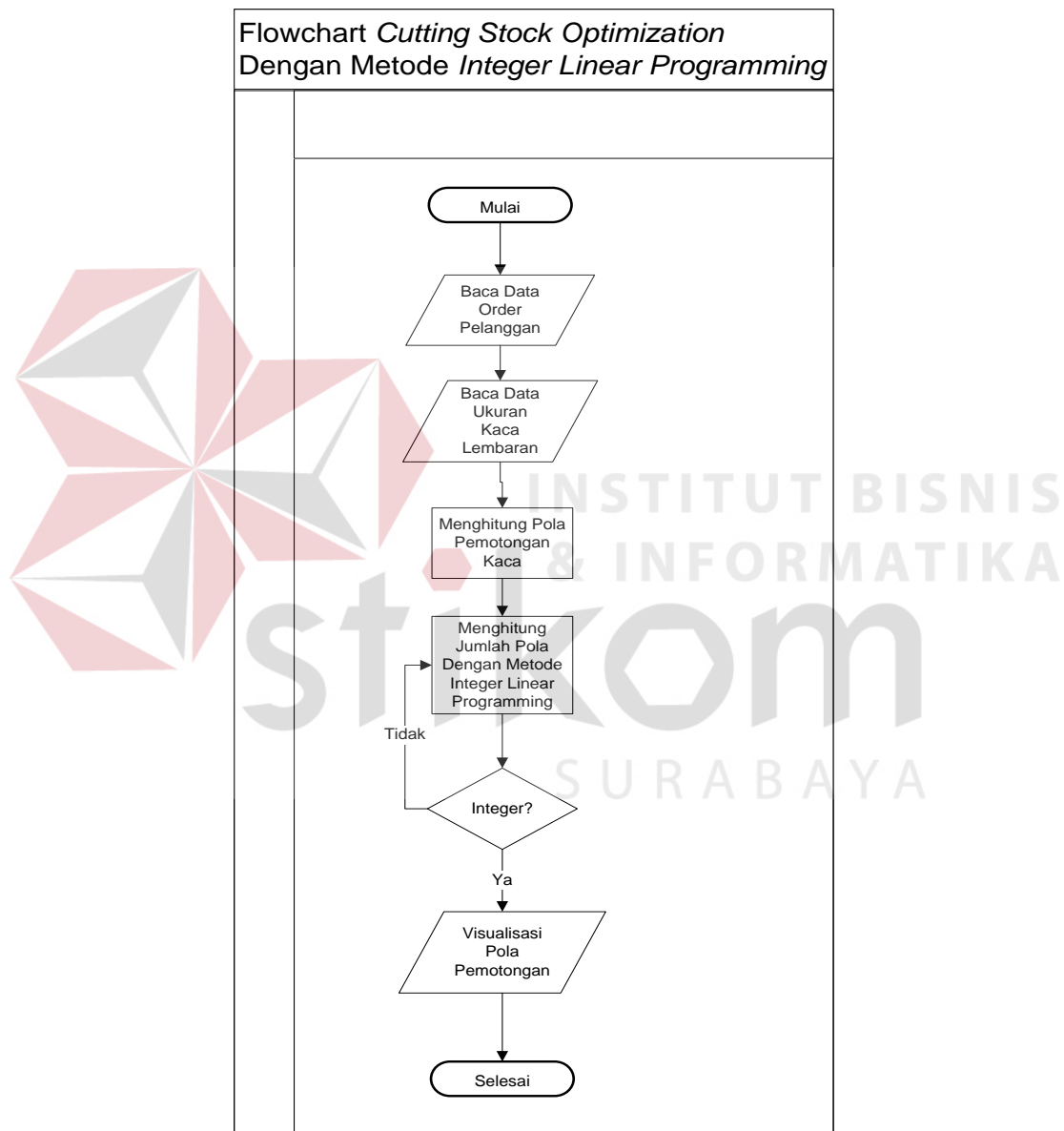
Unit Program merupakan kumpulan dari setiap *pseudocode* yang ada dalam setiap fungsi yang akan dibangun yang berfungsi sebagai dasar dalam membangun aplikasi dan menerapkan fungsi-fungsi tersebut ke dalam pemrograman dan konstruksi aplikasi yang akan dikembangkan. Program unit tersebut seperti terlihat pada Tabel 3.29.

Tabel 3.26 Prosedur dan Program Unit Sistem

NamaFungsional	Program Unit
Mencatat Order Pelanggan	Otentikasi() Menginputkan_Order_Pelanggan() Menyimpan_Order_Pelanggan() Mencetak_Bukti_Order_Pelanggan()
Membuat Purchase Order	Otentifikasi() Menampilkan_Data_Purchase_Order() Mengupdate_Jumlah_Purchase_Order() Mencetak_Purchase_Order()
Menghitung Pola Pemotongan	Otentikasi() Menampilkan_Data_Order_Pelanggan() Menghitung_Pola_Pemotongan_Kaca() Mengecek_Ketersediaan_Kaca() Menyimpan_Jumlah_Kekurangan_Kaca() Menghitung_Dengan_Metode_Integer_Linear_Programming() Menyimpan_Data_Hasil_Pemotongan() Mengupdate_Stok_Kaca() Menyimpan_Stok_Sisa_Kaca()
Membuat Laporan Hasil Pemotongan	Otentikasi() Menginputkan_Periode_Laporan() Menampilkan_Laporan_Hasil_Pemotongan() Mencetak_Laporan_Hasil_Pemotongan()

3.3.2.12. Flowchart Program

Berikut merupakan urutan flowchart dari *cutting stock optimization* dengan metode *Integer Linear Programming*. Untuk lebih jelasnya dapat dilihat pada gambar 3.24.




Gambar 3.24 *Flowchart Program*

3.3.3.13 Pseudocode

Berikut ini merupakan susunan *pseudocode* secara detil dari prosedur-prosedur yang telah dirancang:

Tabel 3.27 *Pseudocode* Program Unit

No.	Program Unit	Pseudocode
1.	Otentikasi()	<pre> Start Declare String X, Y, Z, A X = Id Pegawai Y = Password Z = db.Id Pegawai A = db.Password If X = Z then If Y = A then Show Menu Utama Else Print "Password Anda Salah" End if Else Print "Username Anda Salah" End if End </pre>
2.	Menghitung_Pola_Pemotongan_Kaca ()	<pre> Start Declare String A Double X, Y, B, X1, Y1 Integer jumPolaT, jumPolaL, jumPola A = id order pelanggan X = panjang kaca yang diorder Y = lebar kaca yang diorder B = db.jumlah kaca lembaran While B <> 0 X1 = db.panjang kaca lembaran Y1 = db.lebar kaca lembaran /*"Pola Horizontal" jumPolaT = X1 / X </pre>

		<pre> jumPolaT = Int(jumPolaT) jumPolaL = Y1 / Y jumPolaL = Int(jumPolaL) jumPola = jumPolaT * jumPolaL /*“Pola Vertikal” jumPolaT1 = [X1] / [Y] jumPolaT1 = Int(jumPolaT1) jumPolaL1 = Y1 / X jumPolaL1 = Int(jumPolaL1) jumPola1 = jumPolaT1 * jumPolaL1 B = B - 1 End while End </pre>
3.	 <p>Menghitung Dengan Metode _Integer_Linear_Programming()</p>	<pre> Start Declare Integer A, B Double X, X1, Y, Y1, X0, Y0, Z0 X = panjang kaca yang diorder Y = lebar kaca yang disorder X1 = db.panjang kaca lembaran Y1 = db lebar kaca lembaran A = Jumlah Pola Potong Horizontal B = Jumlah Pola Potong Vertikal A = A * X B = B * Y C = A + B D = Jumlah Pola Pematongan While D <> 0 X0 = Y1 / Y Y0 = (C - (Y * X0)) / X Z0 = (Y * X0) * (X * Y0) [D] = [D] + 1 End while End </pre>

